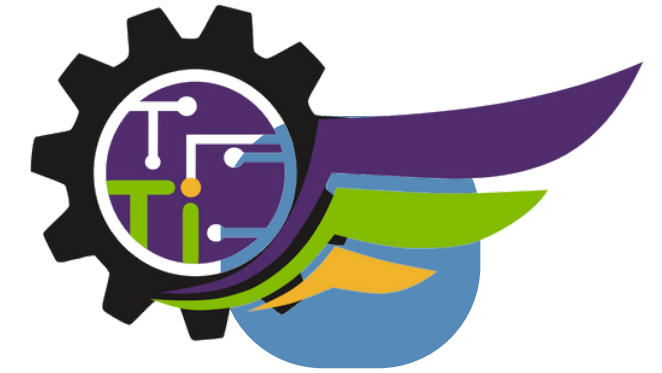




**UNIVERSIDAD POLITÉCNICA
DE PUEBLA**



Analizador Sintáctico

En el lenguaje JAVA

**Lenguajes y Autómatas
7A ITI**

Diana Aylin González Romero
Miguel Ángel Zacahua Morales
Johanna Rocha Santiago



RESUMEN DE CONTENIDO

IF

SWITCH

FOR

WHILE

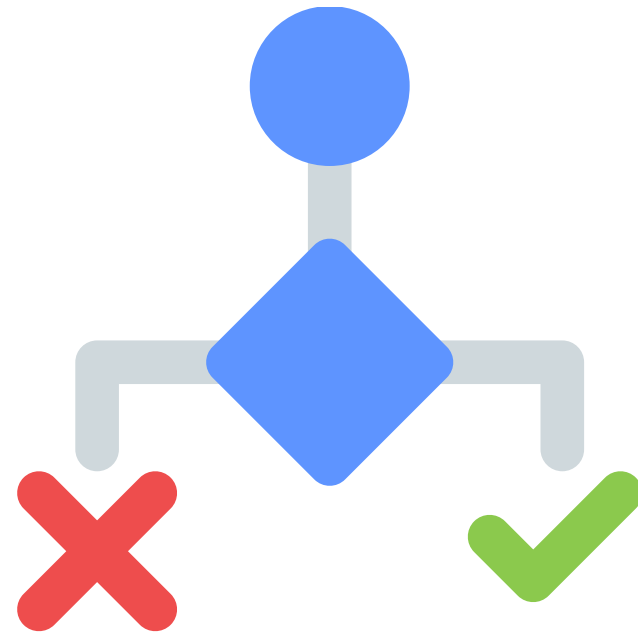
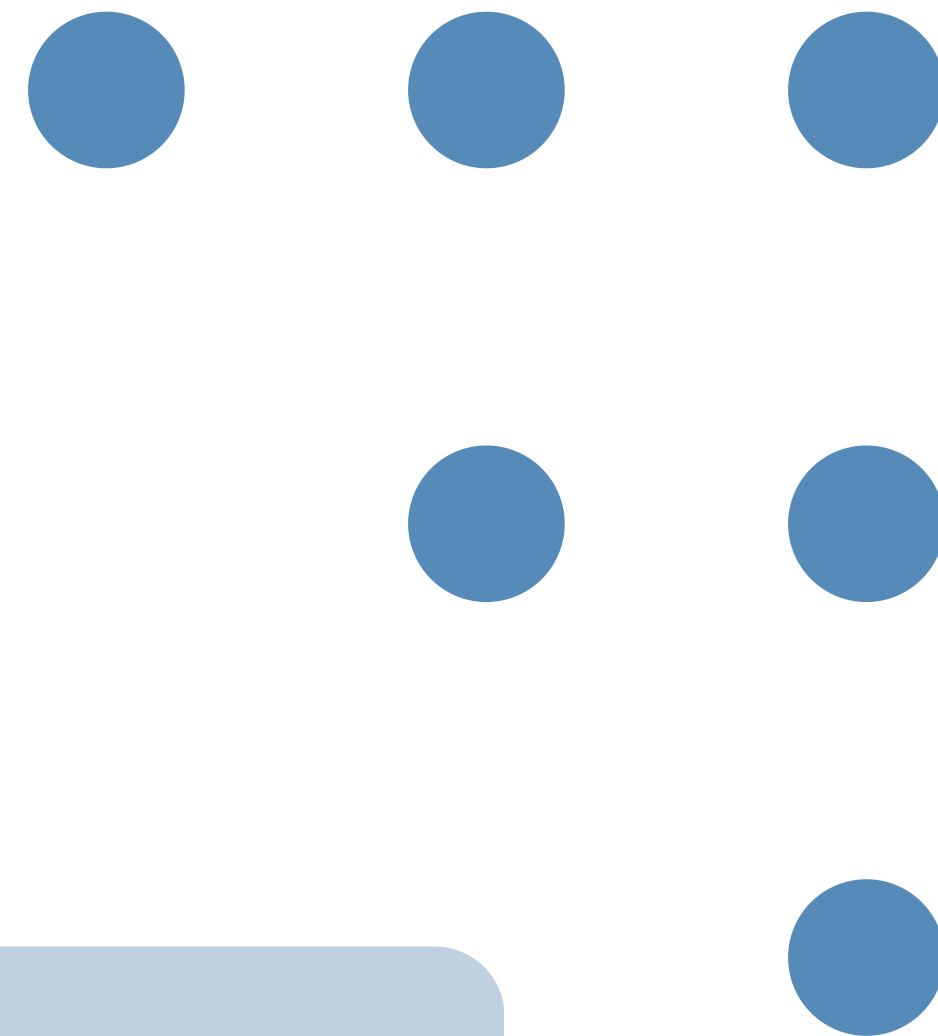
FUNCIONES

EXPRESIONES ALGEBRAICAS



IF

ANALIZADOR



IF

¿QUÉ ES? Y ¿CÓMO FUNCIONA?

Un if se utiliza para evaluar una expresión condicional. Si la condición es verdadera, ejecutará un bloque de código. Si es falsa, es posible ejecutar otras sentencias. La condición es una expresión lógica que solo devuelve true o false, y se ejecutará si su valor es true.

CÓDIGO

```
Main.java
1 if (x > 0) {
2     System.out.println("x es positivo");
3 } else {
4     System.out.println("x no es positivo");
5 }
6
```

$\langle \text{condiciona_ifl} \rangle ::= \text{if} \langle \text{condicion_logica} \rangle \{$
 $\langle \text{sentencias} \rangle;$
 $[\text{else} \{$
 $\langle \text{sentencias} \rangle;]$
 $\}$

BNF

$\langle \text{condicion_logica} \rangle ::= \text{if}(\langle \text{identificador} \rangle \langle \text{comparador} \rangle \langle \text{valor} \rangle)$

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

$\langle \text{valor} \rangle ::= \langle \text{digito} \rangle \langle \text{digito} \rangle^*$

$\langle \text{sentencias} \rangle ::= \langle \text{sentencia} \rangle \mid \{ \langle \text{sentencia} \rangle \}$

$\langle \text{sentencia} \rangle ::= \text{"System.out.println" "("} \langle \text{mensaje} \rangle \text{"}"}$

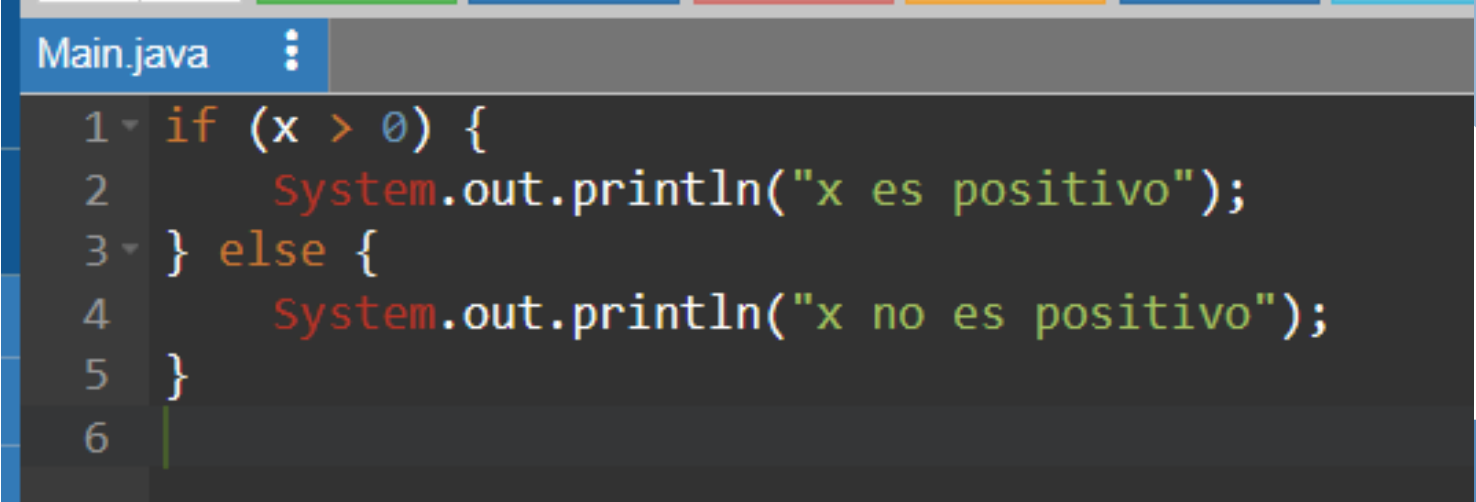
$\langle \text{mensaje} \rangle ::= \langle \text{texto} \rangle$

$\langle \text{texto} \rangle ::= \text{"x es positivo"} \mid \text{"x no es positivo"}$

$\langle \text{digito} \rangle ::= \text{"0"} \mid \text{"1"} \mid \dots \mid \text{"9"}$

$\langle \text{comparador} \rangle ::= \text{"<"} \mid \text{">"} \mid \text{"=="} \mid \text{"!="} \mid \text{"<="} \mid \text{">="}$

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



```
Main.java
1 if (x > 0) {
2     System.out.println("x es positivo");
3 } else {
4     System.out.println("x no es positivo");
5 }
6
```



ANALIZADOR

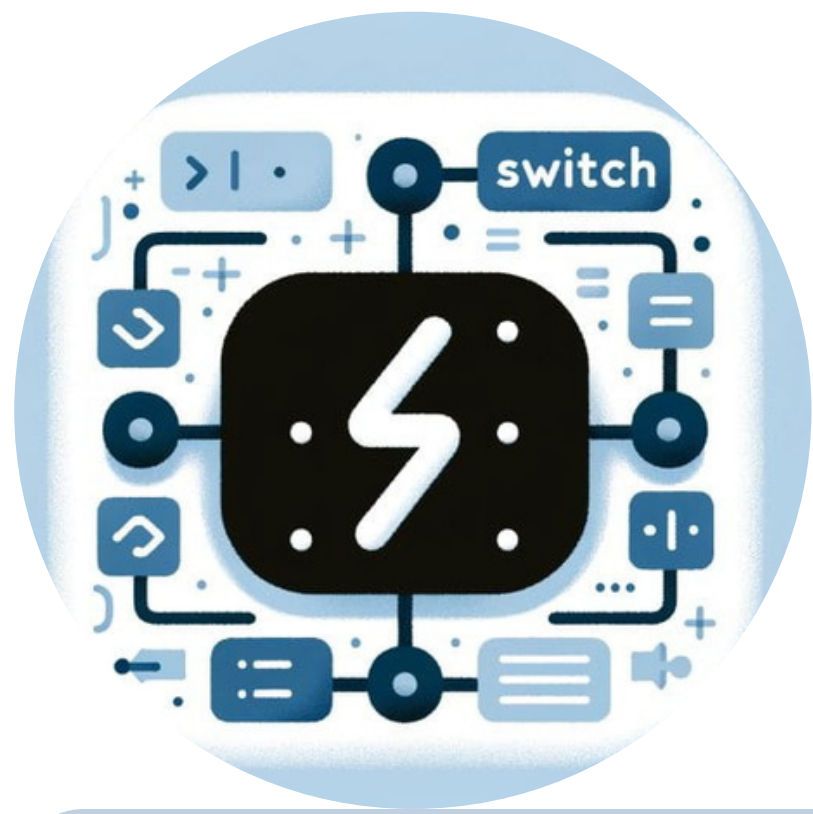
Es una estructura de control que permite elegir entre múltiples casos basándose en una expresión

Sintaxis Switch

```
switch (variable) {  
    case valor1:  
        break;  
    case valor2:  
        break;  
    default:  
}
```

Ejemplo

```
switch (variable) {  
    case 1:  
        System.out.println("La variable es igual a 1");  
        break;  
    case 2:  
        System.out.println("La variable es igual a 2");  
        break;  
    default:  
        System.out.println("La variable no coincide ");  
}
```

EXPRESION EN BNF

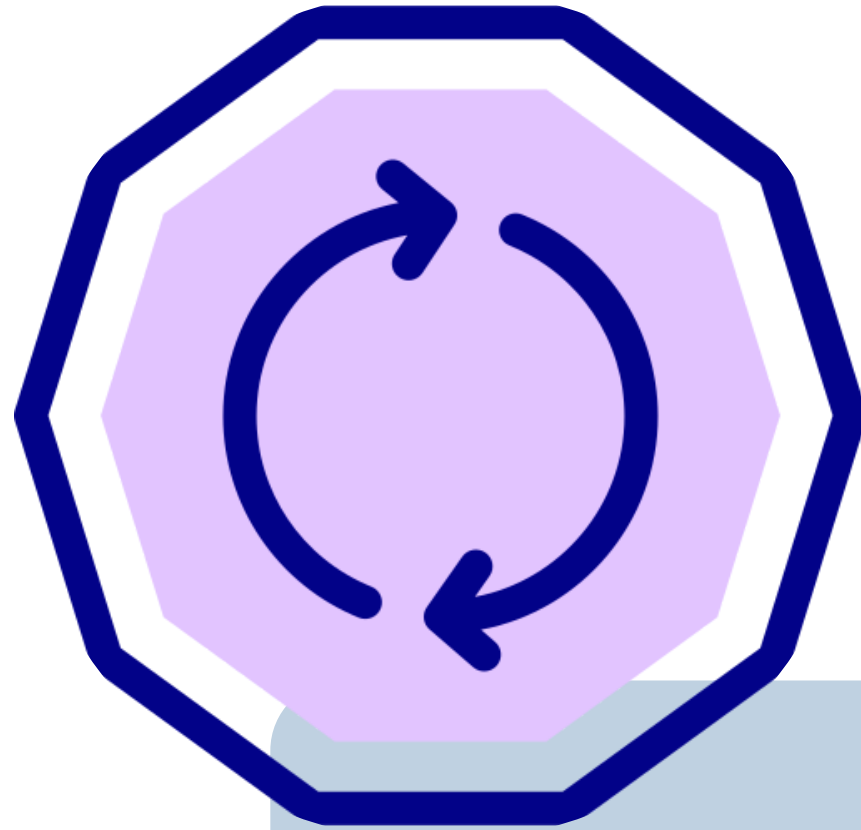
```
<Control_Switch> ::= "Switch" "(" <Expresion> ")" "{" <Bloque_expresiones> "}"  
<Bloque_expresiones> ::= <Caso_Switch> ":" { <Bloque_declaraciones> } | "break ;"  
<Caso_Switch> ::= "Case" <expresion> ":" | "default :"
```



For Loop

FOR

ANALIZADOR



FOR

¿PARA QUE SIRVE?

Sirve principalmente para ejecutar un bloque de código repetidamente hasta que se cumpla una condición específica.

Ejemplo:

```
for (int i = 0; i < 5; i++) {  
    System.out.println("El valor de i es: " + i);  
}
```

<declaracion_for> ::= <ciclo_for>

<ciclo_for> ::= for (<inicio_for> ; <condicion_for> ; <control>) {
 <contenido>
}

<inicio_for> ::= int <identificador> = <valor>

<condicion_for> ::= <identificador> <comparador> <valor>

<control> ::= <identificador> ++ | <identificador> --

<valor> ::= 0|...|9

<identificador> ::= A|...|Z|a|...|z

<comparador> ::= < | > | = | != | == | >= | <=

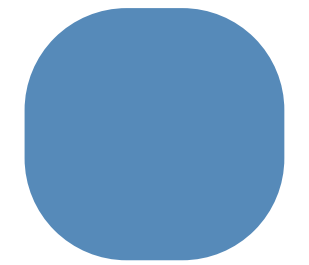
<contenido> ::= <conjunto_sentencias>

<conjunto_sentencias> ::= System.out.println(<mensaje>) | <suma> | <sentencia_if> | ...

<mensaje> ::= <texto> + <variable>

<texto> ::= A|...|Z|a|...|z

<variable> ::= <identificador>



WHILE

ANALIZADOR

While

Es una estructura de control que permite ejecutar un bloque de código repetidamente mientras una condición específica sea verdadera

Sintaxis

```
While( Condicion ) {  
    //Bloque de codigo a ejecutar  
}
```

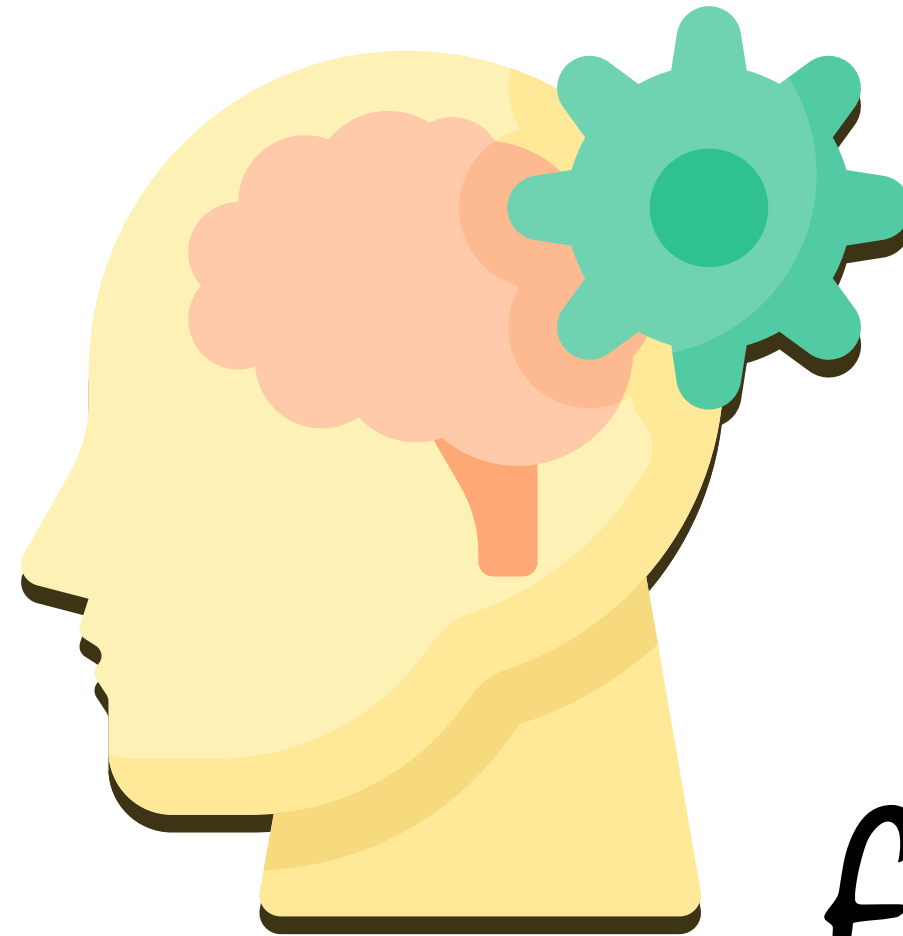
Ejemplo

```
While (var1 <= 5){  
    system.out.println(var1);  
    i++;  
}
```

Expresión BNF

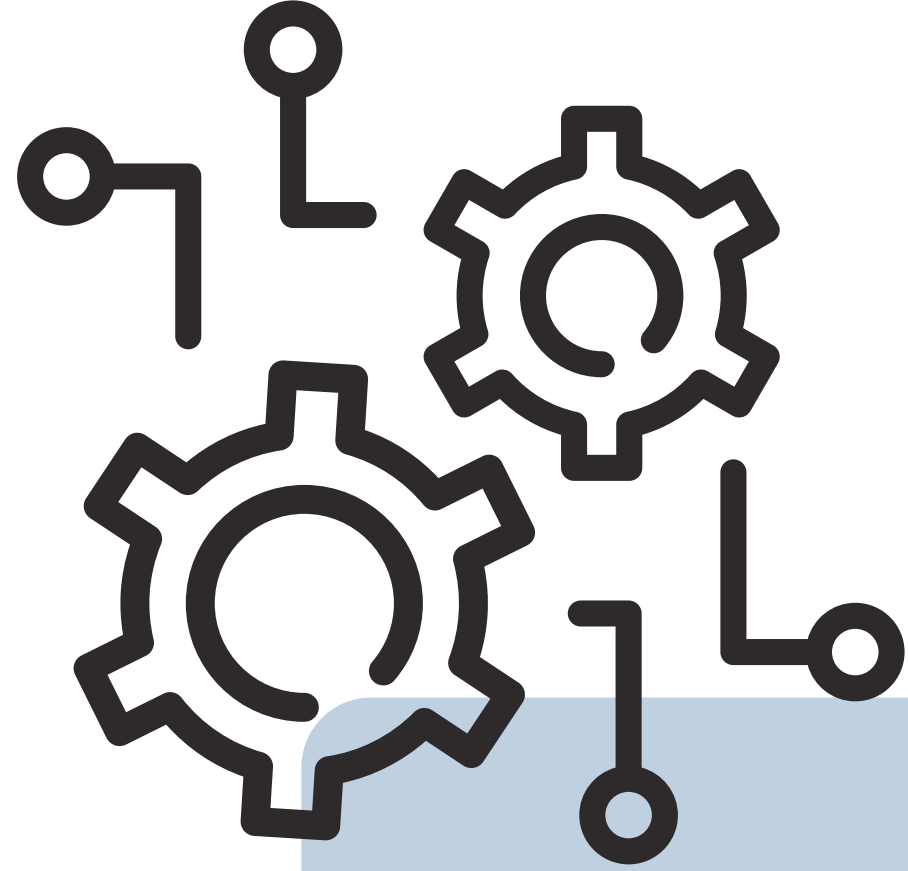
<Control_While> ::= "While" " : " <Condicion> ") " " { " <bloque expresiones> " }

FUNCIONES



$f(x)$

ANALIZADOR



FUNCIONES

¿QUÉ ES? Y ¿CÓMO FUNCIONA?

Bloque de código que realiza una tarea específica y se ejecuta cuando es llamado. Las funciones pueden recibir datos como entrada (parámetros), procesar estos datos y devolver un resultado (valor de retorno).

CÓDIGO

```
1 public class Main {  
2     public static int sumar(int a, int b) {  
3         return a + b;  
4     }  
5  
6     public static void main(String[] args) {  
7         int resultado = sumar(10, 5);  
8         System.out.println("La suma es: " + resultado);  
9     }  
10 }  
11
```

La suma es: 15

...Program finished with exit code 0
Press ENTER to exit console.

FUNCIÓN

```
5  
6 public static void main(String[] args) {  
7     int resultado = sumar(10, 5);  
8     System.out.println("La suma es: " + resultado);  
9 }  
10  
11
```

BNF

<funcion> ::= <modificadores> <tipo_retorno> <nombre_función> (parametros) {
 <declaración_variables>;
 <sentencias>;
 <contenido>;
 return <valor>;
}

FUNCIÓN

```
5  
6 public static void main(String[] args) {  
7     int resultado = sumar(10, 5);  
8     System.out.println("La suma es: " + resultado);  
9 }  
10  
11 }
```

BNF

$\langle \text{funcion_main} \rangle ::= \text{"public" "static" "void" "main" "(" "String" "[" "]" \langle \text{identificador} \rangle ")" "{"}$
 $\langle \text{declaracion_variable} \rangle ";" \langle \text{contenido} \rangle ; \}"$

$\langle \text{declaracion_variable} \rangle ::= \langle \text{tipo_primitivo} \rangle \langle \text{identificador} \rangle "=" \langle \text{valor} \rangle ";"$

$\langle \text{tipo_primitivo} \rangle ::= \text{"int"}$

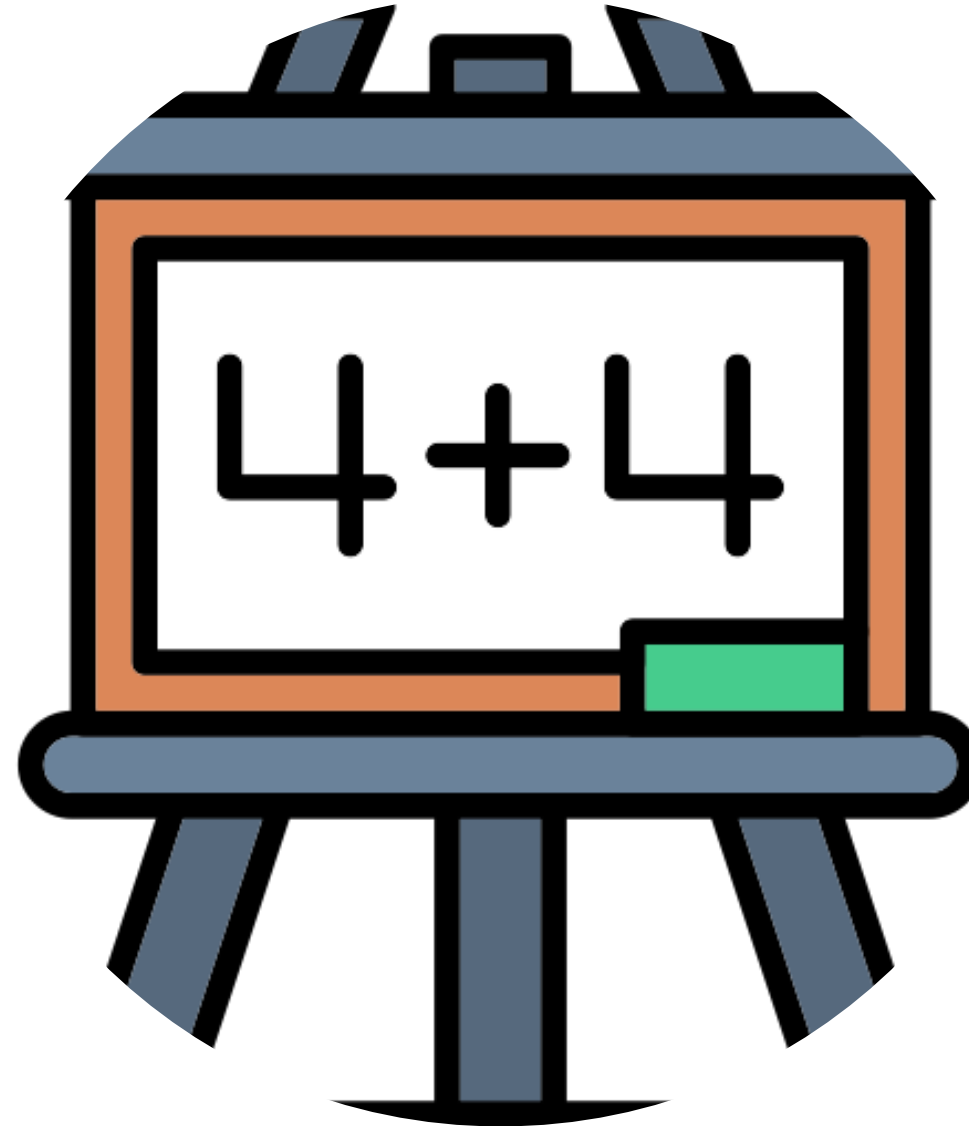
$\langle \text{valor} \rangle ::= \langle \text{identificador} \rangle "(" \langle \text{numero} \rangle "," \langle \text{numero} \rangle ")"$

$\langle \text{numero} \rangle ::= \langle \text{digito} \rangle +$

$\langle \text{identificador} \rangle ::= \langle \text{digito} \rangle \langle \text{digito} \rangle ^*$

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

EXPRESIONES
ALGEBRAICAS



ANALIZADOR



EXPRESIONES ALGEBRAÍCAS

Se utilizan en una amplia variedad de campos para modelar, resolver problemas, analizar sistemas y realizar cálculos matemáticos y computacionales.

{
 Suma
 Resta
 Multiplicación
 división

```
int a = 5;
int b = 3;
```

$\langle \text{declaracion_variables} \rangle ::= \text{int } \langle \text{variable} \rangle = \langle \text{valor} \rangle \mid \langle \text{expresion_algebraica} \rangle ;$
 $\langle \text{expresion_algebraica} \rangle ::= \langle \text{suma} \rangle \mid \langle \text{resta} \rangle \mid \langle \text{multiplicacion} \rangle \mid \langle \text{division} \rangle$
 $\langle \text{variable} \rangle ::= \langle \text{letra} \rangle \mid \langle \text{letra} \rangle \langle \text{letra} \rangle$
 $\langle \text{letra} \rangle ::= A|...|Z|a|...|z$

```
int suma = a + b;
System.out.println("La suma de " + a + " y " + b + " es " + suma);
```

$\langle \text{suma} \rangle ::= \langle \text{variable} \rangle + \langle \text{variable} \rangle ;$

```
int resta = a - b;
System.out.println("La resta de " + a + " y " + b + " es " + resta);
```

$\langle \text{resta} \rangle ::= \langle \text{variable} \rangle - \langle \text{variable} \rangle ;$

```
int multiplicacion = a * b;  
System.out.println("El producto de " + a + " y " + b + " es " + multiplicacion);
```

<multiplicacion> ::= <variable> * <variable> ;

```
int division = a / b;  
System.out.println("La división de " + a + " y " + b + " es " + division);
```

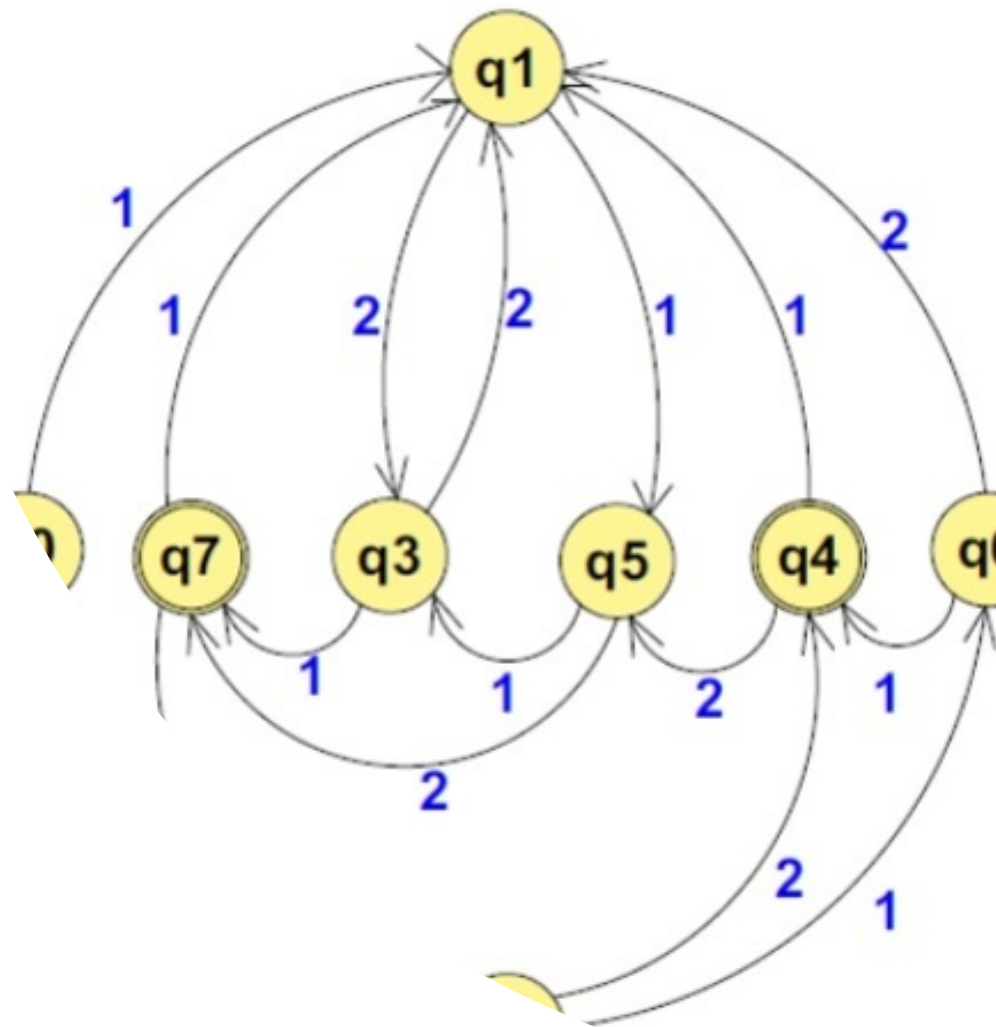
<division> ::= <variable> / <variable> ;

<resultado> ::= <evaluacion> ;

<evaluacion> ::= System.out.println(<mensaje>)

<mensaje> ::= <texto> + <variable> + <variable> + <texto> + <variable>

CONCLUSIÓN



- En conclusión, los analizadores sintácticos son herramientas poderosas que permiten comprender y procesar la estructura de los lenguajes, facilitando la comunicación entre humanos y computadoras, así como el desarrollo de una amplia gama de aplicaciones informáticas



¡MUCHAS
GRACIAS!

