



---

## EJERCICIO 1

---

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

$P = \{$

programa  $\rightarrow$  declaraciones sentencias

declaraciones  $\rightarrow$  declaraciones declaracion | declaracion

declaracion  $\rightarrow$  tipo lista\_var;

tipo  $\rightarrow$  *int* | *float*

lista\_var  $\rightarrow$  lista\_var, *identificador* | *identificador*

sentencias  $\rightarrow$  sentencias sentencia | sentencia

sentencia  $\rightarrow$  *identificador* = expresion; |

*if* (expresion) sentencias *else* sentencias | *while* (expresion) sentencias

expresion  $\rightarrow$  expresion + expresion | expresion - expresion |

expresion \* expresion | expresion / expresion | *identificador* | *numero*

expresion  $\rightarrow$  (expresion)

}

$N = \{\text{programa, declaraciones, declaracion, tipo, lista\_var, sentencias,}$   
sentencia, expresion}

$\Sigma = \{\textit{int}, \textit{float}, \textit{identificador}, \textit{numero}, \textit{if}, \textit{else}, \textit{while}, +, -, *, /, =,$   
;, ,, (, )}

$S = \text{program}$

---

## EJERCICIO 2: ELIMINACIÓN DE LA AMBIGÜEDAD

---

$\text{sentencia\_completa} \rightarrow \text{if}(\text{expresion}) \text{sentencia\_completa} \text{ else}$   
 $\text{sentencia\_completa} \mid \text{while}(\text{expresion}) \text{sentencias} \mid$   
 $\text{identificador} = \text{expresion};$   
 $\text{sentencia\_incompleta} \rightarrow \text{if}(\text{expresion}) \text{sentencia\_completa} \mid$   
 $\text{while}(\text{expresion}) \text{sentencias};$

---

## EJERCICIO 3: ELIMINACIÓN DE LA RECURSIVIDAD IZQUIERDA

---

$\text{declaraciones} \rightarrow \text{declaracion} \text{declaraciones}'$   
 $\text{declaraciones}' \rightarrow \text{declaracion} \text{declaraciones}' \mid \epsilon$

$\text{sentencias} \rightarrow \text{sentencia} \text{sentencias}'$   
 $\text{sentencias}' \rightarrow \text{sentencia} \text{sentencias}' \mid \epsilon$

---

## EJERCICIO 4: FACTORIZACIÓN IZQUIERDA

---

$\text{expresion} \rightarrow \text{expresion\_suma}$   
 $\text{expresion\_suma} \rightarrow \text{expresion\_suma} + \text{expresion\_producto} \mid$   
 $\text{expresion\_suma} - \text{expresion\_producto} \mid \text{expresion\_producto}$   
 $\text{expresion\_producto} \rightarrow \text{expresion\_producto} * \text{factor} \mid$   
 $\text{expresion\_producto} / \text{factor} \mid \text{factor}$   
 $\text{factor} \rightarrow \text{identificador} \mid \text{numero} \mid (\text{expresion})$

## EJERCICIO 5

$P = \{$   
 programa  $\rightarrow$  declaraciones sentencias  
 declaraciones  $\rightarrow$  declaracion declaraciones'  
 declaraciones'  $\rightarrow$  declaracion declaraciones'  $\mid \epsilon$   
 declaracion  $\rightarrow$  tipo lista\_var;  
 tipo  $\rightarrow$  *int*  $\mid$  *float*  
 lista\_var  $\rightarrow$  lista\_var, *identificador*  $\mid$  *identificador*  
 sentencias  $\rightarrow$  sentencia sentencias'  
 sentencias'  $\rightarrow$  sentencia sentencias'  $\mid \epsilon$   
  
 sentencia  $\rightarrow$  sentencia\_completa  $\mid$  sentencia\_incompleta  
 sentencia\_completa  $\rightarrow$  *if* (expresion) sentencia\_completa *else*  
 sentencia\_completa  $\mid$  *while* (expresion) sentencias  $\mid$   
 identificador = expresion;  
 sentencia\_incompleta  $\rightarrow$  *if* (expresion) sentencia\_completa  $\mid$   
*while* (expresion) sentencias;  
  
 expresion  $\rightarrow$  expresion + expresion  $\mid$  expresion - expresion  $\mid$   
 expresion \* expresion  $\mid$  expresion / expresion  $\mid$  *identificador*  $\mid$  *numero*  
 expresion  $\rightarrow$  (expresion)  
 $\}$

$N = \{\text{programa, declaraciones, declaraciones', declaracion, tipo,}$   
 lista\_var, sentencias, sentencias', sentencia, sentencia\_completa,  
 sentencia\_incompleta, expresion $\}$