

# Laboratorio 4

---

Ing. Max Alejandro Cerna

25 de agosto de 2024

## 1. DESCRIPCIÓN

El objetivo del laboratorio es que el estudiante pueda demostrar su conocimiento acerca de JFlex.

## 2. LABORATORIO

Revisar el siguiente material asincrono para la realización del laboratorio:

<https://www.youtube.com/watch?v=R6znBEekUVA>

Para iniciar la resolución por favor utilice las siguiente tarea de GitHub Classroom:

<https://classroom.github.com/a/c3-Af-GL>

## 3. IDENTIFICADOR DE TOKENS CON JFLEXER

Como primera tarea, el día de hoy se le solicita analizar el código fuente presente en su repositorio de trabajo, expecíficamente la clase TokenConstants.

---

```
1 package gt.com.url;
2
3 public enum TokenConstants {
4     ID,
5     NUM,
6     REAL,
7     KEYWORD,
8     COMMENT,
```

```

9      DOMAIN ,
10     EMAIL ,
11     PHONE ,
12     IP_ADDRESS ,
13     DATE ,
14     EOF
15 }

```

Observe que esta clase es una extensión de la original que se presentó en el material asíncrono. En consecuencia usted deberá crear un analizador léxico capaz de detectar los siguientes lexemas y convertirlos en Token:

1. Números (NUM): 123321, 442424 ✓
2. Números reales (REAL): 3.14159 ✓
3. Palabras clave (KEYWORD): ['if','IF','while','WHILE','ELSE','else'] ✓
4. Comentarios simples de una sola línea: //comentario sin sentido" ✓
5. Nombres de dominio en internet: "http://tumejortrabajo.com" ✓
6. Direcciones de email validas: "foo@gmail.com" ✓
7. Números de teléfono de Guatemala: [-502 7760 1214", "502 78584248", "502 8757 4898", "50212435678, -50278546878"...] ✓
8. Direcciones IPv4: ["172.21.0.1", "192.168.0.1", "127.0.0.1"...] ✓
9. Fechas validas: ["12-08-2024", "2024-08-12", "12/08/2024", "2024/08/12"...] ✓

Cada uno de estas expresiones regulares será validada por los tests descritos en `IdLexerTest.java` el cual usted debe analizar para **1. Encontrar los nombres correctos para las clases, 2. Corregir la estructura del proyecto.**

Al momento de reconocer un Token, su analizador léxico debe imprimir en línea de comandos el Token que ha sido reconocido, con la siguiente estructura (ejemplo para email):

```

1 Token -tokenType=EMAIL, lexeme='estudiante@yahoo.com' -

```

Al momento que su analizador léxico no sea capaz de reconocer un conjunto de caracteres debe recuperarse del error mediante la búsqueda del salto de línea, y a la vez imprimir la cadena **"pacman no pasara"** al momento de la recuperación.

Así mismo debe implementar en este proyecto soporte para invocación vía CLI. Es decir vía `java -jar`, dicho jar podrá ser ejecutado con el siguiente comando:

```

1 java -jar target/laboratorio-5-1.0-SNAPSHOT-jar-with-dependencies.jar -f test.txt

```

Este último test validará el comportamiento de su analizador sobre el archivo de entrada `test.txt`.

#### 4. PREGUNTAS

Utilizando el archivo README.md responda en formato Markdown las interrogantes ahí incluidas.

Fecha límite de entrega: Domingo 1 de septiembre, 23:59