



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



Día, Fecha:	Sábado,
Hora de inicio:	02/09/2023 07:10

Lenguajes Formales y de Programación “B-”

Elder Anibal Pum Rojas

CLASE 4

LABORATORIO LFP "B-"

Aux. Elder Pum



Agenda

- Anuncios
- Dudas
 - Solución de dudas general
- Clase
 - Definición y Función del Analizador Léxico
 - Patrones, Tokens y Lexemas
 - Errores Léxicos
 - Operadores entre lenguajes
 - Expresiones Regulares
- Tarea 2
- Parte Práctica
 - Python





Anuncios

Anuncios

Tarea No. 2.

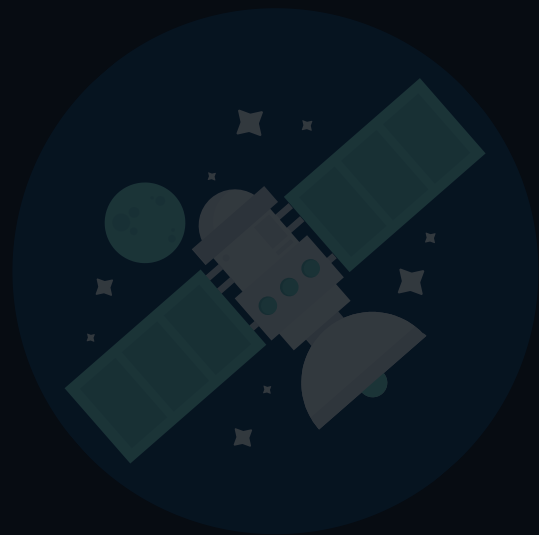
Recordar el formato de entrega para evitar penalizaciones.



Anuncios

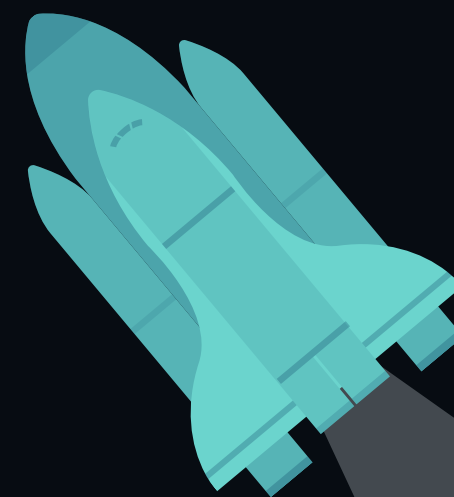
Anuncios

Evaluación Docente en el Portal DTT.
10 minutos para realizarla y después
continuaremos con nuestra clase
normalmente.



¿DUDAS?

¿DUDAS?



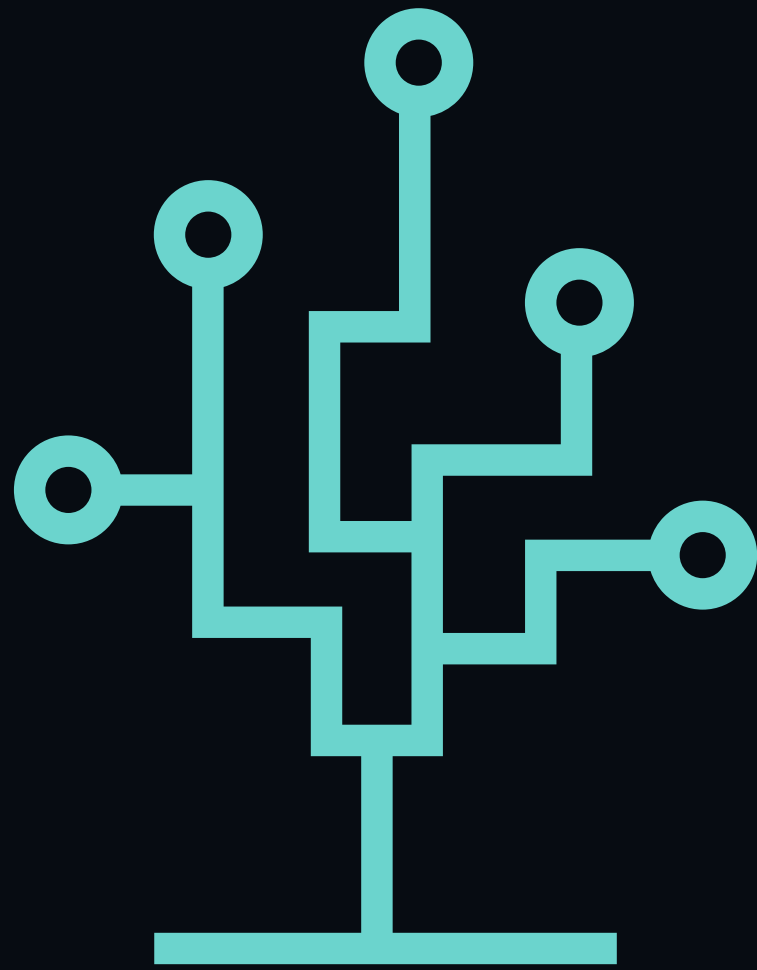
Analizador Léxico

El analizador léxico es la primera fase de un compilador.

Su principal función consiste en leer los caracteres de entrada y elaborar como salida una secuencia de componentes léxicos que utiliza el analizador sintáctico para hacer el análisis.



Analizador Léxico



Patrones



Tokens

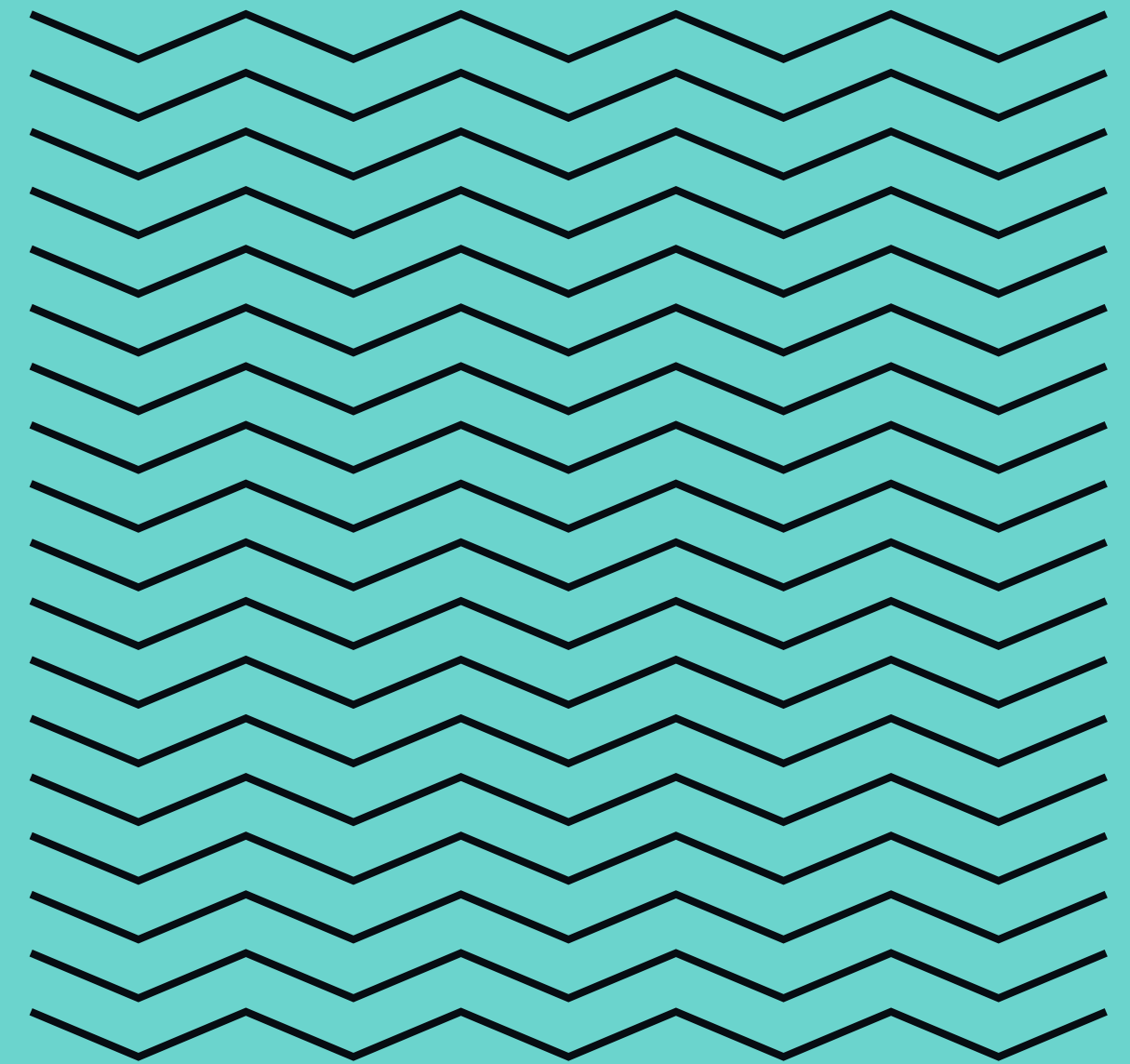


Lexemas

Patrones

if, letra seguida de letras y números

Es una descripción de la forma que pueden tomar los lexemas de un token. En el caso de una palabra clave como token, el patrón es sólo la secuencia de caracteres que forman la palabra clave.



Tokens

CADENA, ENTEROS, DECIMAL

Es un par que consiste en un nombre y un valor de atributo opcional. El nombre del token es un símbolo abstracto que representa un tipo de unidad léxica; por ejemplo: CADENA, ENTEROS, DECIMAL.



Lexema

variable1, n1, var1

Es una secuencia de caracteres en el programa fuente, que coinciden con el patrón para un token y que el analizador léxico identifica como una instancia de ese token.

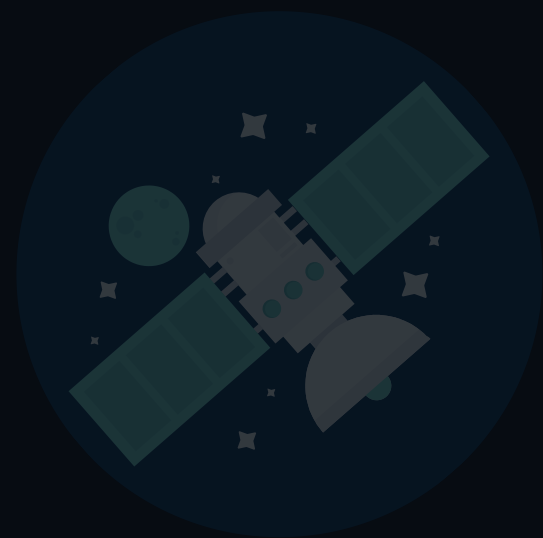
```
let subjectAverage;  
query(  
  "SELECT * FROM marks WHERE subject_ID=" + subject  
  function (datasetsWithSubject) {  
    if (datasetsWithSubject.length > 0) {  
      subjectAverage = 0;  
      datasetsWithSubjectLength = datasetsWithSubject.length;  
      datasetsWithSubject.forEach((dataset) => {  
        subjectAverage += parseFloat(dataset["marks"]);  
      });  
      subjectAverage = subjectAverage / datasetsWithSubjectLength;  
    } else {  
      subjectAverage = 0;  
    }  
  }  
);
```

Analizador Léxico



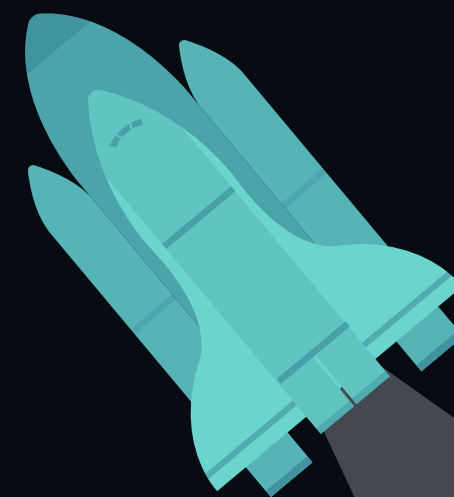
ERRORES

Se detectan cuando el analizador intenta reconocer componentes léxicos y la cadena de caracteres de la entrada no encaja con ningún patrón. Son situaciones en las que se usa un carácter inválido que no pertenecen al vocabulario del lenguaje de programación.



¿DUDAS?

¿DUDAS?



Operaciones entre lenguajes



Cerradura de Kleene



Concatenación



Cerradura estrella



Cerradura positiva

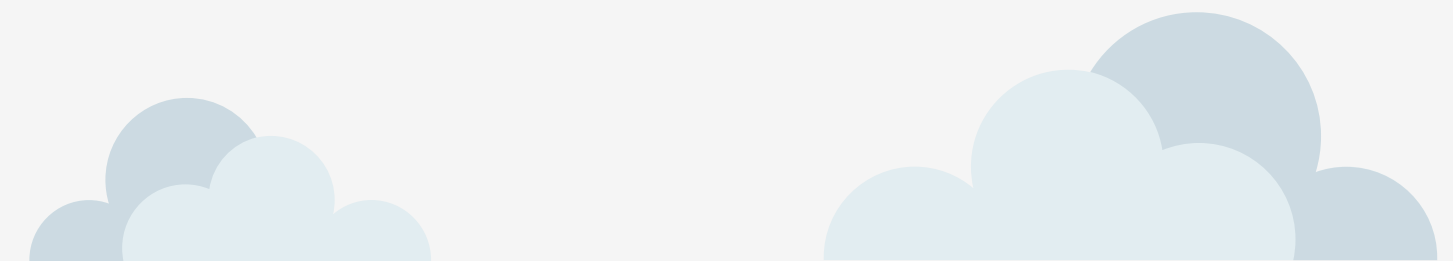
Cerradura de Kleene

Σ^*

Es el lenguaje con todas las cadenas que se pueden formar con el alfabeto.

Sea $\Sigma = \{0, 1\}$

$\Sigma^* = \{0, 1, 00, 01, 10, 11, 000, 100, 101, 111, \dots\}$



Concatenación

L y M = LM

La concatenación de lenguajes se forma combinando todas las cadenas del lenguaje L con las cadenas del lenguaje M.

Sean:

$L = \{a, b, c\}$

$M = \{1, 2\}$

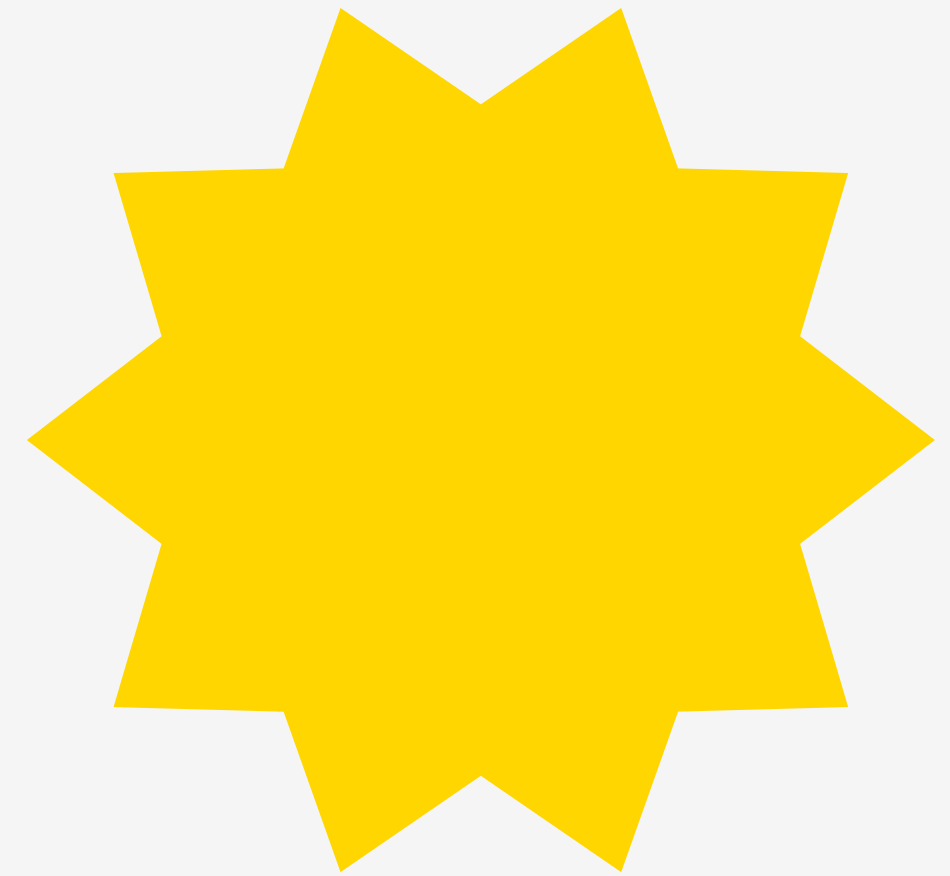
$LM = \{a1, a2, b1, b2, c1, c2\}$



Cerradura Estrella

L^*

Es la unión de todos los lenguajes potencia de L desde $n=0$ hasta infinito.



Cerradura Estrella

Sea $\Sigma = \{a, b\}$, $L = \{abb\}$;

$L^* = L^0 \cup L^1 \cup \dots \cup L^\infty = \{\varepsilon\} \cup \{abb\} \cup \{abbabb\} \cup \dots \cup \{abb\}^\infty$

$L^* = \{\varepsilon, abb, abbabb, abbabbabb, \dots\}$

Cerradura Positiva

L^+

Es la unión de todos los lenguajes potencia de L , desde $n=1$ hasta infinito, que se pueden formar con el alfabeto (Σ)

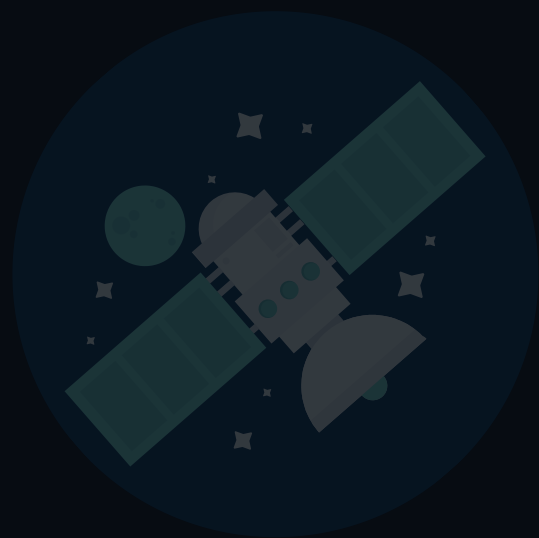


Cerradura Positiva

Sea $\Sigma = \{a, b\}$, $L = \{abb\}$;

$L^+ = L^1 \cup L^2 \cup \dots \cup L^\infty = \{abb\} \cup \{abbabb\} \cup \dots \cup \{abb\}^\infty$

$L^+ = \{ abb, abbabb, abbabbabb, \dots \}$

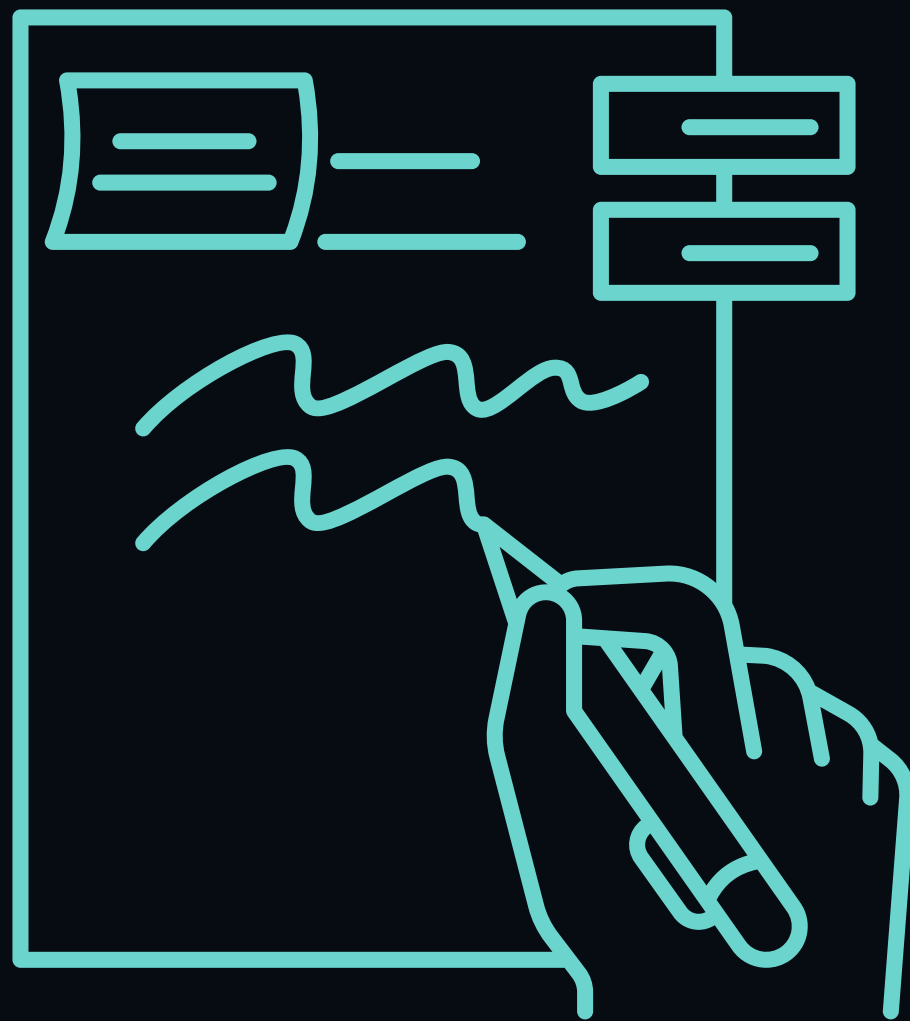


¿DUDAS?

¿DUDAS?



Expresiones Regulares



ER

Es una nueva forma de expresar los lenguajes regulares y tiene como finalidad facilitar la manipulación y simplifiacación de los mismos. La equivalencia entre lenguajes regulares y expresiones regulares es como se indica en la tabla a continuación:

Expresiones Regulares



Lenguaje regular	Expresión regular
$\{a\}$	a
$\{\epsilon\}$	ϵ
\emptyset^*	ϵ
$\{a\}^+$	a^+
$\{a\}^*$	a^*
$\{ab\}$	ab
$\{a\} \cup \{b\} = \{a, b\}$	$a \cup b$

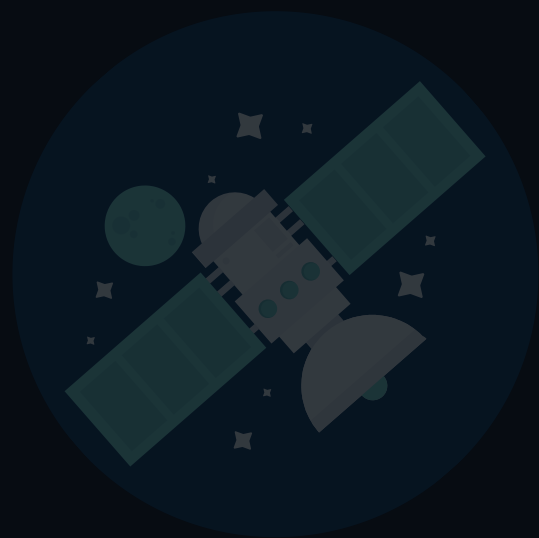
Expresiones Regulares



Operación	Jerarquía
paréntesis	1ª
*, +, n	2ª
Concatenación (·)	3ª
∪	4ª

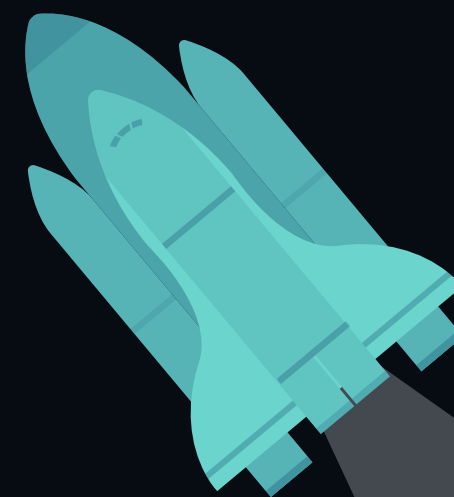
Expresiones Regulares

No.	Equivalencia
1	$a \cup b = b \cup a$
2	$a \cup a = a$
3	$a \cup \emptyset = \emptyset \cup a = a$
4	$\emptyset^* = \epsilon$
5	$(a \cup b) \cup c = a \cup (b \cup c)$
6	$a\epsilon = \epsilon a = a$
7	$a\emptyset = \emptyset a = \emptyset$
8	$b^* = \epsilon \cup b^+$
9	$a^+ = aa^*$
10	$a(b \cup c) = ab \cup ac; (a \cup b)c = ac \cup bc$
11	$(ab)c = a(bc)$
12	$a^* = a^*a^* = a^{**} = (\epsilon \cup a)^* = (a \cup \epsilon)a^* = a^*(a \cup \epsilon) = \epsilon \cup aa^*$
13	$(a \cup b)^* = (a^* \cup b^*)^* = (a^*b^*)^* = (a^*b)^*a^* = a^*(ab^*)^*$
14	$a(ba)^* = (ab)^*a$
15	$(a^*b)^* = \epsilon \cup (a \cup b)^*b$
16	$(ab^*)^* = \epsilon \cup a(a \cup b)^*$
17	$b(a \cup \epsilon)^*(a \cup \epsilon) \cup b = ba^*$
18	$aa^* = a^*a$



¿DUDAS?

¿DUDAS?



Ejemplo de Expresiones Regulares



$(ab \mid c)^*$

$(a \mid b^*)^* d$

$(cb)^* ca(ab)^* \mid b(ba)^* b \mid (ab)^* a(ba)^* b$

Tarea 2



Realizar un ensayo sobre los autómatas:

- Definición de un autómata
- Tipos de autómatas
- Diferencias entre cada autómata
- Ejemplos correspondientes a cada autómata

Fecha de entrega: 08 de septiembre de 2023, antes de las 23:59 horas

No se aceptan entregas tarde.

Python



Importación



Empaquetado

Importación



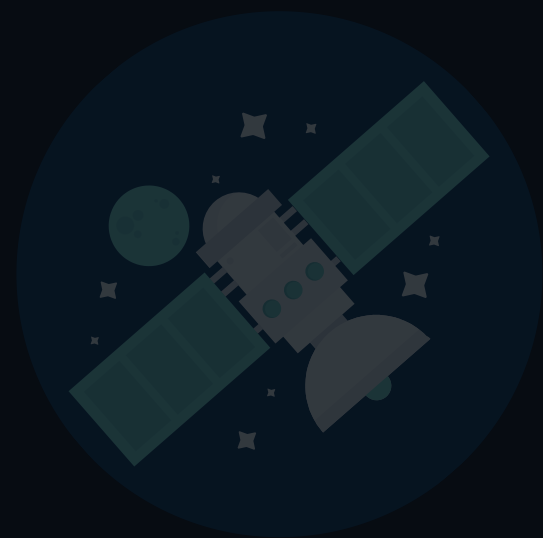
```
1 # mimodulo.py
2 pi = 3.141592
3 def sumar(a, b):
4     return a + b
5 def es_par(n):
6     return n % 2 == 0
```



```
1 # principal.py
2 import mimodulo
3 resultado = mimodulo.sumar(7, 5)
4 print(resultado)
5 print(mimodulo.pi)
```

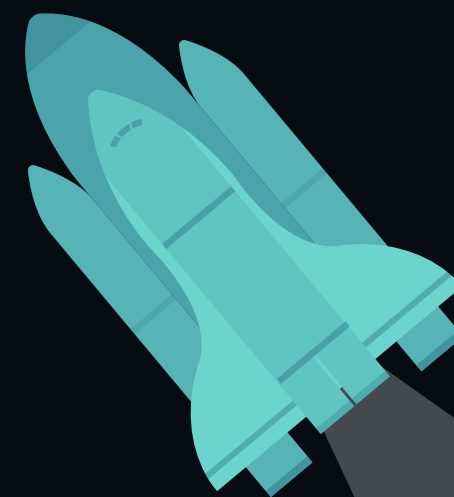
Empaquetado

Ejemplo en Python



¿DUDAS?

¿DUDAS?



Gracias por asistir

¿Dudas o preguntas?



LABORATORIO LENGUAJES FORMALES Y DE
PROGRAMACION