

Sintáctico: se encarga de que los tokens estén en el orden adecuado

REGLAS:

Las expresiones usan notación infija  
El número de paréntesis abiertos es siempre igual al de paréntesis cerrados.

Mayúscula  $\rightarrow$  TOKEN  $\rightarrow \mathbb{Z}$   
minúscula  $\rightarrow$  no terminal  $\rightarrow a$

Lenguaje: el lenguaje que genera una gramática es el conjunto de todas sus sentencias.  
Conjunto de todos los posibles programas

$$S \rightarrow S + S$$
$$S \rightarrow X$$
$$S \rightarrow X$$
$$S \rightarrow S' + S \rightarrow X + X$$
$$S \rightarrow \underbrace{S' + S}_{\text{genera}} \rightarrow \underline{S + S} + S$$

genera  $\hookrightarrow X + \dots + X$

$$s \rightarrow X \cancel{mt} + X$$
$$mt \rightarrow + X \overset{?}{\cancel{mt}}$$
$$\textcircled{mt} \rightarrow \lambda$$
$$S \rightarrow X m + \rightarrow X$$
$$S \rightarrow X m + \rightarrow X + X m +$$

SON GRAMÁTICAS  
EQUIVALENTES

$\rightarrow$  Dado cualquier lenguaje, existen infinitas gramáticas equivalentes

Queir n es una sentencia valida:

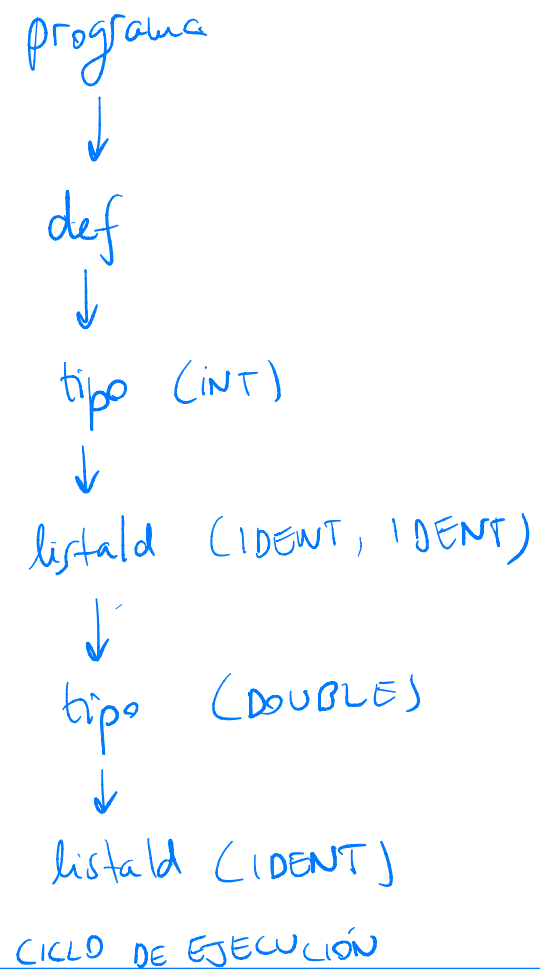
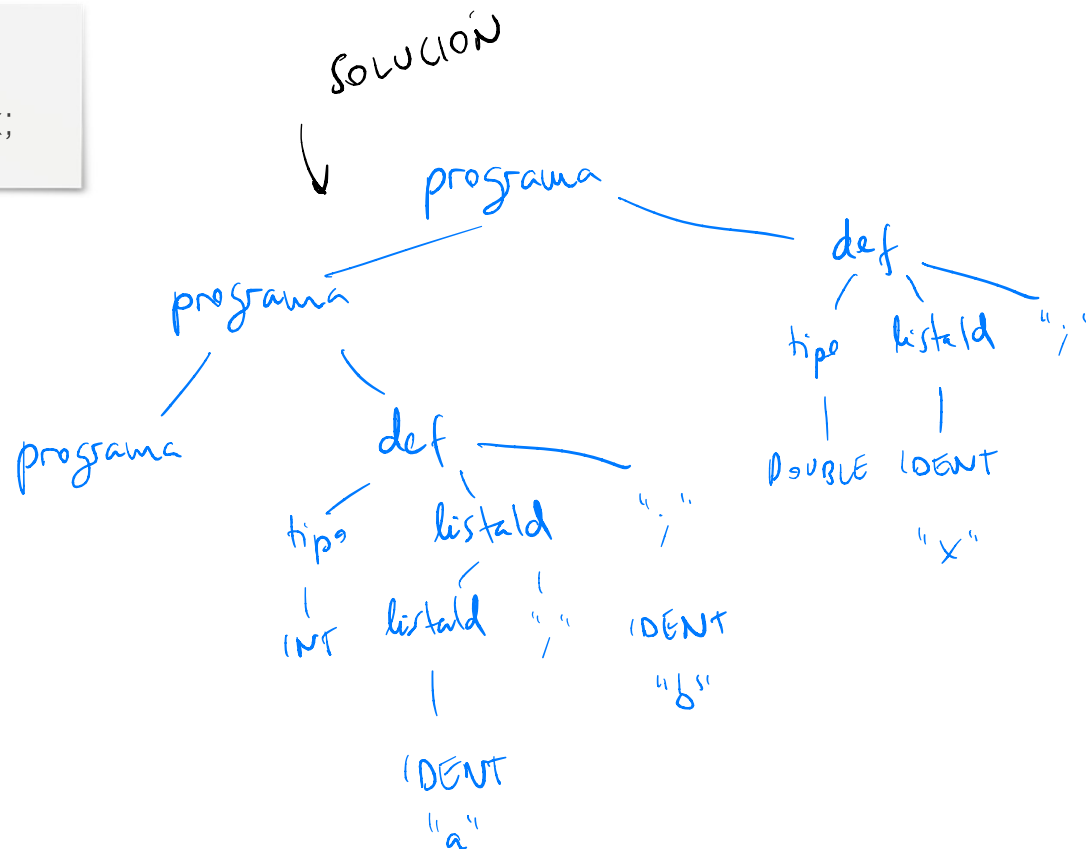
**programa**  $\rightarrow$  programa def |  $\lambda$

**def**  $\rightarrow$  tipo listald ';

**tipo**  $\rightarrow$  INT | DOUBLE

**listald**  $\rightarrow$  IDENT | listald ',' IDENT

int a, b;  
double x;



# TIPOS DE ALGORITMOS

## Dirección

▼ descendente

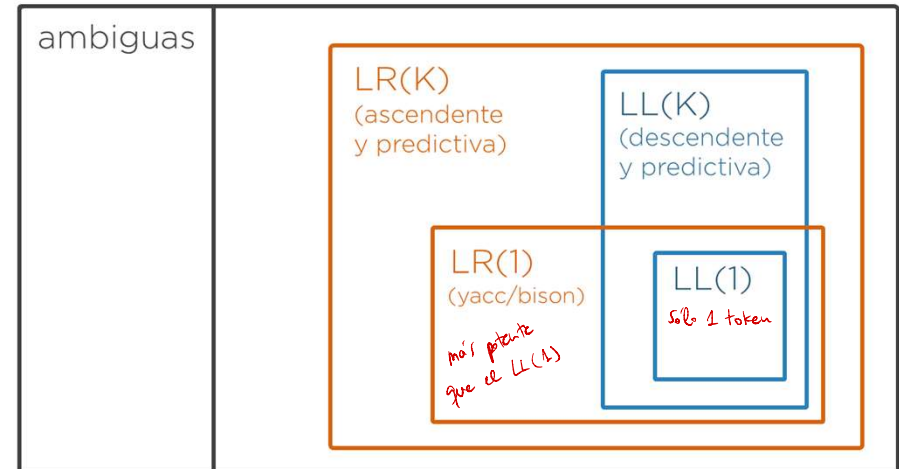
▲ ascendente

## Selección

🔄 backtracking

🎯 predictivo

# TIPOS DE GRAMÁTICAS



**prog** → decl decl

**decl** → variables | typedef

**variables** → VAR ID restolds

**restolds** → ',' ID restolds  
| ':' tipo

**tipo** → INT | REAL

**typedef** → TYPE ID restolds

```
• void prog ( ) {  
    decl ( );  
    decl ( );  
}  
• void decl ( ) {  
    if ( token.getType == VAR.ID )  
        variables ( );  
    else  
        typedef ( );  
}  
• void variables ( ) {  
    restolds ( );  
}  
• void restolds ( ) {
```

```
• void tipo ( ) {  
    if ( token.getType == INT ) {  
        int ( );  
    } else  
        double ( );  
}  
• void typedef ( ) {  
    restolds ( );  
}
```

↑  
está mal  
mirar apuntes