



LENGUAJES FORMALES Y  
DE PROGRAMACION

ABRIL 2022

# Manual Técnico Proyecto2

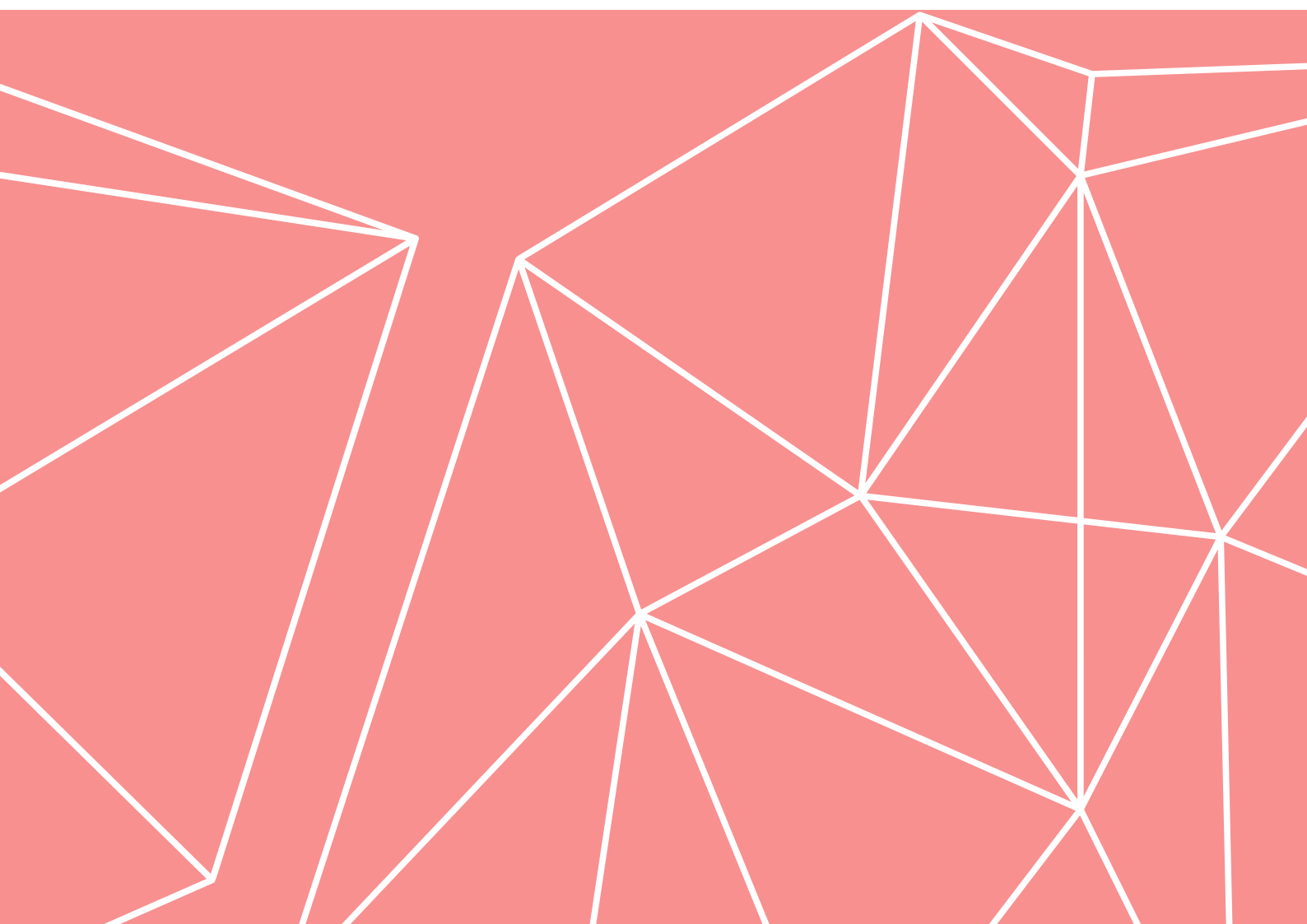
## Proyecto 2

UNIVERSIDAD SAN CARLOS DE  
GUATEMALA

- Facultad de Ingeniería
- Escuela de Ciencias y Sistemas

ANGELA GABRIELA PINELO FLORES

Carnet: 202002536



---

# INDICE

---

202002536 - Angela Gabriela Pinelo Flores

INTRODUCCIÓN, OBJETIVOS, REQUERIMIENTOS .....	01
ALCANES, LOGICA DEL SISTEMA .....	02
DESCRIPCIONES .....	03,04
ER Y AFD .....	05
GLC.....	06
TABLA DE TOKENS.....	07

---

# INTRODUCCIÓN

---

202002536 - Angela Gabriela Pinelo Flores

El siguiente proyecto demuestra en menor medida cómo funciona un analizador Léxico y un analizador Sintáctico, la implementación de autómatas para el analizador léxico y el flujo de un analizador sintáctico generamos reportes, obteniendo la información directamente desde un archivo csv. Todo el programa fue desarrollado con Python y sus librerías.

## OBJETIVOS

- Desarrollar una solución de software implementando un analizador léxico mediante autómatas.
- Desarrollar una solución de software implementando un analizador sintáctico a partir de la construcción de una gramática libre del contexto.
- Aplicar los conocimientos adquiridos en el laboratorio, del lenguaje de programación Python.
- Desarrollar una interfaz gráfica utilizando el lenguaje Python.

## REQUERIMIENTOS

- Conocimientos en los siguientes lenguajes: Python, HTML y CSS
- IDE, de preferencia Visual Studio Code (Puede no ser necesario)
- Sistema Operativo de preferencia
- Tener instalado Python 3 o posterior
- Tener instaladas las siguientes librerías de Python: TkInter, PrettyTable

---

# ALCANCES

---

202002536 - Angela Gabriela Pinelo Flores

La intención del proyecto es que el encargado del desarrollo de esta aplicación aprenda a reconocer caracteres válidos para un lenguaje en específico, por medio de un analizador léxico y posteriormente de su análisis, generar reportes de los tokens válidos, reportes de errores léxicos, y lo principal, posteriormente al análisis léxico comprender la lógica de un analizador sintáctico implementando una gramática libre de contexto que es adjuntada posteriormente.

## LÓGICA DEL PROGRAMA

A continuación se presenta una breve descripción de las clases, métodos, funciones y módulos que fueron implementadas para la realización del programa :

- main (módulo dónde inicia el programa)
- Analisis (módulo donde se encuentra la clase analizador que incluye el analizador léxico y el sintáctico)
- Token (Clase de tipo objeto que representa nuestros Tokens)
- Error (Clase de tipo objeto que representa nuestros Errores)

---

# DESCRIPCIONES

---

202002536 - Angela Gabriela Pinelo Flores

## **main**

Es en este módulo donde se crea una clase GUI utilizando la librería de Tkinter de python, que representa la interfaz que esta conformada por una ventana "login" que representa el inicio del programa para que el usuario ingrese su nombre. Luego esta el chat en sí dentro de la misma clase que es otra ventana con un text area y varios componentes que simulan un chat.

## **Analisis; Analizador**

En esta clase se encuentra la funcionalidad del programa en su mayoría. Se encuentra la función AnalisisLexico que es donde empieza el flujo del programa, esta función fue creada a partir de un autómata finito determinista en base a una expresión regular (autómata adjuntado posteriormente). El análisis léxico está conformado por 6 estados. También existe en esta clase la función de AnalisisSintactico que es donde se procede a generar un análisis más exacto para poder darle funcionalidad al sistema. Dentro de esta clase hay un conjunto más extenso de funciones que apoyan a ambos analizadores, entre ellas estan:

- `reporteTokens()`: Para mostrar el reporte de tokens.
- `reporteErrores()`: Para mostrar el reporte de errores.

# DESCRIPCIONES

202002536 - Angela Gabriela Pinelo Flores

- `reporteTokens()`: Para mostrar el reporte de tokens.
- `reporteErrores()`: Para mostrar el reporte de errores.
- `estadoRecursoivo()` El apoyo principal del analizador sintáctico.

## Token/Error

Estas clases representan objetos de tipo Token y Error que son mayormente utilizados para agregar Tokens y Errores a nuestras listas y poder generar los respectivos reportes.

```
class Token:
    def __init__(self,lexema,tipo,linea,columna):
        self.lexema = lexema
        self.linea = linea
        self.columna = columna
        self.tipo = tipo

    def enviarDataTok(self):
        return [self.lexema, self.tipo, self.linea, self.columna]

    def getTok(self): ...

class Error :
    def __init__(self,tipo,descripcion, linea,columna):
        self.descripcion = descripcion
        self.linea = linea
        self.columna = columna
        self.tipo = tipo

    def enviarDataError(self):
        return [self.descripcion, self.tipo, self.linea, self.columna]

    def getError(self): ...
```

# ER y AFD

202002536 - Angela Gabriela Pinelo Flores

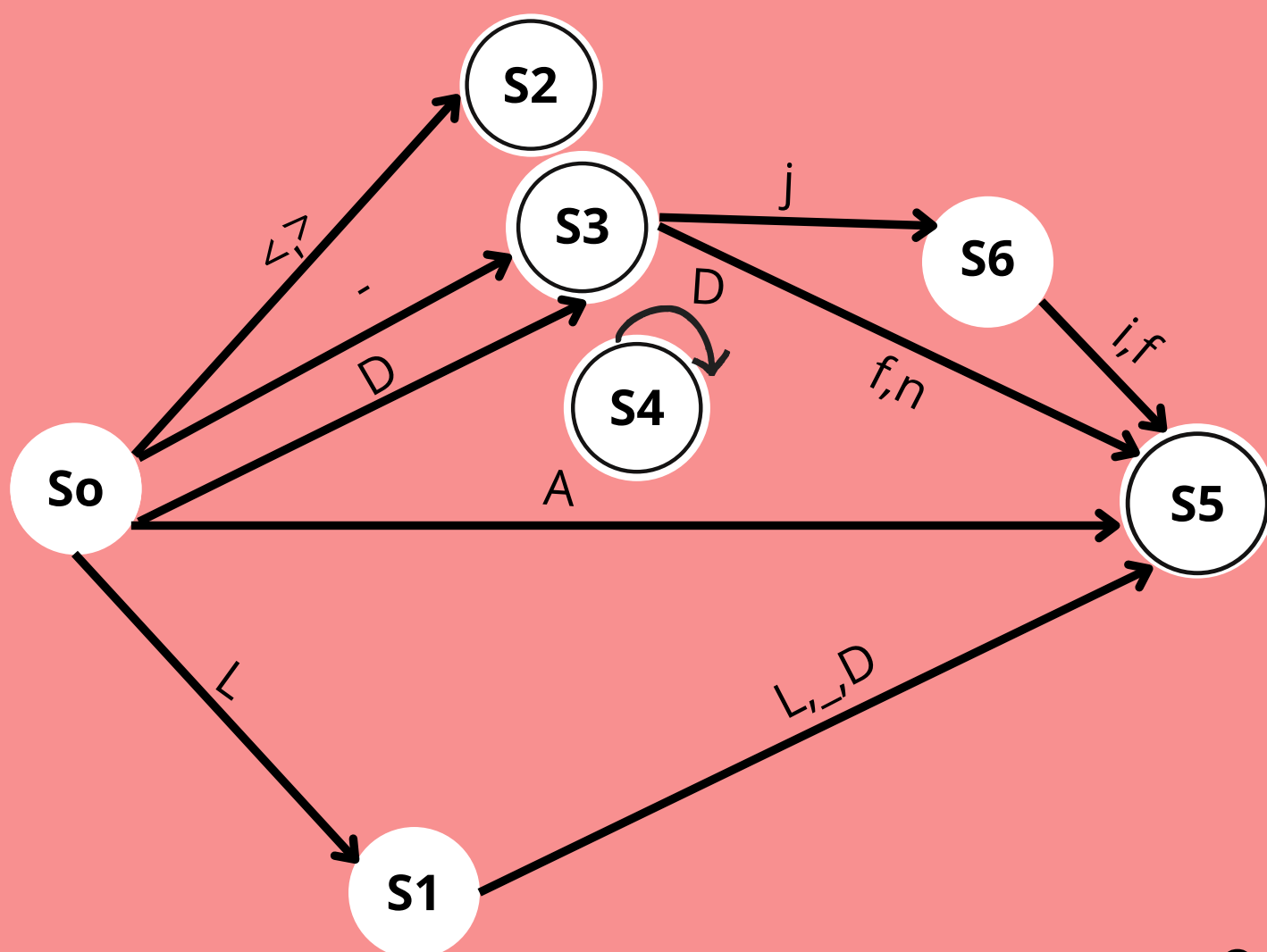
La siguiente expresión regular fue utilizada para desarrollar el Autómata finito determinista:

L = Letras en mayúscula

D= Dígitos

A= cualquier carácter

$(L(L|D|_)|<|-|>|DD|"A*"|-(f|n|j(i|f)))\#$



# GLC

202002536 - Angela Gabriela Pinelo Flores

S	→	INCIO	
INICIO	-->	RESULTADO	equipo=letras bandera= -f letras,-ji entero,jf entero año= DDDD
		JORNADA	
		GOLES	
		TABLA	T={< , > , " , - , D , L , banderas}
		PARTIDOS	N={S, RESULTADO, JORNADA, GOLES,
		TOP	TABLA, PARTIDOS, TOP, ADIOS}
		ADIOS	
RESULTADO	-->	pr_resultado"equipo"pr_vs"equipo"pr_temporada<año-año>	
JORNADA	-->	pr_jornada entero pr_temporada <año-año>   pr_jornada entero pr_temporada <año-año> bandera	
GOLES	-->	pr_goles pr_total "equipo" pr_temporada <año-año>   pr_goles pr_local "equipo" pr_temporada <año-año>  pr_goles pr_visitante "equipo" pr_temporada <año-año>	
TABLA	-->	pr_tabla pr_temporada <año-año>   pr_tabla pr_temporada <año-año> bandera	
PARTIDOS	-->	pr_partidos"equipo"pr_temporada<año-año>   pr_partidos"equipo"pr_temporada<año-año>bandera	
TOP	-->	pr_top pr_superior pr_temporada <año-año>   pr_top pr_inferior pr_temporada <año-año>   pr_top pr_superior pr_temporada <año-año> -n entero   pr_top pr_inferior pr_temporada <año-año> -n entero	
ADIOS	-->	pr_adios	



# TABLA TOKENS

202002536 - Angela Gabriela Pinelo Flores

TOKEN	ER	EJEMPLO
PALABRAS RESERVADAS	$L(L)^*$	TEMPORADA ADIOS RESULTADO
EQUIPOS	$"A^*"$	"Real Madrid" "Barcelona"
Bandera	$-(f j i j f n)$	-f -n -jf
Símbolos	$S$	< > -
Nombre doc	$L(L D _)^*$	Jornada1Reporte 3
Año	$DDDD$	2019 2020 2011

Pequeño ejemplo de autómata para los equipos:

