



Directorio de Tokens

Palabras Reservadas

PRI1 ciclo
PRI2 si
PRI3 siNo
PRI4 leer
PRI5 imprimir
PRI6 declarar
PRI7 segun
PRI8 caso
PRI9 romper
PRI10 continuar
PRI11 regresar
PRI12 defecto
PRI13 funcion
PRI14 mientras
PRI15 hace

Caracteres especiales

@ # \$ () { } [] ` " ; | ,
CE1 CE2 CE3 CE6 CE7 CE8 CE9 CE10 CE11 CE12 CE13 CE14

Banderas

cierto PRB1
falso PRB2

Variables

variable PRV1
constante PRV2

Operadores Aritméticos

+ - * / ^ % =
OPA1 OPA2 OPA3 OPA4 OPA5 OPA6 ASIG



Operadores Relacionales

< > <= >= != ==

OPR1 OPR2 OPR3 OPR4 OPR5 OPR6

Operadores Lógicos

! || &&

OPL1 OPL2 OPL3

"cadena" CADENA

Entero CONSTANT

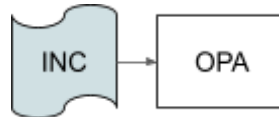
Real CONSTRE

Exponencial CONSTEX

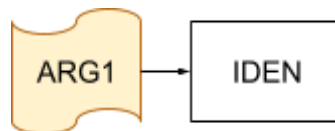
##comentario## COMENTARIO



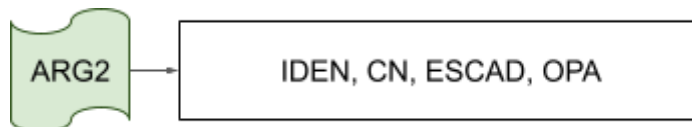
condicion → Operacion Logica , Operacion Relacional, Bandera booleana



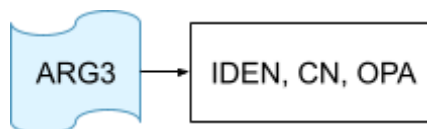
incremento → Operacion Aritmetica



argumento1 → Identificador Valido



argumento2 → Identificador Valido, Constante Numerica, Cadena, Operacion Aritmetica



argumento3 → Identificador Valido, Constante Numerica, Operacion Aritmetica



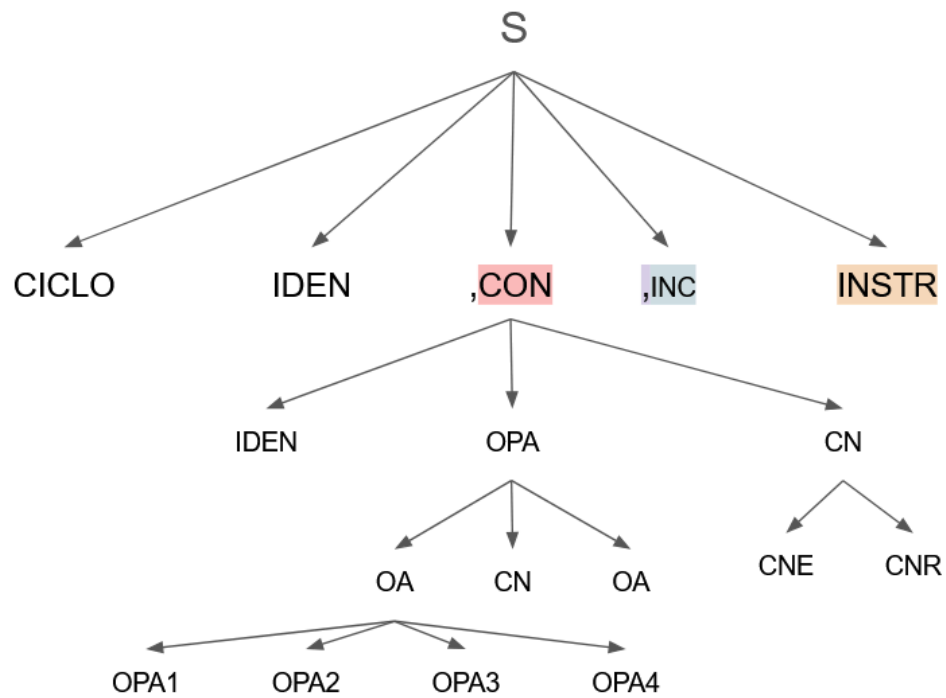
Instrucciones

Palabra Reservada PRI1	ciclo
SINTAXIS	
<pre>ciclo(inicialización; <i>condicion</i> ; <i>incremento</i>;) { [Conjunto de instrucciones] };</pre>	
EJEMPLOS	
<pre>ciclo(declarar variable x=0; x<50 x==0; x=2*x){ };</pre> <pre>ciclo(declarar variable z=1; z <100; z=5*z){ };</pre> <pre>ciclo(declarar variable x=0; cierto; x=2*x){ };</pre>	

El bucle **ciclo** (for) se utiliza para repetir un bloque de código un número específico de veces.

inicializacion → [Instrucción **declarar**]



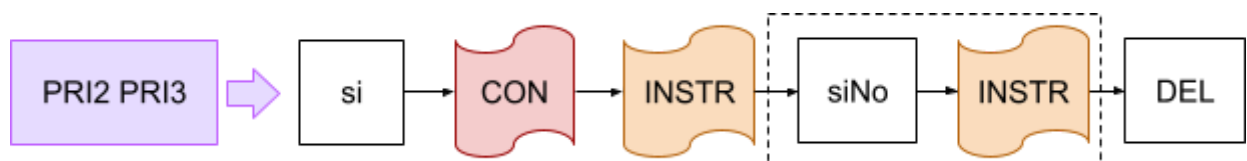


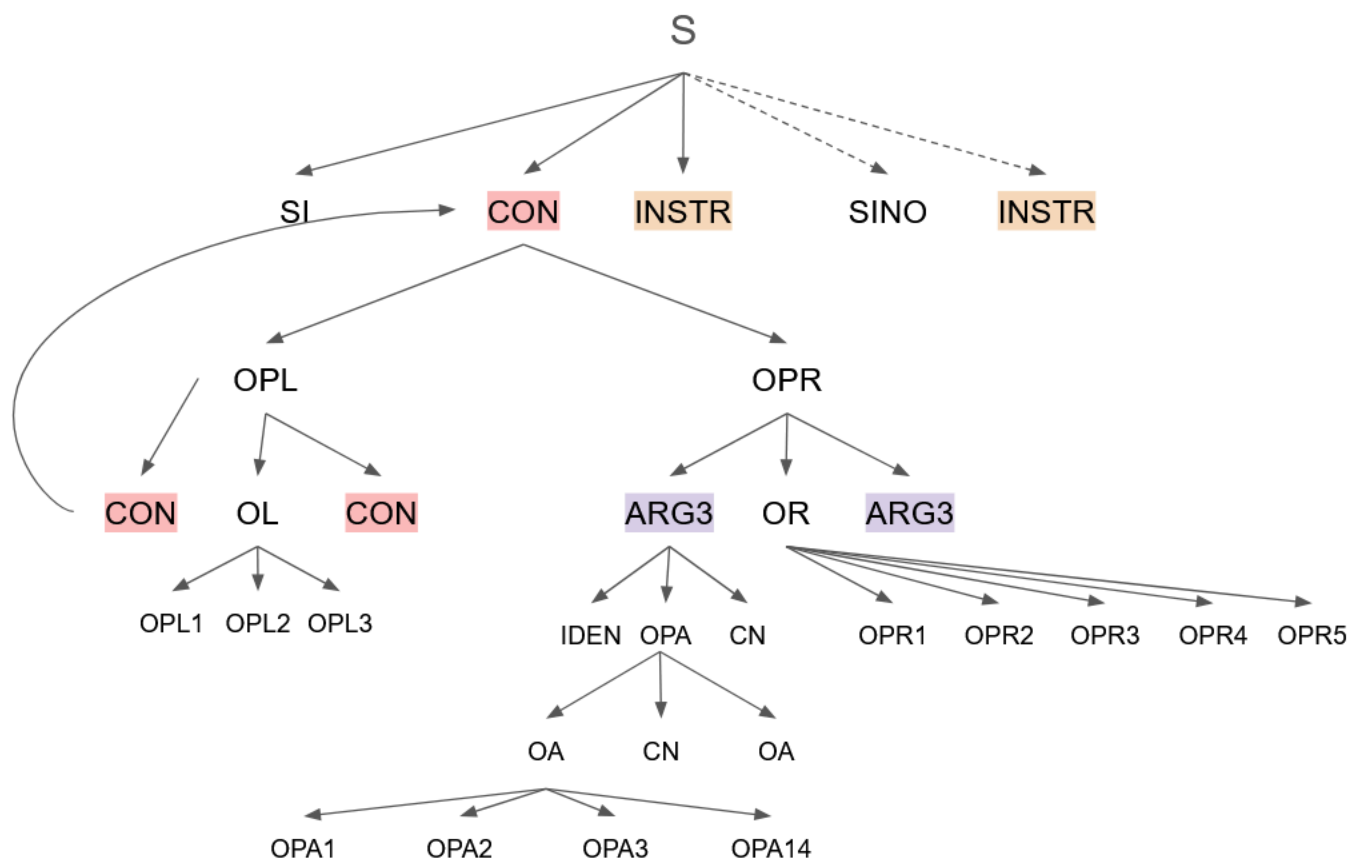
$S \rightarrow \text{PRI1 IDEN CON}$
 $\text{CON} \rightarrow \text{IDEN OPA CN}$
 $\text{OPA} \rightarrow \text{OA CN OA}$
 $\text{CN} \rightarrow \text{CNE CNR}$
 $\text{OA} \rightarrow \text{OPA1 OPA2 OPA3 OPA4}$



Palabra Reservada PRI2 PRI3	si, siNo
SINTAXIS	
<pre>si <i>condicion</i> { [Conjunto de instrucciones] } siNo { [Conjunto de instrucciones] };</pre>	
EJEMPLOS	
<pre>si x == 5 { imprimir "Es igual a 5" }; si x > 3 { imprimir "Es mayor"; } siNo { imprimir "No es mayor"; }; si x > 9 && x < 12 {imprimir "Es mayor a 9 y menor a 12" } siNo {imprimir "No es mayor a 9 y menor a 12" };</pre>	

La estructura condicional **si siNo** (if else) se utiliza para ejecutar bloques de código basados en condiciones específicas.





$S \rightarrow \text{PRI2 CON INSTR PRI3 INSTR}$

$\text{CON} \rightarrow \text{OPL OPR}$

$\text{OPL} \rightarrow \text{CON OL CON}$

$\text{OL} \rightarrow \text{OPL1 OPL2 OPL3}$

$\text{OPR} \rightarrow \text{ARG3}$

$\text{ARG3} \rightarrow \text{IDEN OPA CN}$

$\text{OR} \rightarrow \text{OPR1 OPR2 OPR3 OPR4 OPR5}$

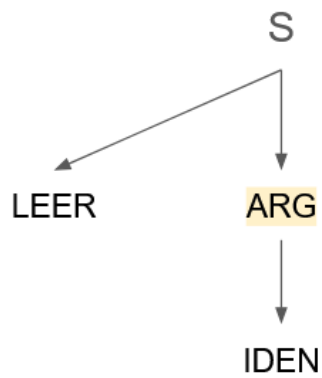
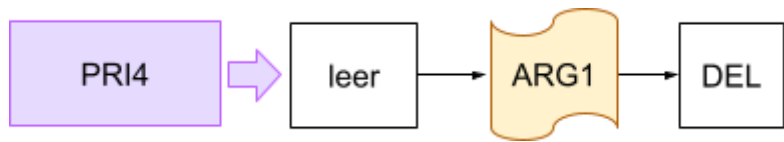
$\text{OPA} \rightarrow \text{OA CN OA}$

$\text{OA} \rightarrow \text{OPA1 OPA2 OPA3 OPA4}$



Palabra Reservada PRI4	leer
SINTAXIS	
<code>leer <i>argumento1</i>;</code>	
EJEMPLOS	
<code>leer x; leer edad; leer promedio;</code>	

leer (read) se utiliza para leer datos del usuario.



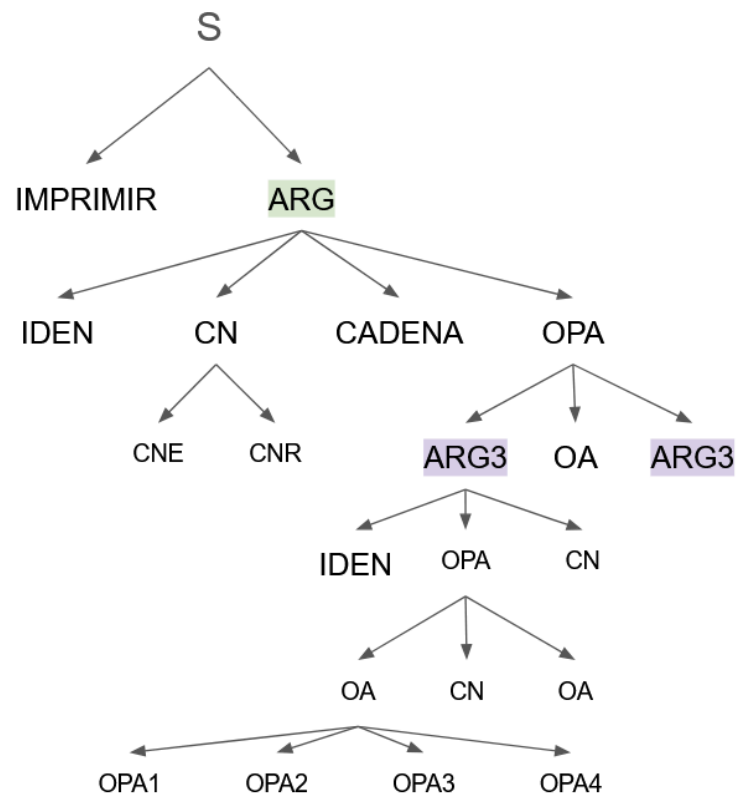
$S \rightarrow \text{PRI4 ARG}$
 $\text{ARG} \rightarrow \text{IDEN}$



Palabra Reservada PRI5	imprimir
SINTAXIS	
<code>imprimir <i>argumento2</i>;</code>	
EJEMPLOS	
<code>imprimir x; imprimir 3+5; imprimir "Texto";</code>	

imprimir (print) se utiliza para mostrar resultados en la pantalla.



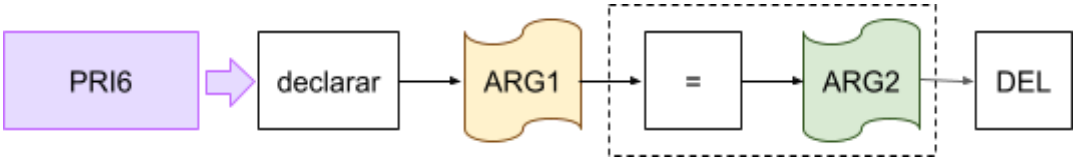


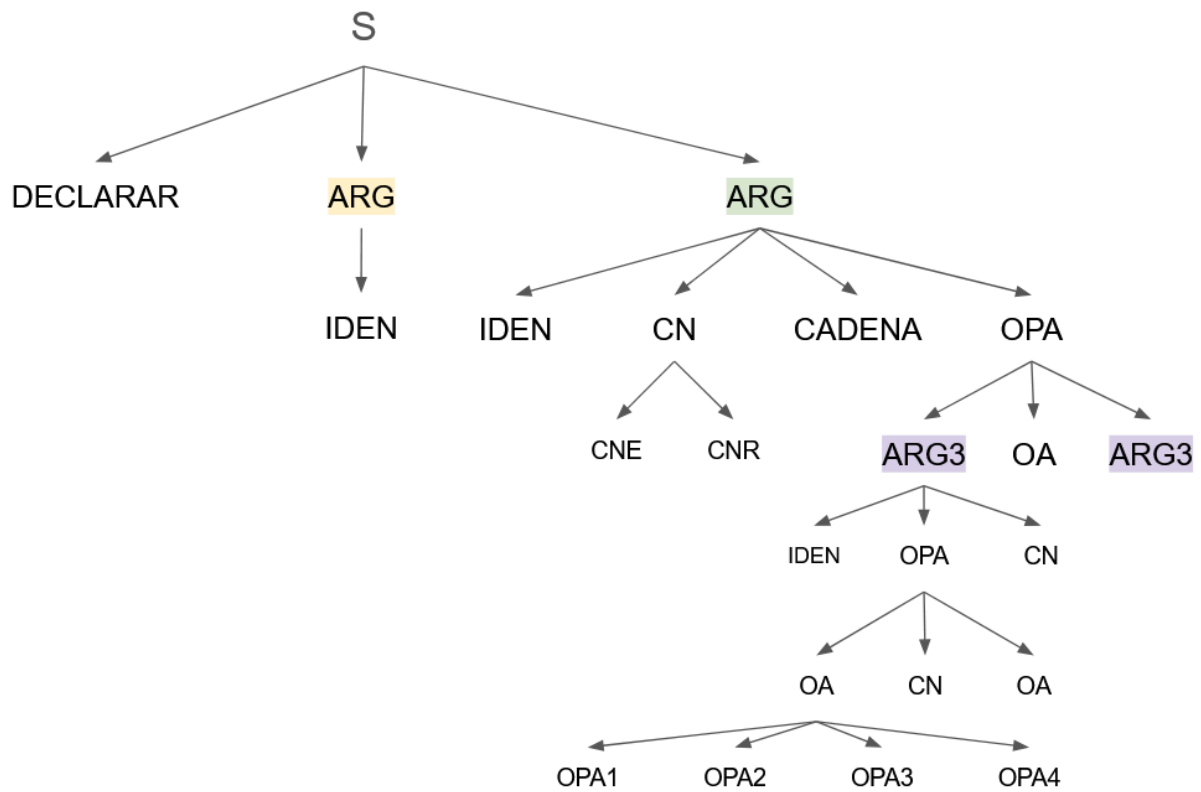
$S \rightarrow \text{PRI5 ARG}$
 $\text{ARG} \rightarrow \text{IDEN CN ESCAD OPA}$
 $\text{CN} \rightarrow \text{CNR CNE}$
 $\text{OPA} \rightarrow \text{ARG3 OA ARG3}$
 $\text{ARG3} \rightarrow \text{IDEN OPA CN}$
 $\text{OPA} \rightarrow \text{OA CN OA}$
 $\text{OA} \rightarrow \text{OPA1 OPA2 OPA3 OPA4}$



Palabra Reservada PRI6	declarar
SINTAXIS	
<pre>declarar variable <i>argumento1</i>; declarar variable <i>argumento1</i> = <i>argumento2</i> ;</pre>	
EJEMPLOS	
<pre>declarar variable x; declarar variable suma = x + y; declarar variable edad = 23;</pre>	

La palabra clave **declarar** (let) se utiliza para declarar variables.





$S \rightarrow \text{PRI6 ARG}$
 $\text{ARG} \rightarrow \text{IDEN CN ESCAD OPA}$
 $\text{CN} \rightarrow \text{CNR CNE}$
 $\text{OPA} \rightarrow \text{ARG3 OA ARG3}$
 $\text{ARG3} \rightarrow \text{IDEN OPA CN}$
 $\text{OPA} \rightarrow \text{OA CN OA}$
 $\text{OA} \rightarrow \text{OPA1 OPA2 OPA3 OPA4}$



Palabras Reservadas PRI7 PRI8

segun, caso

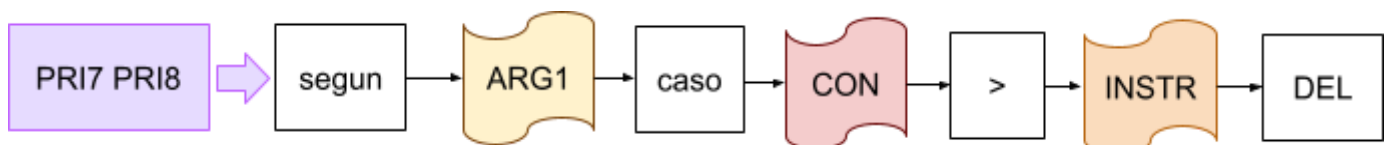
SINTAXIS

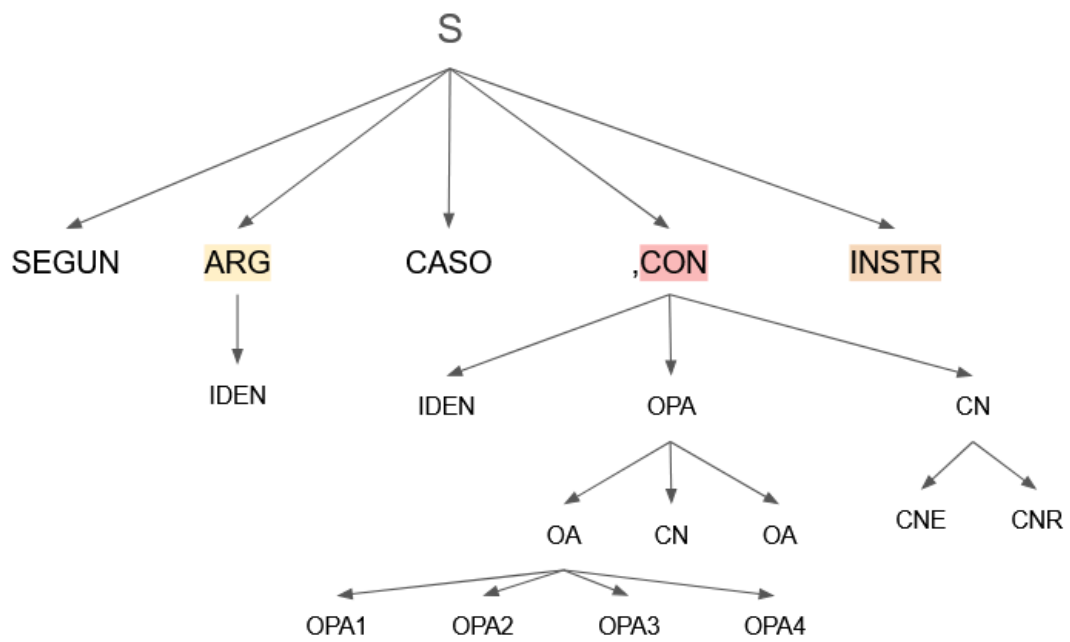
```
segun argumento1 {  
  caso condicion >  
    [Conjunto de instrucciones]  
  caso condicion >  
    [Conjunto de instrucciones]  
};
```

EJEMPLOS

```
segun dia  
{  
  caso 1 >  
    imprimir "lunes";  
    romper;  
  caso 2 >  
    imprimir "lunes";  
    romper;  
};
```

La estructura **segun caso** (switch case) se utiliza para ejecutar uno de varios bloques de código según el valor de una expresión





$S \rightarrow \text{PRI7 ARG PRI8 CON INSTR}$

$\text{ARG} \rightarrow \text{IDEN}$

$\text{CON} \rightarrow \text{OPL OPR}$

$\text{OPL} \rightarrow \text{CON OL CON}$

$\text{OL} \rightarrow \text{OPL1 OPL2 OPL3}$

$\text{OPR} \rightarrow \text{ARG3 OR ARG3}$

$\text{OR} \rightarrow \text{OPR1 OPR2 OPR3 OPR4}$



Palabras Reservadas PRI9 PRI10 PRI11

romper, continuar, regresar

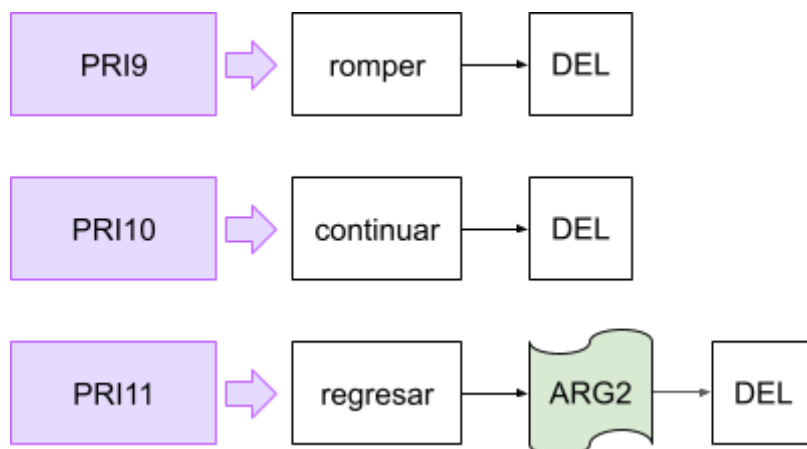
SINTAXIS

```
romper;  
continuar;  
regresar;  
regresar argumento2;
```

EJEMPLOS

```
romper;  
continuar;  
regresar;  
  
ciclo(declarar variable i=0; i <10; i=i+1 ){  
    si i == 5 {romper;}  
};  
  
ciclo(declarar variable i=0; i <15; i=i+1 ){  
    si i == 7 {continuar;}  
};  
  
ciclo(declarar variable i=0; i <15; i=i+1 ){  
    si i == 7 {regresar i;}  
};
```

Las palabras clave **romper** **continuar** y **regresar** (break, continue y return) son usadas para controlar el flujo de ejecución dentro de bucles y funciones.



Palabras Reservadas PRI14 PRI15

mientras , hacer

SINTAXIS

```
mientras condicion {  
  [Conjunto de instrucciones]  
};
```

```
hacer {  
  [Conjunto de instrucciones]  
}mientras condicion ;
```

EJEMPLOS

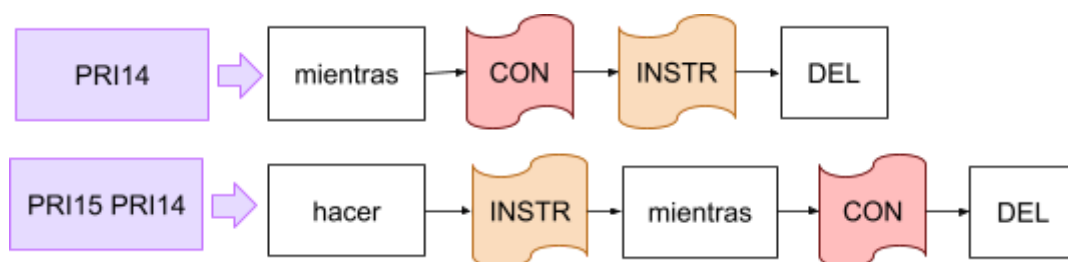
```
mientras i <= 35 {  
  leer nota;  
  sumatorio = 1 * (sumatorio + nota);  
};
```

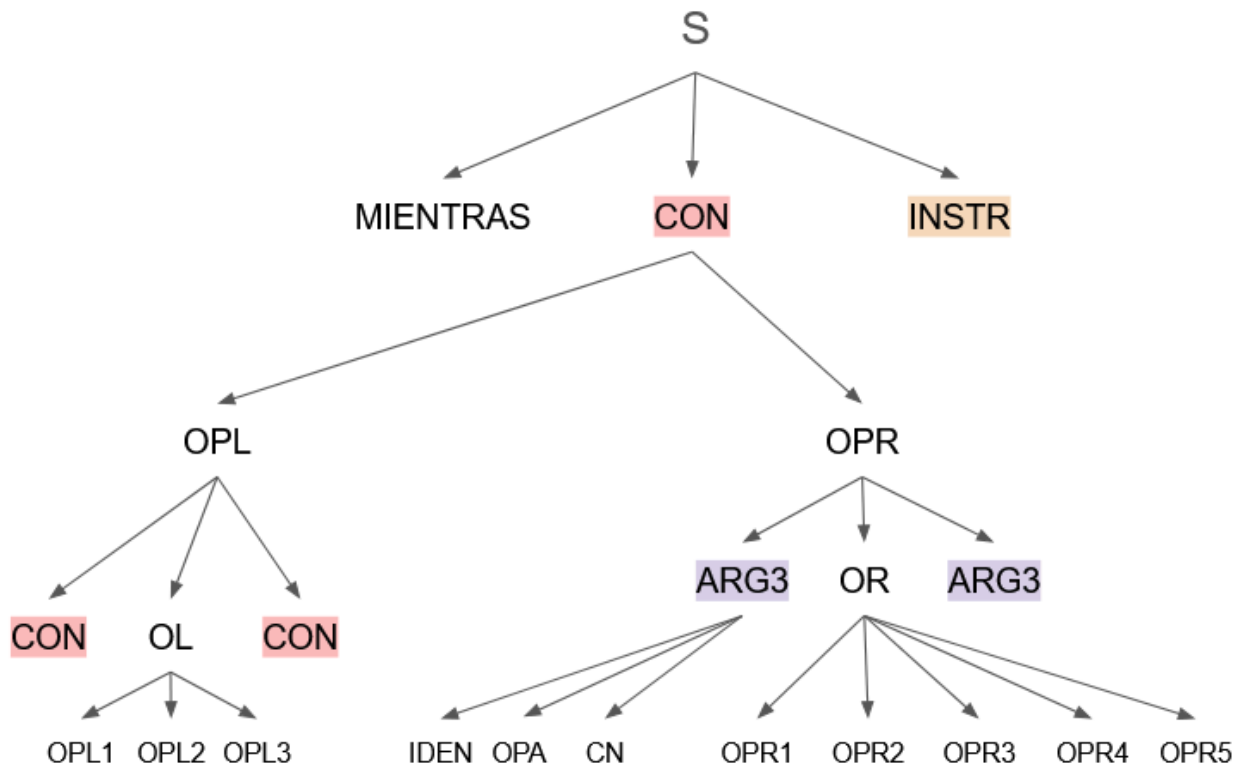
```
mientras z < 70{  
  imprimir z;  
};
```

```
hacer {  
  leer nota;  
  sumatorio = 1 * (sumatorio + nota);  
} mientras i <= 35;
```

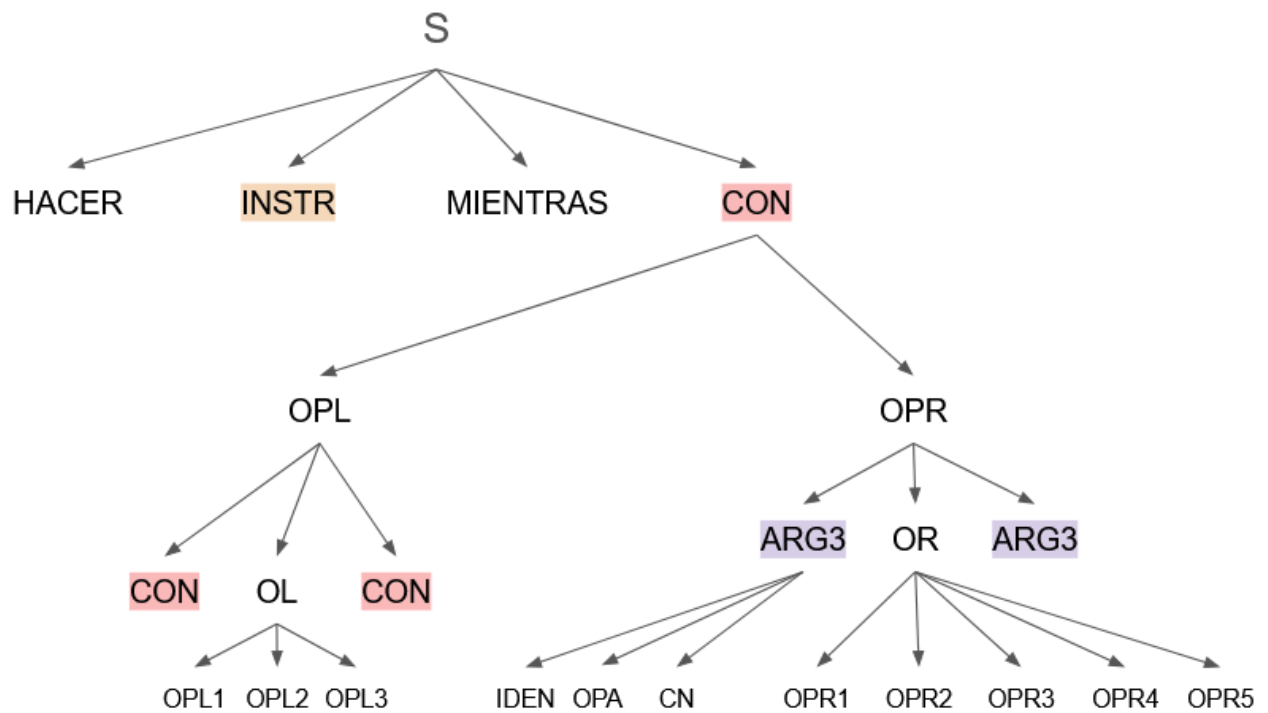
El bucle **mientras** (while) se utiliza para ejecutar un bloque de código repetidamente mientras se cumple una condición específica.

El bucle **hacer mientras** (do while) es una estructura de control que ejecuta un bloque de código al menos una vez, y luego sigue ejecutando el bloque repetidamente mientras una condición específica sea verdadera.





S → PRI14 CON INSTR
CON → OPL OPR
OPL → CON OL CON
OL → OPL1 OPL2 OPL3
OPR → ARG3 OR ARG3
ARG3 → IDEN OPA CN
OR → OPR1 OPR2 OPR3 OPR4



$S \rightarrow \text{PRI14 INSTR PRI15 CON}$
 $\text{CON} \rightarrow \text{OPL OPR}$
 $\text{OPL} \rightarrow \text{CON OL CON}$
 $\text{OL} \rightarrow \text{OPL1 OPL2 OPL3}$
 $\text{OPR} \rightarrow \text{ARG3 OR ARG3}$
 $\text{ARG3} \rightarrow \text{IDEN OPA CN}$
 $\text{OR} \rightarrow \text{OPR1 OPR2 OPR3 OPR4}$

