**Objetivo**: Elaborar un traductor básico en Python, que pueda recibir una entrada de cadenas y elaborar un análisis léxico y sintáctico para entregar una salida de acorde a una gramática elaborada.

**Indicaciones**

1. Recibir una entrada, ya sea a través de un txt o dentro del programa tener las cadenas, la última cadena debe reconocer el $ que ha llegado a su fin. Haga uso de bibliotecas como PLY o LARK, RE entre otras más.
2. Mostrar una salida que contenga un cuadro con el detalle del analizador léxico y sintáctico.
   1. Analizador Léxico:
      1. Línea del token encontrado.
      2. Tipo de token.
      3. Valor o elemento de operación.
      4. Otros que considere importante de mostrar.
   2. Analizador Sintáctico:
      1. Genera la salida de la operación.
      2. Construir el árbol (esto es opcional y solo si logra mostrarlo de forma clara). **lark**
3. Elaborar un informe final con los siguientes elementos:
   1. Detalle de los tokens, patrones y lexemas elaborados para el proyecto.
   2. Detalle de las reglas de producción y la gramática elaborada para el proyecto.
   3. Detalle del nombre de las funciones (reglas semánticas) utilizadas para el proyecto.
   4. Capturas del programa en ejecución.
   5. Hoja de desempeño de los integrantes de grupo con él % de trabajo.
   6. Bibliografía base utilizada.
4. Seleccione de los proyectos descritos únicamente uno por grupo.
5. Fecha de presentación será el próximo **lunes 14 de agosto.**
6. Se habilitará un link de subida del informe para ese sábado, no del código, porque será lo que va a explicar y mostrar en día de la presentación.
7. La presentación será según el horario asignado y tendrá 25 minutos para explicarlo, en este día solo será entre el grupo y el docente la demostración.
8. La(s) persona(s) que exponga(n) podrá(n) ser elegida(s) por el grupo, pero todos deben dar su punto de vista sobre el proyecto.

**Proyecto Calculadora de Operaciones**.

Recibir cadenas que contengan operaciones numéricas y que la salida logre identificar el uso de números enteros, decimales, operadores (+,-,\*,/), paréntesis (“()” y/o “[]”), además del resultado de la operación y de forma opción la construcción del árbol.

Ejemplo:

Cadena: 5+3\*(2+1)

Tokens: Numero entero: 5, Operador: +, Paréntesis: Si

Funciones: OperacionSuma: 2+1, OperacionMultiplicacion: 3\*(E)

Resultado: 14

Recuerde que esto es un ejemplo de una línea del txt, el grupo tiene la libertad de trabajar con diferentes nombres y formas de identificar los elementos.

**Proyecto Convertidor de Números**.

Cadenas de números enteros con base decimal (base 10) seguidas de un identificador de conversión de destino los cuales pueden ser:

* Hexadecimal.
* Octal.
* Binario.
* Romano.
* Alternativo (uno inventado o agregado por el grupo).
* Aleatorio (toma uno de los destinos de forma aleatoria).

Ejemplo:

Cadena: 525Romano => Salida: DXXV || Cadena: 525Hexadecimal => Salida: 20D || Cadena: 525Octal => Salida: 1015 || Cadena: 525Alternativo => Salida: “la salida de conversión” || Cadena: 525Aleatorio => Salida: 20D || Cadena: 525Aleatorio => Salida: DXXV

Tokens: serian identificar el destino.

Funciones: Convertir según el destino

Resultado: según la salida del ejemplo.

Recuerde que esto es un ejemplo de una línea del txt, el grupo tiene la libertar de trabajar con diferentes nombres y formas de identificar los elementos.

**Proyecto Lenguaje de Programación.**

Tener la entrada de caracteres que representen un lenguaje definido por ustedes donde se tengan las siguientes operaciones (al menos estas 4, pero ustedes pueden agregar más operaciones):

* Operaciones de Suma y Resta.
* Operaciones de Concatenación.
* Operaciones de Repetición.
* Operaciones de impresión

Ejemplos:

Cadenas-operación-salida

52+ => OperacionSuma=>7

3\*Palabra=>OperacionRepeticion=>PalabraPalabraPalabra

imprimirResultado(53+)=>OperacionImpresion=>8

imprimirResultado(2\*casa)=>OperacionImpresion=>casacasa

A=Soy B=feliz C=con esta clase =>OperacionConcatenacion=>ABC=>Soy feliz con esta clase

Recuerde que esto es un ejemplo de una línea del txt, el grupo tiene la libertar de trabajar con diferentes nombres y formas de identificar los elementos.