



ESCUELA DE
INGENIERÍA EN CIENCIAS Y SISTEMAS
FACULTAD DE INGENIERÍA
UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



Día, Fecha:	Sábado,
Hora de inicio:	21/10/2023 07:10

Lenguajes Formales y de Programación [B-]

Elder Anibal Pum Rojas

CLASE 9

LABORATORIO LFP "B-"

Aux. Elder Pum



Agenda

- Anuncios
- Proyecto 2
 - Dudas Proyecto 2
- Clase Gramáticas Tipo 2
 - Ambigüedad
 - Recursividad
 - Lectura de PDF de Ayuda en UEDI





Anuncios

Anuncios

- Examen Final de Laboratorio a realizarse la otra semana



¿DUDAS?

¿DUDAS?



Gramáticas Tipo 2

Expr \rightarrow *expr op expr*

expr \rightarrow (*expr*)

expr \rightarrow -*expr*

expr \rightarrow *id*

op \rightarrow +

OP \rightarrow -

op \rightarrow *

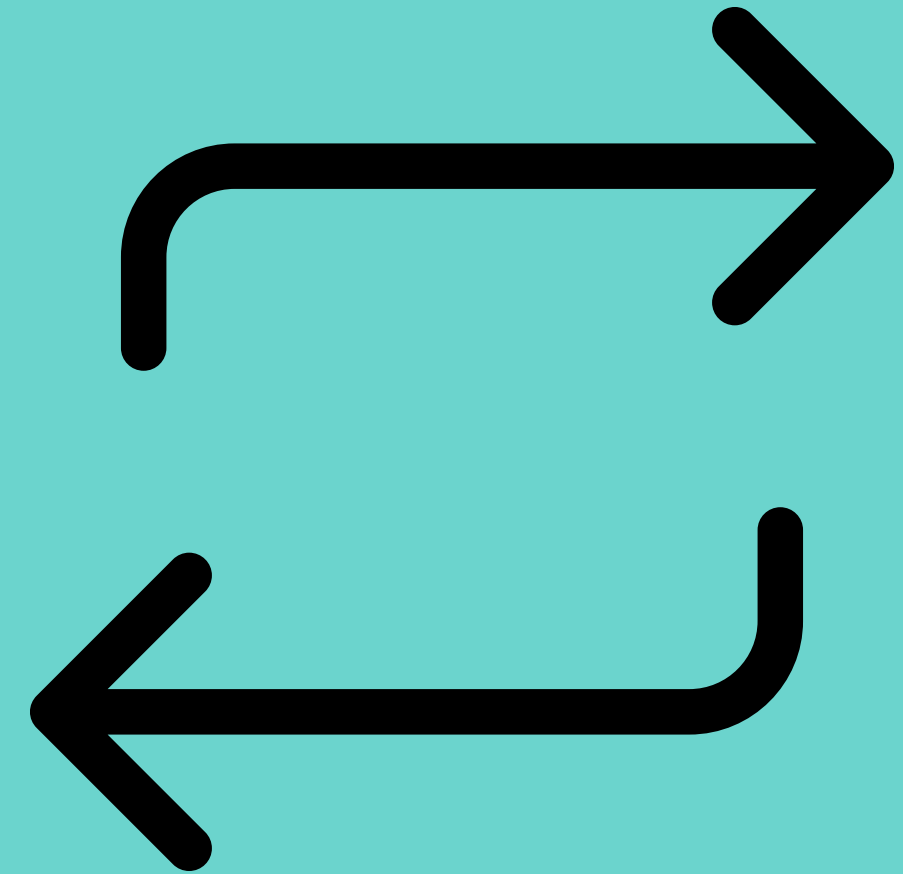
op \rightarrow /

op \rightarrow |



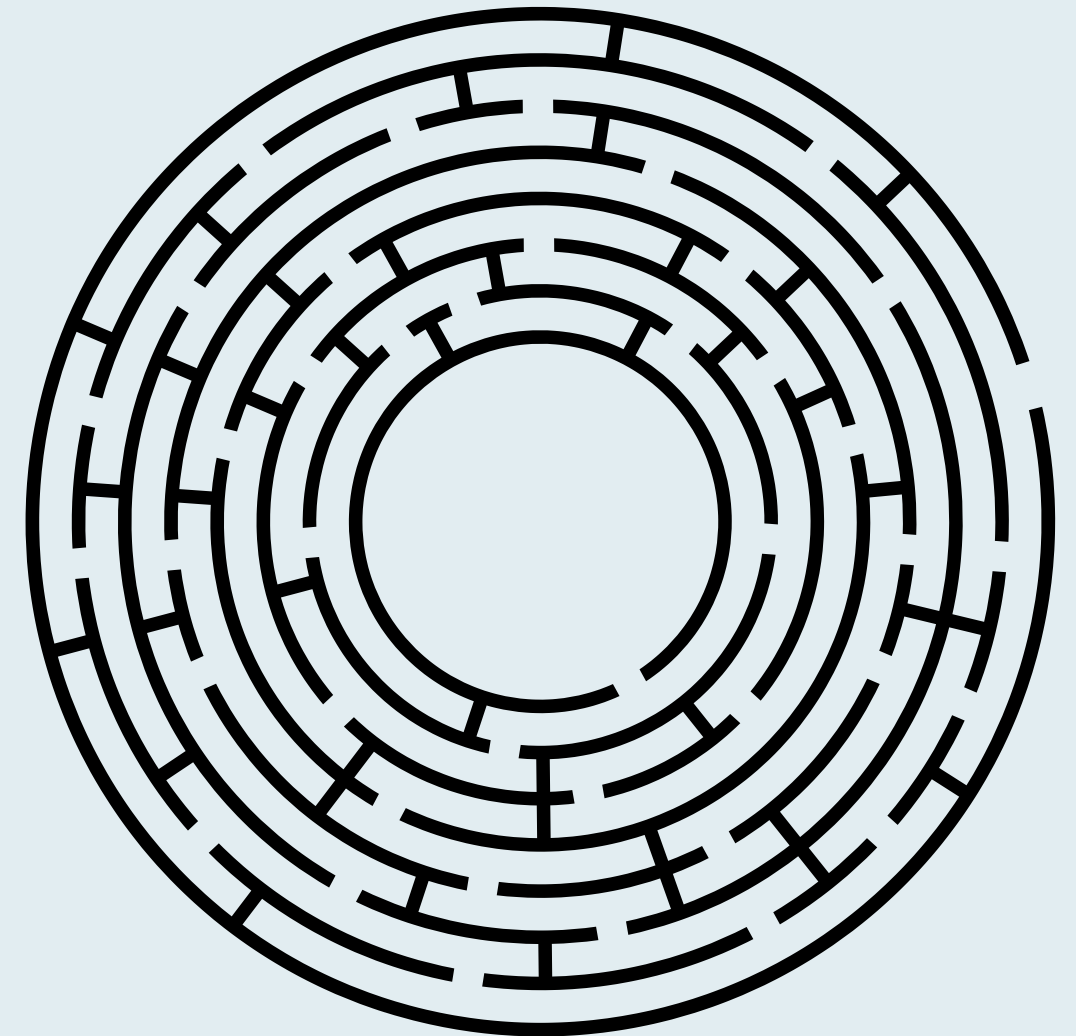
Ambigüedad

Se dice que una gramática que produce más de un árbol de análisis sintáctico para alguna frase es ambigua, o, dicho de otro modo, una gramática ambigua es la que produce más de una derivación por la izquierda o por la derecha para la misma frase.



Recursividad

- En matemáticas se da el nombre de recursión a la técnica consistente en definir una función en términos de sí misma. Puesto que en C una función puede llamar a otras funciones, se permite que una función también pueda llamarse a sí misma.
-
- Toda función definida recursivamente debe contener al menos una definición explícita para alguno de sus argumentos. De no ser así la función puede caer en un bucle infinito.



Recursividad




CÓMO FUNCIONA LA RECURSIVIDAD

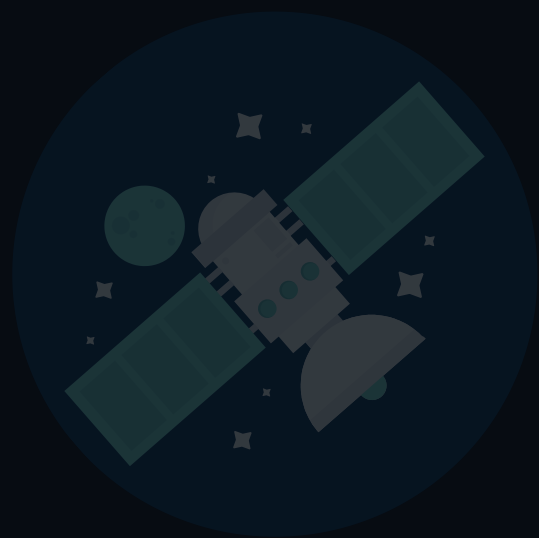


ALGORITMOS Y ESTRUCTURAS DE DATOS

RECURSIVIDAD



Watch on  YouTube



¿DUDAS?

¿DUDAS?



Tipos de Recursividad



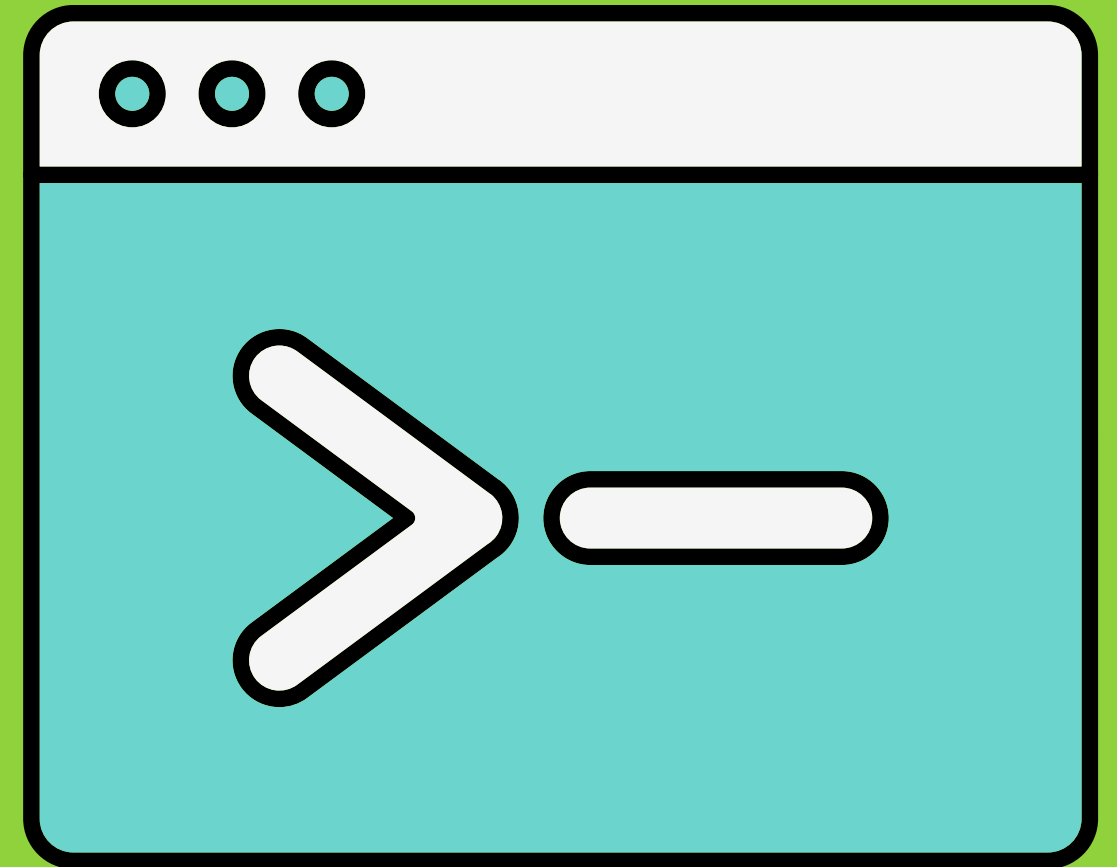
Por la Derecha



Por la Izquierda

Por la Derecha

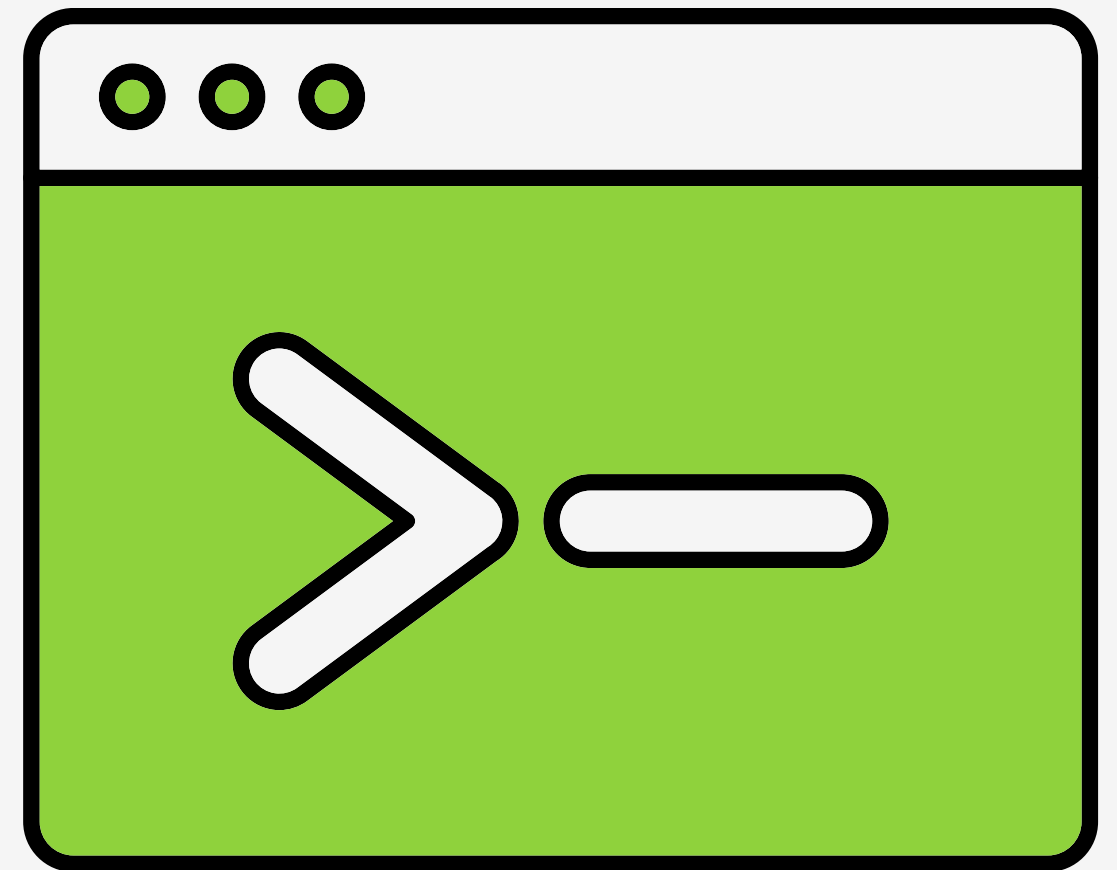
Sean A, X los no terminales y c, b, ϵ los terminales.

$$A \rightarrow c X$$
$$X \rightarrow bX$$
$$| \epsilon$$


Por la Izquierda

Sea A el único no terminal y c, b los terminales.

$$\begin{array}{c} A \rightarrow Ab \\ |c \end{array}$$



Regla para Eliminar Recursividad por la Izquierda

$$A \rightarrow bA'$$

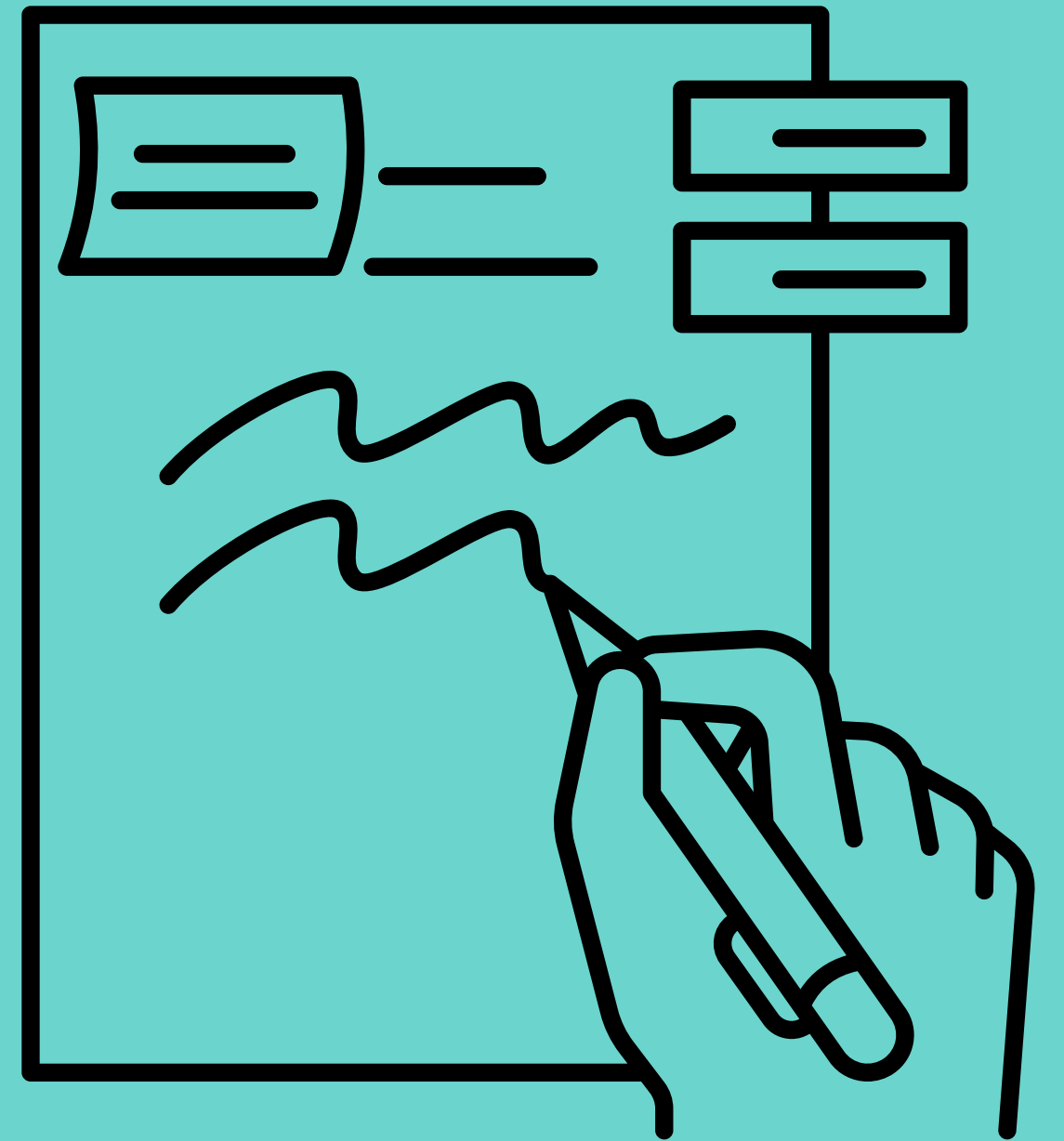
$$A' \rightarrow cA'$$

$|\epsilon$



Notación BNF

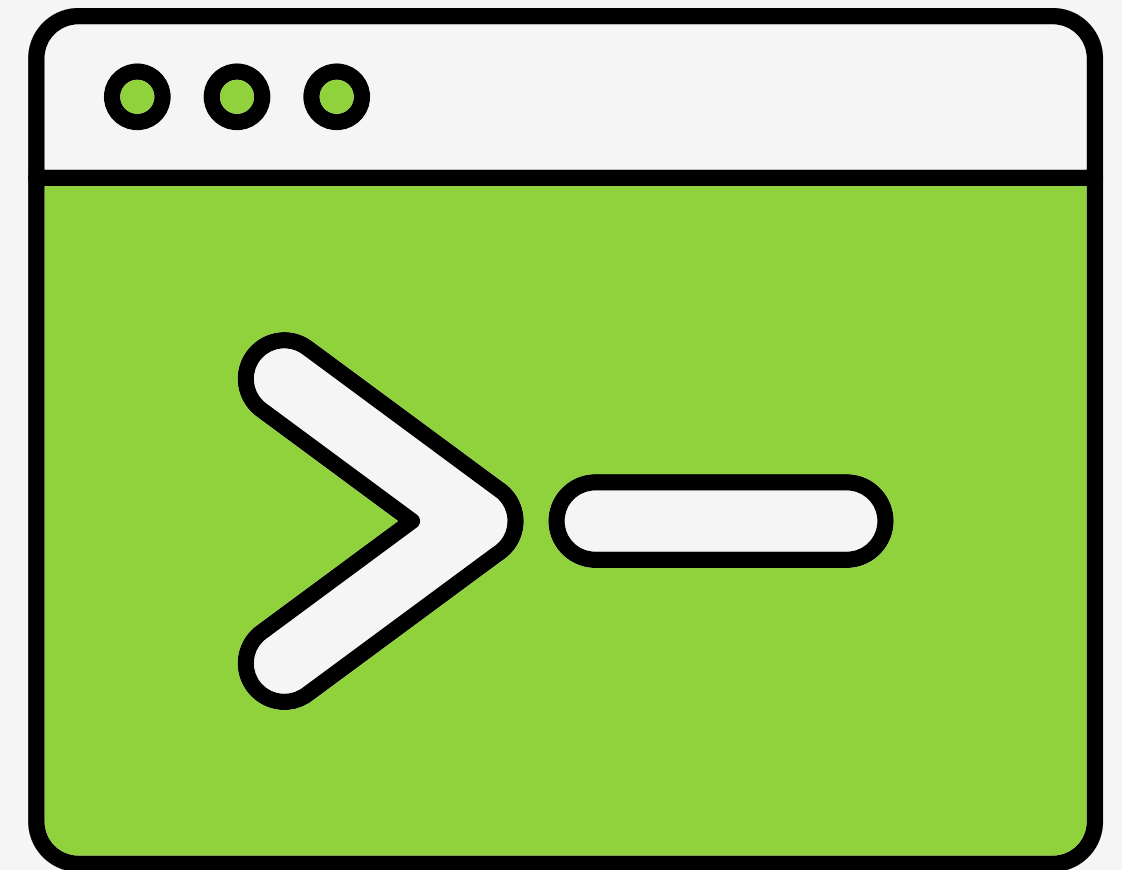
BNF (Backus-Naur Form) es una notación utilizada para describir gramáticas formales. Se compone de reglas de producción que indican cómo se deben combinar los símbolos para formar cadenas válidas en un lenguaje. Cada regla tiene una forma de "símbolo no terminal \rightarrow símbolos terminales y/o no terminales".

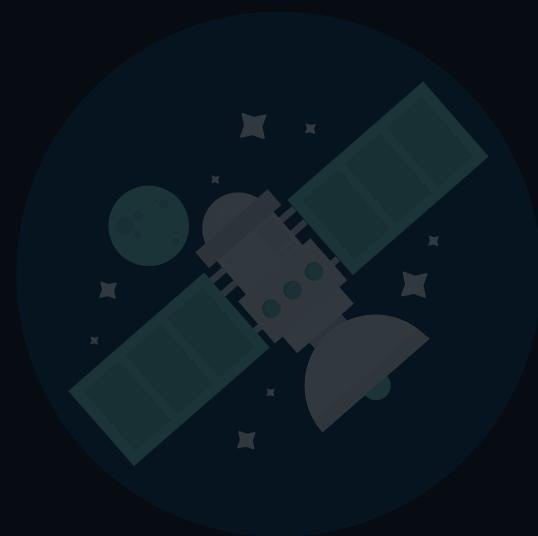


Ejemplo de BNF

Supongamos que queremos definir la sintaxis de una calculadora sencilla que permite sumar y restar números enteros.

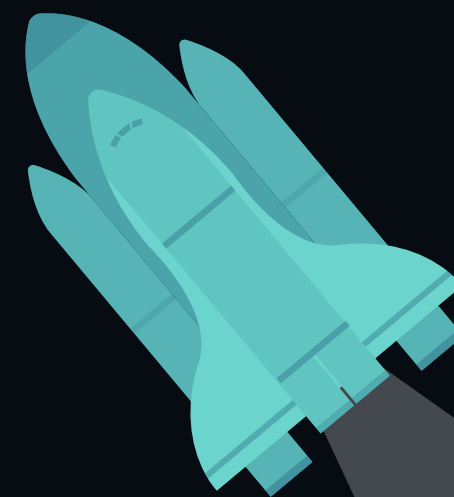
```
Expr ::= Expr + Term
      | Expr - Term
      | Term
Term  ::= Term * Factor
      | Term / Factor
      | Factor
Factor ::= ( Expr )
        | Num
Num    ::= 0 | 1 | 2 | 3 | ... | 9
```





¿DUDAS?

¿DUDAS?



Gracias por asistir

¿Dudas o preguntas?



LABORATORIO LENGUAJES FORMALES Y DE
PROGRAMACION