

AUTÓMATAS Y LENGUAJES FORMALES - H1



IDENTIFICACIÓN DE SPAM EN UN EMAIL USANDO AUTÓMATAS FINITOS

Julián Pérez
Fernando Montañez
Janer Vega

CONTENIDOS

Agenda del día



1. Introducción
2. Propuestas de solución al problema
3. Definición formal del autómata e implementaciones
4. Resultados y conclusiones



RESUMEN

El siguiente proyecto busca por medio de los contenidos vistos en la asignatura resolver alguna problemática o situación de la vida real, como lo es para este caso, la identificación de spam en un correo electrónico usando autómatas finitos para analizar los contenidos de dicho tipo de correo.

INTRODUCCIÓN

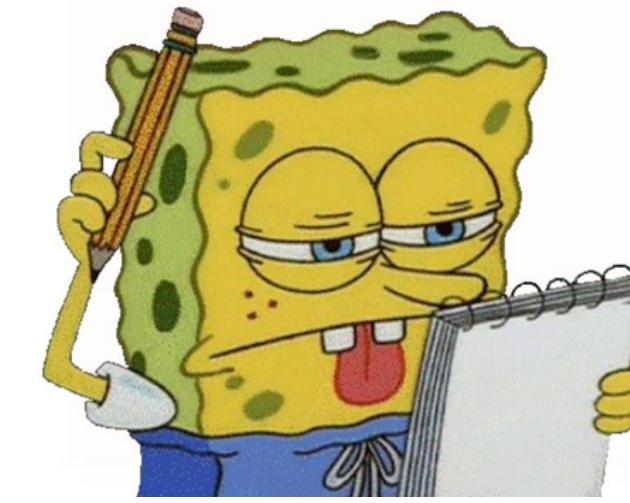
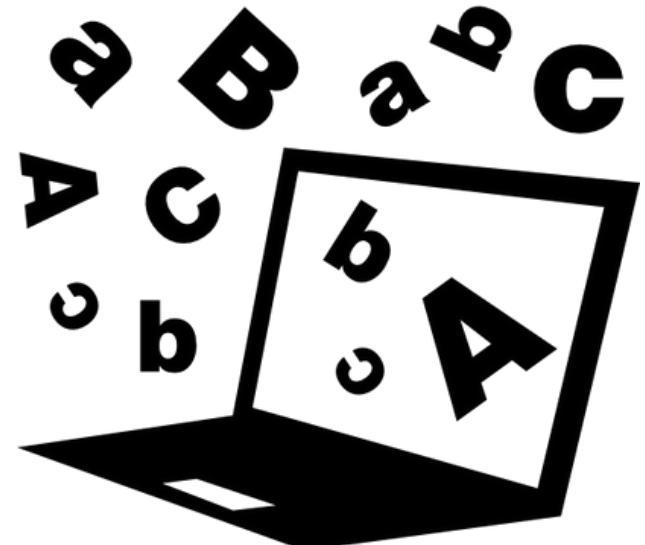
La idea detrás de este proyecto es poder analizar con claridad y eficiencia el contenido de un email clasificado como spam.



Es bien sabido que de muchos correos recibidos dia a dia, algunos cuentan con contenido molesto o malicioso, lo cual puede comprometer de alguna manera nuestra información personal y sensible, caso que queremos evitar en lo posible.

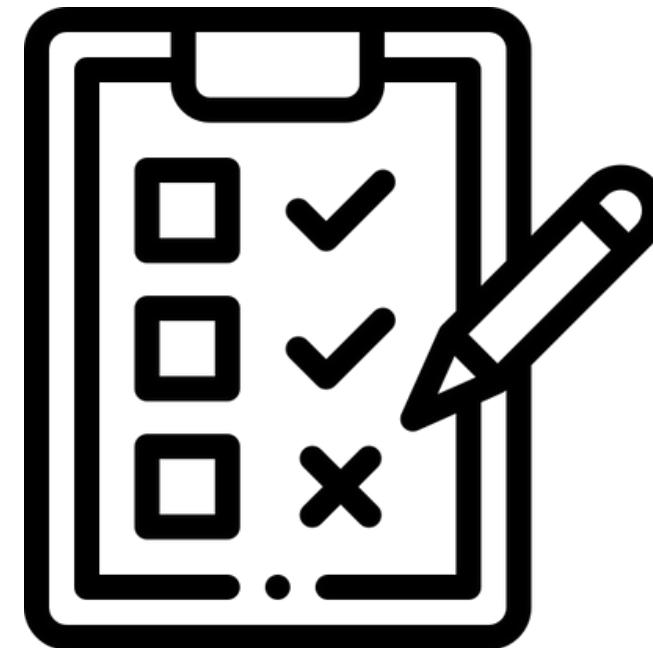
PROPUESTAS PARA SOLUCIONAR EL PROBLEMA

1. Analizar que patrones manejan los emails de tipo spam, para ello se consultaron las palabras clave en este tipo de mensaje



2. En base a esto, se recibe el email en un archivo de texto y se le hace su respectivo tratamiento (espacios en blanco, puntuaciones, acentos, etc) y se agregan a una lista

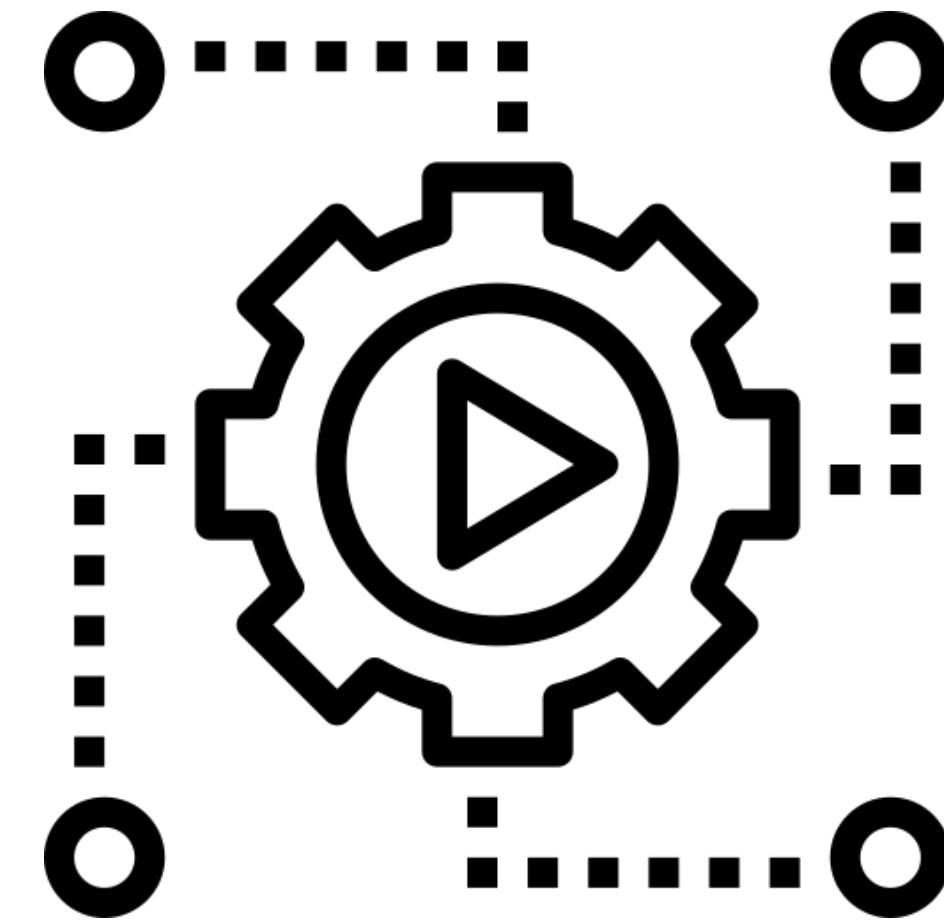
3. Una vez hecho el tratamiento de palabras, vamos a considerar que el correo que tenga 7 o más palabras clave, será marcado como spam. Esto se hace a través de una función que evaluará palabras de la lista y las probará en el autómata

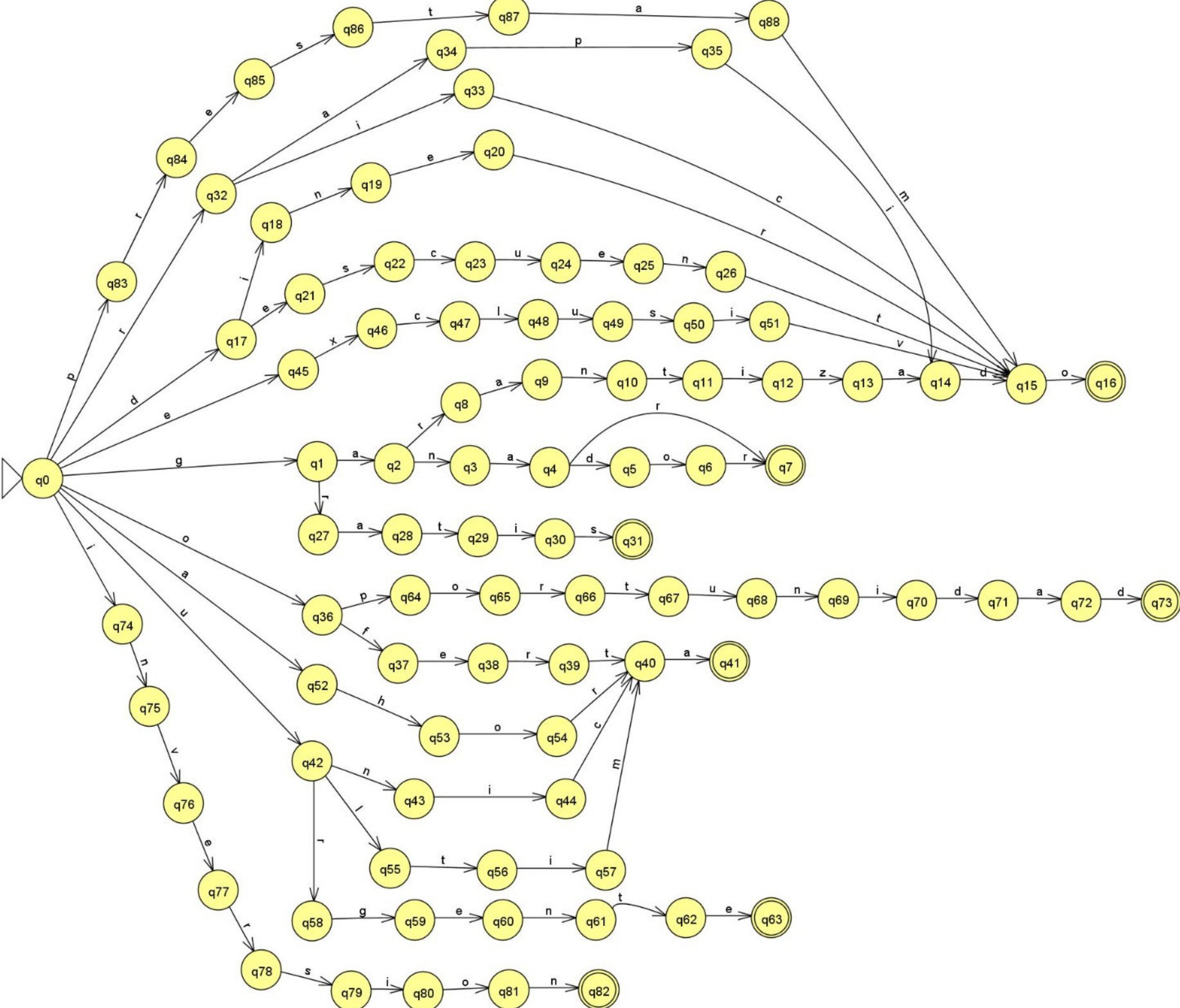


$$a^b + c$$

4. Según los resultados que determine el autómata, este nos dirá si la secuencia de palabras leídas es potencial para spam o no.

DEFINICIÓN FORMAL DEL AUTÓMATA





DEFINICIÓN FORMAL

Recordando que:

$$A = (Q, \Sigma, \delta, q_0, F)$$

Donde:

- Q = Estados
- Σ = Alfabeto
- δ = Funciones de transición
- q_0 = Estado inicial
- F = Estados de aceptación

$$A = (Q, \Sigma, \delta, q_0, F)$$

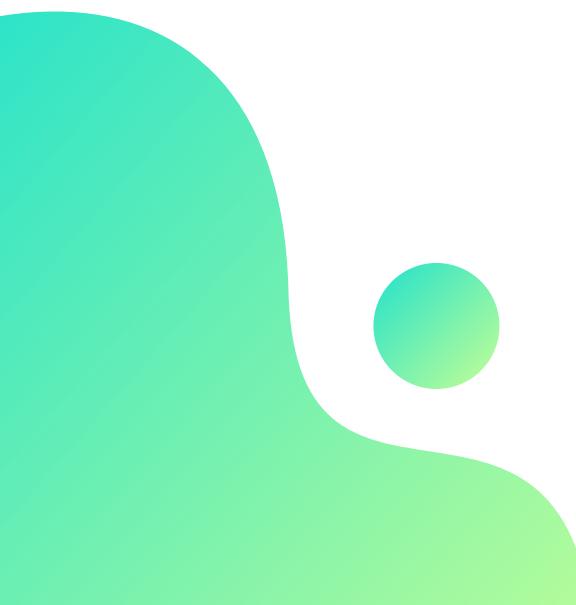
Se tiene que:

- $Q = \{q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8, q_9, q_{10}, \dots, q_{87}, q_{88}\}$ | 88 estados,
- $\Sigma = \{'a', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'l', 'm', 'n', 'o', 'p', 'r', 's', 't', 'u', 'v', 'x', 'z'\}$ | 20 letras,
- $\delta = \{\delta(q_0, p) = q_{83}, \delta(q_0, r) = q_{32}, \delta(q_1, a) = q_2, \delta(q_2, n) = q_3, \dots\}$,
- $q_0 = \{q_0\}$,
- $F = \{q_7, q_{16}, q_{31}, q_{41}, q_{63}, q_{73}, q_{82}\}$



Algunas de las tantas palabras clave en correos de spam son:

gana, dinero, rapido, rico, oferta, unica,
descuento, exclusivo, ahora, ultima, oportunidad,
urgente, garantizado, ganador, prestamo,
inversion, gratis



IMPLEMENTACIONES

Funciones de tratamiento de texto

```
● ● ●  
1 # Convertir de cadena de texto a lista  
2 def convertirALista(texto):  
3     lista = list(texto.split(' '))  
4     return lista  
5  
6 # Leer un archivo de texto  
7 def leerArchivo(archivo):  
8     texto = open('ejemplos/{}.txt'.format(archivo), 'r', encoding='utf-8')  
9     contenido = texto.read()  
10    texto.close()  
11    return contenido # Devolvemos el contenido en forma de cadena
```

```
● ● ●  
1 # Eliminar los simblos de puntuación y acentos del texto  
2 def eliminarPuntuacion(texto):  
3     import string, re, unidecode # Se necesita instalar → 'pip install Unidecode'  
4     texto = texto.translate(str.maketrans('', '', string.punctuation)) # Eliminamos puntuación  
5     texto = unidecode.unidecode(texto) # Eliminamos acentos  
6     # texto = unidecode.unidecode(texto) # Eliminamos ? y ! que reemplazaron a ¿ y i en la 1ra ejecución  
7     # Notar que ¿ y i son casos especiales del español y no se eliminan arriba  
8     # Nota: En donde se elimina la puntuación, lo que se hace es que se cambian: ¿→? y i→!,  
9     # entonces eliminar ¿ y i no tiene sentido. Hay dos posibles soluciones, eliminar ? y !  
10    # o correr una segunda vez la función de eliminar puntuación.  
11    # Nota de la nota: La 2da solución no funciona  
12    texto = re.sub('[?!]', '', texto) # Eliminamos ¿ y i  
13    return texto  
14  
15 def pasarAMinusculas(texto):  
16     texto = texto.lower()  
17     return texto
```



```
1 # Eliminar saltos de linea y los reemplaza por un ' '
2 # Nota: esto nos deja dobles espacios y por ende, cadenas vacias en la lista luego de convertir
3 def eliminarSaltosDeLinea(texto):
4     texto = ' '.join(texto.split('\n'))
5     return texto
6
7 # Eliminar las cadenas vacias que quedan luego de
8 def eliminarCadenasVacias(lista):
9     while '' in lista:
10         lista.remove('')
11     return lista
12
13 # Evaluar que palabras son aceptadas por el autómata
14 def evaluar(palabra):
15     from dfaProyecto import automata
16     d = automata()
17     if d.accepts_input(palabra) != True:
18         return False
19     return True
```

Función main

```
● ● ●  
1 # Traemos todas las funciones  
2 from funciones import *  
3  
4 # Tratamos el archivo de texto  
5 a = leerArchivo('ejemplo1')  
6 a = eliminarPuntuacion(a)  
7 a = pasarAMinusculas(a)  
8 a = eliminarSaltosDeLinea(a)  
9 b = convertirALista(a)  
10 b = eliminarCadenasVacias(b)  
11 # print(b)  
12  
13 puntosSPAM = 0; # Si es mayor o igual a 7 se considerará SPAM  
14  
15 # Vemos que palabras son aceptadas por el autómata  
16 for i in range(0, len(b)):  
17     if len(b[i]) ≥ 4: # Solo evaluamos palabras con longitud 4 o mayor  
18         if evaluar(b[i]):  
19             puntosSPAM += 1  
20  
21 # Definimos si un correo es potencialmente SPAM  
22 if puntosSPAM ≥ 7:  
23     print('El correo evaluado es potencialmente SPAM.')  
24 else:  
25     print('El correo evaluado no es SPAM')
```

Implementación del autómata en Google Collab

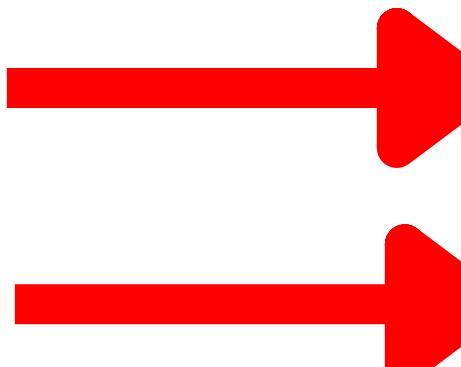


RESULTADOS Y CONCLUSIONES

1. Se lograron cumplir los objetivos propuestos desde la presentación de la idea del proyecto (identificación del mensaje, filtración del correo).
2. La estrategia pensada para resolver el problema fue la adecuada.
3. Los conceptos de autómatas finitos fueron útiles durante el desarrollo del problema
4. A futuro, esta implementación del autómata puede mejorar añadiéndole más palabras clave



Resultados de ejecución



```
PS D:\OneDrive - UNIVERSIDAD INDUSTRIAL DE SANTANDER\Autómatas\Proyecto final  
\IdentificadorSPAM> python .\main.py  
El correo evaluado es potencialmente SPAM.  
PS D:\OneDrive - UNIVERSIDAD INDUSTRIAL DE SANTANDER\Autómatas\Proyecto final  
\IdentificadorSPAM> python .\main.py  
El correo evaluado no es SPAM  
PS D:\OneDrive - UNIVERSIDAD INDUSTRIAL DE SANTANDER\Autómatas\Proyecto final  
\IdentificadorSPAM>
```

GRACIAS

