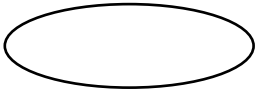

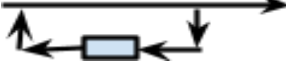

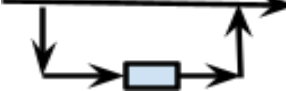


## Práctica 2: Sintaxis

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

Meta símbolos utilizados por		Símbolo utilizado en Diagramas sintácticos	Significado
BNF	EBNF		
palabra terminal	palabra terminal		Definición de un elemento terminal
< >	< >	rectángulo 	Definición de un elemento no terminal
::=	::=	diagrama con rectángulos, óvalos y flechas	Es un metasímbolo y define una producción / regla
	( )	flecha que se divide en dos o más caminos	Selección de una alternativa
< p > < p1 >	{ }	flecha que vuelve a la condición	Repetición
	*		Repetición de 0 o más veces
	+		Repetición de 1 o más veces
	[ ]		Opcional, está presente o no lo está

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 el conjunto de reglas que definen cómo componer letras, dígitos y otros caracteres que permiten formar los programas y que la comunicación entre el programador y la computadora sea clara. Permite escribir programas válidos y coherentes, correctos sintácticamente.

Los elementos de la sintaxis son:

– Alfabeto o conjunto de caracteres:

- Conjunto finito de símbolos admitido en el lenguaje.
- La secuencia de bits que compone cada carácter la determina la implementación.

– Identificadores:

- Cadena de letras y dígitos, que deben comenzar con una letra
- Si se restringe la longitud se pierde legibilidad

– Operadores:

- Con los operadores de suma, resta, etc. la mayoría de los lenguajes utilizan +, -. En los otros operadores no hay tanta uniformidad

– Comentarios y uso de blancos:

- Hacen los programas más legibles

– Palabra clave y palabra reservada:

- Palabra clave o keywords: son palabras que tienen un significado dentro de un contexto.
- Palabra reservada: son palabras claves que además no pueden ser usadas por el programador como identificador de otra entidad.

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

- Reglas Léxicas/Lexicográficas: Conjunto de reglas para formar las “word”, a partir de los caracteres del alfabeto.
- Reglas Sintácticas: Conjunto de reglas que definen cómo formar las “expresiones” y “sentencias”.

**Ejercicio 4:** ¿En la definición de un lenguaje, a qué se llama palabra reservada? ¿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 palabras claves que no pueden ser utilizadas por el programador para identificar otra entidad, es decir, son palabras con un significado especial dentro del lenguaje y que no pueden ser utilizadas para nombrar variables o funciones. Estas, se corresponden con el conjunto de símbolos terminales de la gramática.

Un ejemplo de palabra reservada en JavaScript es var. Esta palabra se utiliza para declarar una variable en JavaScript, por lo que no puede ser utilizada como nombre de variable. Otros ejemplos de palabras reservadas en JavaScript incluyen if, else, while, for, function, return, true, false, null, undefined, try, catch, throw, typeof, entre otras.

**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

b- Indique porqué es ambigua y corrijala

a- Los componentes de la misma son:

G: Gramática, son reglas finitas que definen un conjunto infinito de posibles sentencias válidas en el lenguaje. Está formada por una 4-tupla  $G = (N, T, S, P)$

N: Es el conjunto de símbolos no terminales, en este caso los no terminales serían `<numero_entero>` y `<digito>`.

T: Es el conjunto de símbolos terminales. En este caso es el conjunto: {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

S: Es el símbolo distinguido de la gramática. En este ejemplo es: `<numero_entero>`.

P: Es el conjunto de producciones de la gramática. En este ejemplo tenemos dos producciones:

- `<numero_entero> ::= <digito><numero_entero> | <numero_entero><digito> | <digito>`
- `<digito> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9`

b - La solución es ambigua debido a que una sentencia puede derivarse de más de una forma, en este caso está utilizando recursión por derecha y por izquierda al mismo tiempo, lo que podría generar problemas en el árbol de sintaxis. La solución correcta sería:

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

$N = \{<numero\_entero>, <digito>\}$

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

$S = <numero\_entero>$

$P = \{$

`<numero_entero> ::= <digito> | <digito><numero_entero>`

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

$\}$

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

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

$N = \{<palabra>, <mayúscula>, <minúscula>\}$

$T = \{A, \dots, Z, a, \dots, z\}$

$S = <palabra>$

$P = \{$

`<palabra> ::= <mayúscula><palabra> | <minúscula><palabra> | <minúscula> | <mayúscula>`

`<mayúscula> ::= A | ... | Z`

`<minúscula> ::= a | ... | z`

$\}$

**Ejercicio 7:** Defina en EBNF la gramática para la definición de números reales. Inténtelo desarrollar para BNF y explique las diferencias con la utilización de la gramática EBNF.

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

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

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

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

$P = \{$

$\langle \text{numero\_real} \rangle ::= [-] \{ \langle \text{digito} \rangle \}^+ [., \{ \langle \text{digito} \rangle \}^*]$

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

$\}$

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

$N = \{ \langle \text{real} \rangle, \langle \text{integer} \rangle, \langle \text{digito} \rangle \}$

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

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

$P = \{$

$\langle \text{real} \rangle ::= \langle \text{integer} \rangle " " \langle \text{integer} \rangle \mid + \langle \text{integer} \rangle " " \langle \text{integer} \rangle \mid - \langle \text{integer} \rangle " " \langle \text{integer} \rangle \mid \langle \text{integer} \rangle$   
 $\mid + \langle \text{integer} \rangle \mid - \langle \text{integer} \rangle$

$\langle \text{integer} \rangle ::= \langle \text{digito} \rangle \mid \langle \text{digito} \rangle \langle \text{integer} \rangle$

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

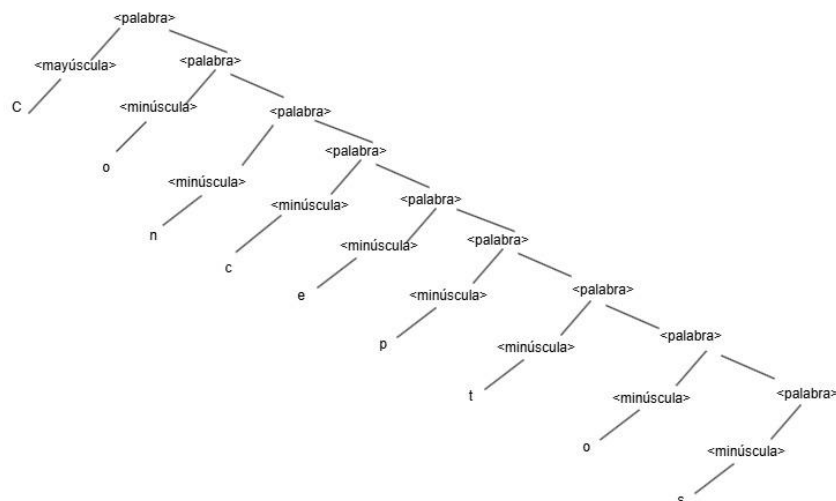
$\}$

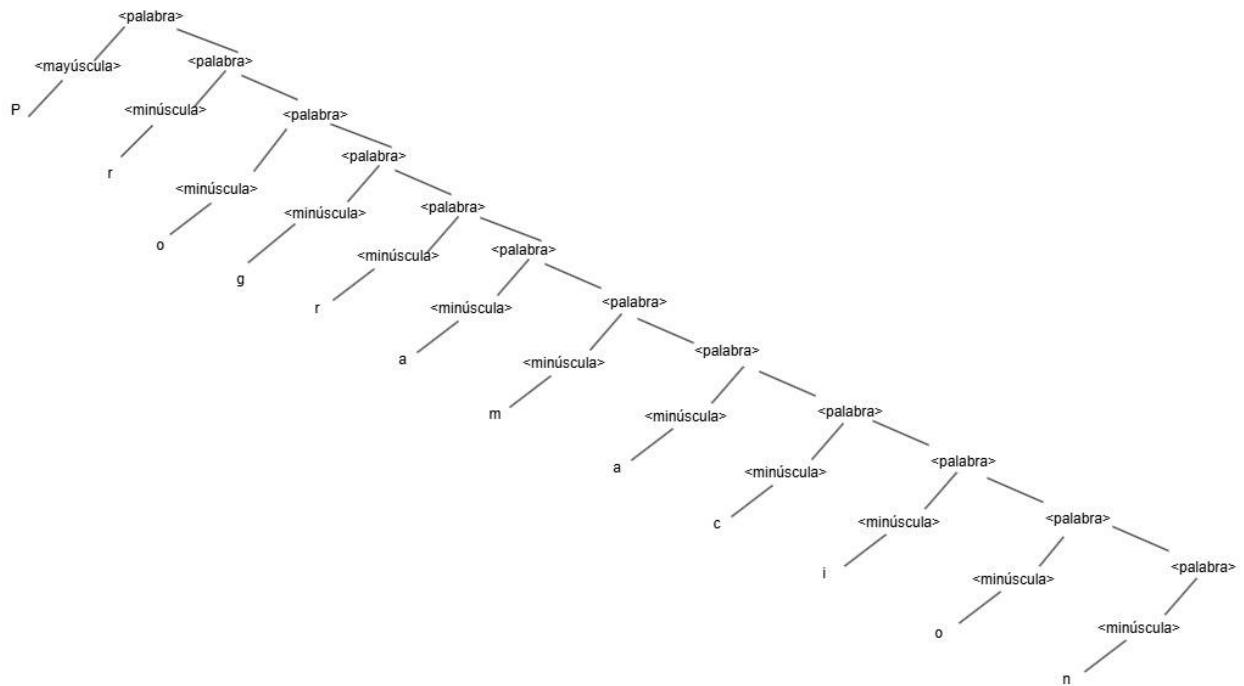
La diferencia más clara entre las dos formas es la cantidad de símbolos no terminales que necesita la BNF para poder definir lo mismo que definimos en la EBNF con menos símbolos.

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

- Conceptos
- Programación
- 1255869
- 854,26
- Conceptos de lenguajes

8. a.





```

graph TD
    A["<numero_real>"] --> B["<digito>"]
    A --> C["<digito>"]
    A --> D["<digito>"]
    A --> E["<digito>"]
    A --> F["<digito>"]
    A --> G["<digito>"]
    A --> H["<digito>"]
    B --> I["1"]
    C --> J["2"]
    D --> K["5"]
    E --> L["5"]
    F --> M["8"]
    G --> N["6"]
    H --> O["9"]

```

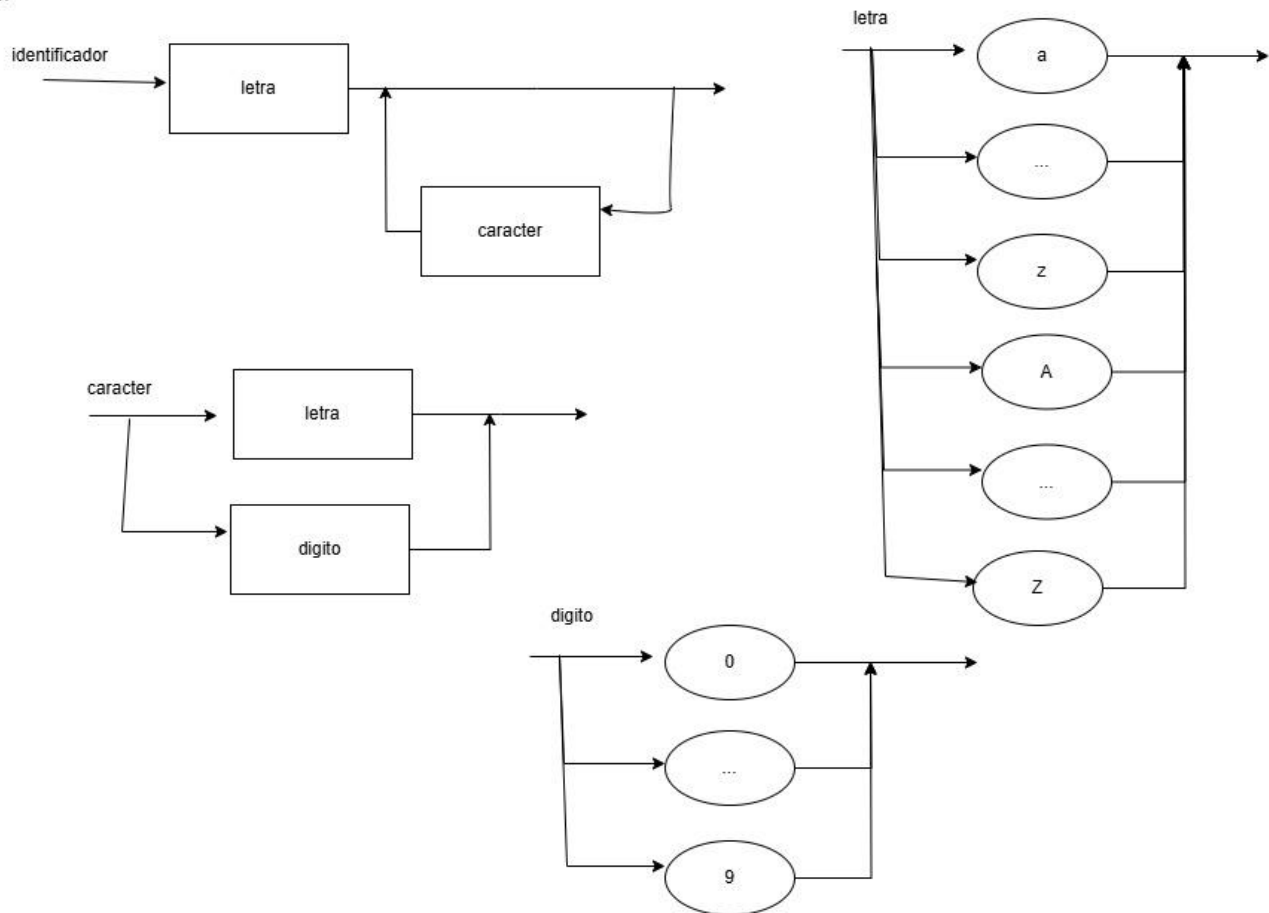
```

graph TD
    A["<numero_real>"] --> B["<digito>"]
    A --> C["<digito>"]
    A --> D["<digito>"]
    A --> E["'.'"]
    A --> F["<digito>"]
    A --> G["<digito>"]
    B --> B1["8"]
    C --> C1["5"]
    D --> D1["4"]
    F --> F1["2"]
    G --> G1["6"]

```

5

9.

**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.

$G = \{N, T, S, P\}$

$N = \{<expr>, <variable>, <operador>, <número>, <dígito>, <letra>\}$

$T = \{a, \dots, z, A, \dots, Z, 0, \dots, 9, "+", "-", "*", "/" \}$

$S = <expr>$

$P = \{ <expr> ::= ( <variable> \mid <número> ) \{ <operador> ( <id> \mid <número> ) \}^+$

$<número> ::= [(+|-)] \{ <dígito> \}^+$

$<variable> ::= <letra> \{ ( <letra> \mid <dígito> ) \}^*$

$<dígito> ::= ( 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9 )$

$<operador> ::= ( + \mid - \mid * \mid / )$

$<letra> ::= ( a \mid \dots \mid z \mid A \mid \dots \mid Z )$

$\}$

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

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

$N = \{<exp>, <term>, <elem>, <iden>, <letra>\}$

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

$S = \{<exp>\}$

$P = \{ <exp> ::= <term>\{ (+ | -) <term> \}^*$

$<term> ::= <elem>\{ (* | /) \}^*$

$<elem> ::= (<iden> | <num>)$

$<iden> ::= <letra>\{ (<letra> | <digito>) \}^*$

$<letra> ::= (a | b | c | \dots)$

$<num> ::= <digito>\{ <digito> \}^*$

$<digito> ::= (0 | 1 | 2 | \dots)$

$\}$

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

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

$N = \{ <sentencia\_for>, <bloque>, <variable>, <letra>, <cadena>, <digito>, <otro>, <operación>, <llamada\_a\_funcion>, <numero>, <sentencia> \}$

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

$<variable> ::= <letra> | <cadena>$

$<cadena> ::= \{ ( <letra> | <digito> | <otro> ) \}^+$

$<letra> ::= (a | .. | z | A | .. | Z)$

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

$<bloque> ::= <sentencia> | <sentencia> <bloque> | <bloque> <sentencia>;$

$<sentencia> ::= <sentencia\_asignacion> | <llamada\_a\_funcion> | <sentencia\_if> | <sentencia\_for> |$

$<sentencia\_while> | <sentencia\_switch> \}$

Errores:

- Una gramática está formada por una 4-tupla.  $G = (N, T, S, P)$ , falta los conjuntos de símbolos terminales (T) y el símbolo distinguido de la gramática (S) que pertenece a N.
- En P no están definidos:  $<otro>$ ,  $<operación>$ ,  $<llamada\_a\_funcion>$ ,  $<numero>$ .
- En N no están definidos:  $<sentencia\_asignacion>$ ,  $<sentencia\_if>$ ,  $<sentencia\_while>$ ,  $<sentencia\_while>$ .
- La gramática es ambigua, ya que la producción  $<bloque> ::= <sentencia> | <sentencia> <bloque> | <bloque> <sentencia>;$  es recursiva tanto derecha por izquierda, pudiéndose derivar de más de una forma.

**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>))

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

$N = \{ <div>, <inner\_html>, <etiqueta>, <palabra> \}$

$T = \{a, \dots, z, A, \dots, Z, <, >, /, \}$

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

$P = \{ \langle \text{div} \rangle ::= "<\text{div}>" \{ \langle \text{elemento\_html} \rangle \mid \langle \text{palabra} \rangle \}^+ "</\text{div}>"$

$\langle \text{elemento\_html} \rangle ::= \langle \text{etiqueta} \rangle (\langle \text{elemento\_html} \rangle \mid \langle \text{palabra} \rangle) \langle \text{etiquetaCierre} \rangle$

$\langle \text{etiqueta} \rangle ::= "<" \langle \text{palabra} \rangle ">"$

$\langle \text{etiquetaCierre} \rangle ::= "</" \langle \text{palabra} \rangle ">"$

$\langle \text{palabra} \rangle ::= \{ \langle \text{letra} \rangle \}^+$

$\langle \text{letra} \rangle ::= (a \mid \dots \mid z \mid A \mid \dots \mid Z)$

$\}$

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

La gramática a usar para este ejercicio sería la de los números enteros, pero una gramática pura no puede garantizar la primalidad de un número, la verificación de la misma debe hacerse fuera de la gramática, en una etapa semántica o algorítmica.

**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)

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

$N = \{ \langle \text{funcion} \rangle \}$

$T = \{a, \dots, z, A, \dots, Z, 0, \dots, 9\}$

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

$P = \{$

$\langle \text{funcion} \rangle ::= \text{"function"} \langle \text{palabra} \rangle \langle \text{parametros} \rangle \text{"{" } \{ \langle \text{palabra} \rangle \}^+ \text{"}"$

$\langle \text{parametros} \rangle ::= \text{"(" } [ \{ \langle \text{palabra} \rangle \} \mid \text{"{ } \{ \langle \text{palabra} \rangle \}^+ \text{"}] \text{"}"$

$\langle \text{palabra} \rangle ::= \langle \text{letra} \rangle \{ (\langle \text{letra} \rangle \mid \langle \text{dig} \rangle)^*$

$\langle \text{letra} \rangle ::= (a \mid \dots \mid z \mid A \mid \dots \mid Z)$

$\langle \text{dig} \rangle ::= (0 \mid \dots \mid 9)$

$\}$