

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA
ESCUELA DE CIENCIAS Y SISTEMAS
LENGUAJES FORMALES Y DE PROGRAMACION
Ing. Otto Rodríguez



DAVID ROBERTO DIAZ PRADO

201807420

SECCION A+

Aux: Elmer Real

Para poder tener una solución al problema planteado en el enunciado se dio a conocer que se trabajaría con Automatas y Gramaticas para esto se empleo un paradigma orientado a objetos

```
class Estado():
    def __init__(self,nombre):
        self.Nombre=nombre
        self.Inicial=False
        self.Aceptacion=False

class Transicion():
    def __init__(self,Eo,Ef,s):
        self.EstadoInicial=Eo
        self.EstadoFinal=Ef
        self.Simbolo=s

class Automata():
    def __init__(self,nombre,estado,transicion,alfabeto):
        self.Nombre=nombre
        self.ListaEstados=estado
        self.Alfabeto=alfabeto
        self.ListaTransiciones=transicion
```

```
class NoTerminal:
    def __init__(self,nombre):
        self.Nombre=nombre
        self.Inicial=False
        self.Aceptacion=False
        self.Producciones=[]

class Gramatica:
    def __init__(self,nombre,terminales,nterminales):
        self.Nombre=nombre
        self.ListaTerminales=terminales
        self.ListaNoTerminales=nterminales
```

Estas fueron las clases que se utilizaron para poder realizar dicha tarea

La lógica para crear el algoritmo para la validación de una cadena dentro de un autómata o una cadena fue la siguiente

```
def valAuto(cadena,nombreAuto):
    state=""
    for automata in listaAutomatas:
        if automata.Nombre==nombreAuto:
            for estado in automata.ListaEstados:
                if estado.Inicial:
                    state=estado.Nombre
    for automata in listaAutomatas:
        if automata.Nombre==nombreAuto:
            for letra in cadena:
                for tr in automata.ListaTransiciones:
                    if (tr.EstadoInicial==tr.EstadoFinal) and (tr.EstadoInicial==state):
                        if letra==tr.Simbolo:
                            state=tr.EstadoFinal
                            break
                        if state==tr.EstadoInicial:
                            if tr.Simbolo==letra:
                                state=tr.EstadoFinal
                                break
    if esEA(state,nombreAuto):
        return "valida"
    else:
        return "no valida"
```

esto devuelve si la cadena es valida o no