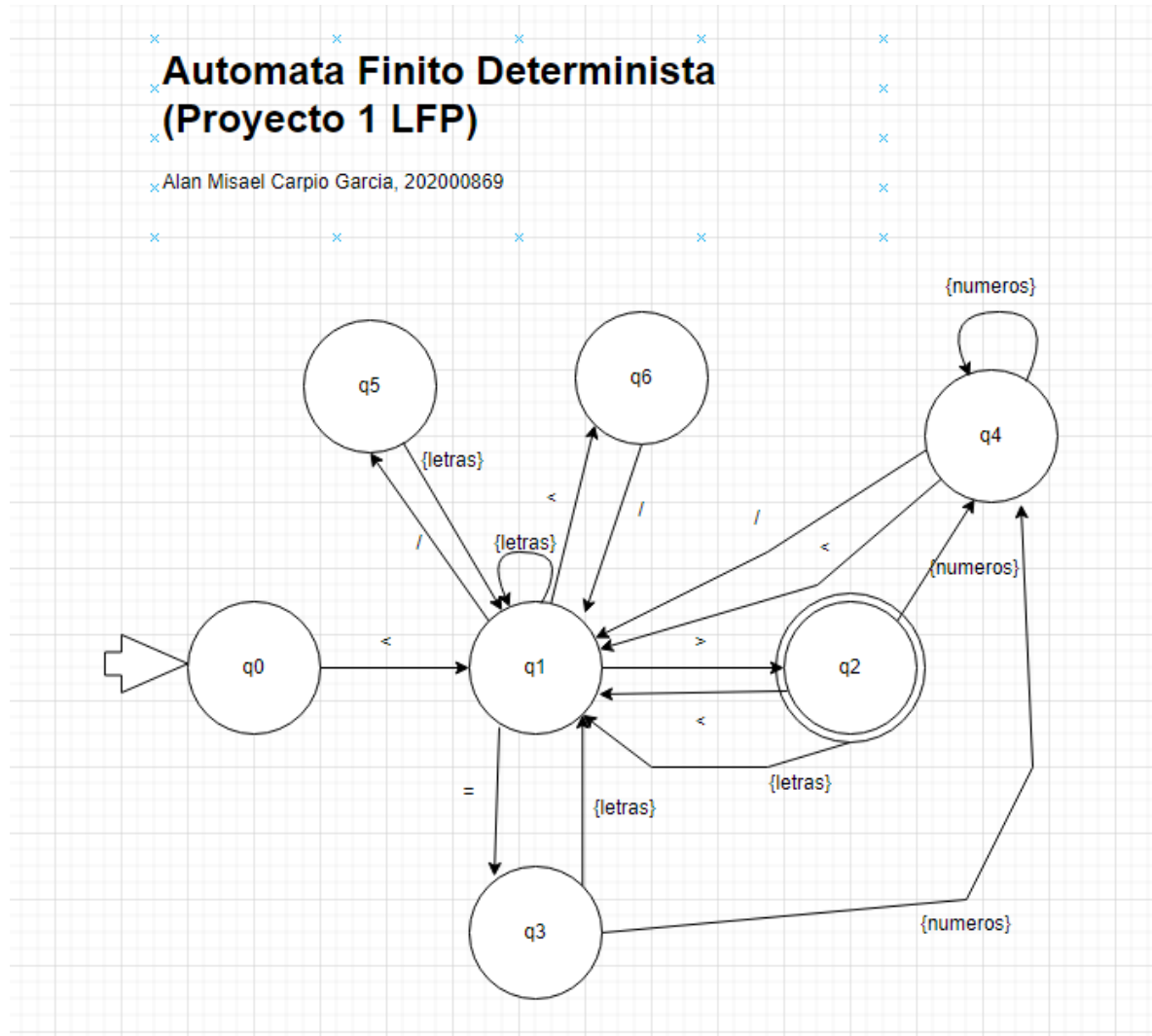


Manual de Tecnico

Para resolver este Proyecto se utilizó un DFA (Autómata Finito Determinista) para poder hacer el análisis léxico del documento

Diagrama del DFA



1.Alfabeto = {/, <,>, =, [letras], [números]}

2.Conjunto de estados = {q0, q1, q2, q3, q4, q5, q6}

3.Estado inicial = {q0}

4.Aceptacion = {q2}

5. Función de Transición

Estados	/	<	>	=	letras	números	FDC
Q0	-----	Q1	-----	-----	-----	-----	No acepto
Q1	Q5	Q6	Q2	Q3	Q1	-----	Acepto
Q2	-----	Q1	-----	-----	Q1	Q4	Acepto
Q3	-----	-----	-----	-----	Q1	Q4	No acepto
Q4	Q1	Q1	-----	-----	-----	Q4	No acepto
Q5	-----	-----	-----	-----	Q1	-----	No acepto
Q6	Q1	-----	-----	-----	-----	-----	No acepto

Para la interfaz grafica se utilizo Tkinter, se utilizaron 5 clases para darle solución al problema la clase autómatas en nuestro DFA pondré se hace el análisis del documento y sacar los tokens respectivos para poder sacar la información necesaria, la clase operaciones se encarga de hacer las operaciones matemáticas cuando en el documento tenemos las etiquetas operación en el documento, la clase HTML se encarga de guardar la información necesaria para la elaboración de la pagina web tanto como sus ajustes de estilos etc... ,la clase tokens guarda los tokens no validos que el autómatas no reconoce para luego posteriormente si tiene errores el documento los mostrara en la pagina web y la clase página tiene el código donde se generara la página web donde estarán todos los datos cuando termine el análisis del documento. Se utilizó programación orientada a objetos para poder guardar la información ya analizada del documento

Clase Autómata

```
import re
from math import *

class automata:
    > class operaciones():...
    > class HTML():...
    > class tokens:...
    > def __init__(self) -> None:
        self.letras = ["a","b","c","d","e","f","g","h","i","j","k","l","m","n","o","p","q","r","s","t","u","v","w","x","y","z",".",",","/"]
        self.numeros = ["1","2","3","4","5","6","7","8","9","0","+", "-", "(", ")", "%", "!", "@", "#", "$", "%", "&"]
        self.operadores = ["suma","resta","multiplicacion","division","potencia","raiz","inverso","seno","coseno","tangente","mod"]
        self.estado_actual = 0
        self.estado_anterior = 0
        self.fila = 1
        self.columna = 0
        self.cadena = ""
        self.token = ""
        self.tabla_tokens = []
        self.tabla_tokensMalos = []
        self.tabla_operaciones = []
        self.tabla_HTML = []
        self.HTML_dic = {}
        self.HTML_etiqueta = ""
        self.HTML_tipo = ""
        #-----
        self.operador = []
        self.operador_cont = 0
        self.operador_guardar = ""
        self.cierre_opera = False
        self.cierre_1 = False
        self.cierre = False
        self.numeros_eti = []
        self.numeros_eti_aux = []
        self.texto = ""
        pass
    > def analizar(self,cadena):...
    > def Guardar_token(self,columna,fila,token):...
    > def Guardar_tokenError(self,columna,fila,token):...
    > def Guardar_operacion(self,tipo,dato):...
    > def Guardar_HTML(self,etiqueta,tipo,dato):...
```

Clase Pagina

```
from multiprocessing.connection import wait
from tkinter import Tk
from Automata import *

class pagina:
    def pagina_html(self, automata_1):
        lista_op = automata_1.tabla_operaciones
        lista_tk = automata_1.tabla_tokensMalos
        lista_ht = automata_1.tabla_HTML
        TextoHT = ""
        TituloHT = ""
        ColorT = ""
        sizeT = 0
        ColorD = ""
        sizeD = 0
        ColorC = ""
        sizeC = 0
        for iter in lista_ht:
            if iter.etiqueta.lower() == "texto":
                TextoHT = iter.datos
            elif iter.etiqueta.lower() == "titulo":
                TituloHT = iter.datos
            elif iter.etiqueta.lower() == "estilo":
                if iter.tipo.lower() == "titulo":
                    ColorT = iter.datos["Color"].lower()
                    sizeT = iter.datos["Tamanio"]
                elif iter.tipo.lower() == "descripcion":
                    ColorD = iter.datos["Color"].lower()
                    sizeD = iter.datos["Tamanio"]
                elif iter.tipo.lower() == "contenido":
                    ColorC = iter.datos["Color"].lower()
                    sizeC = iter.datos["Tamanio"]
        w = ""
        for asd in lista_op:
            asd.operar()
        for iter2 in lista_op:
            w += ""
            <p style="color:{color};font-size:{size}px">{tipo} ---- >{res}</p>
            """.format(tipo = iter2.tipo, res = iter2.datos, color = ColorC, size = sizeC)
        tabla = ""
        if len(lista_tk) != 0:
            for iter2 in lista_tk:
                tabla += ""
                <tr>
                <td>{Fila}</td>
                <td>{Columna}</td>
                <td>{Lexema}</td>
                </tr>
```