

# Tecnológico Nacional de México Campus Culiacán



**TECNOLÓGICO  
NACIONAL DE MÉXICO**



## ***“Detector de spam - Documentación”***

***Nombre de los alumnos:***

*Rios Saucedo Jose Lorenzo.*

*Cazarez Ibarra Francisco Javier*

***Docente:*** *Zuriel Dathan Mora Félix.*

***Materia:*** *Inteligencia Artificial.*

***Carrera:*** *Ing. En sistemas computacionales.*

***Semestre:*** *8*

***Horario:*** *9-10AM*

***Fecha:*** *30 de marzo del 2025.*

Primero se comienza importando las librerías necesarias para comenzar a trabajar en nuestro proyecto, **Pandas** para el manejo de datos en Datasets/Frame, **re** para expresiones regulares y patrones de texto, **urllib.parse** para analizar URLs, **datetime** para fechas y horas y finalmente **Collections** para la detección de palabras repetidas en un texto o correo en este caso.

```
import pandas as pd
import re
from urllib.parse import urlparse
from datetime import datetime
from collections import Counter
```

Después se carga el dataset de donde obtendremos los datos, en caso de que no este el archivo en nuestro espacio de trabajo saltara una excepción

```
#Se carga el dataset
try:
    df = pd.read_csv("tarea8/spam_assassin.csv")
except FileNotFoundError:
    print("Error: El archivo spam_assassin.csv no se encontró.")
    exit()
```

Estas funciones extraen partes específicas del texto de un correo

```
#Creamos las funciones para la identificación de características de correo
def extraer_remitente(text):
    remitente_match = re.search(r"From: (.*)?(?:\n|$)", text, re.IGNORECASE)
    if remitente_match:
        return remitente_match.group(1).strip()
    return None

def extraer_asunto(text):
    asunto_match = re.search(r"Subject: (.*)?(?:\n|$)", text, re.IGNORECASE)
    if asunto_match:
        return asunto_match.group(1).strip()
    return None

def extraer_cuerpo(text):
    cuerpo_inicio = re.search(r"\n\s*\n", text)
    if cuerpo_inicio:
        return text[cuerpo_inicio.start():].strip()
    return text.strip()
```

Aquí separamos y ordenamos el contenido del correo con las funciones anteriores, obteniendo las columnas remitentes, asunto y cuerpo del correo

```
#Obtiene el remitente, asunto y cuerpo de cada elemento en el dataset
df['sender'] = df['text'].apply(extraer_remitente)
df['subject'] = df['text'].apply(extraer_asunto)
df['body'] = df['text'].apply(extraer_cuerpo)
```

Este es el inicio de nuestro método de reglas de detección, analiza todo el correo primeramente, luego se va analizando fila por fila de las anteriormente obtenidas en el paso anterior

```
def reglas_de_spam(row):
    texto = row.get('text', '')
    remitente = row.get('sender', '')
    asunto = row.get('subject', '')
    cuerpo = row.get('body', '')
```

Y estas son las reglas, estas funcionan de manera simple como por ejemplo, en la de palabras clave de asunto si se detecta una de las palabras que esta dentro de las reglas lo clasifica como spam y retorna 1, y así es con los demás pero bajo diferentes condiciones

```
# Regla 0: Descartar "Re:" en el asunto
if asunto and asunto.lower().startswith("re:"):
    return 0 # Marcar como no spam

# Regla 1: Asunto con palabras clave sospechosas
palabras_spam_asunto = {"half price", "free", "winner", "urgent", "limited offer", "claim now", "lottery", "earn money", "click now", "prize"}
if asunto and any(word in asunto.lower() for word in palabras_spam_asunto):
    return 1

# Regla 2: Cuerpo con palabras clave de spam
keywords_spam_cuerpo = {"to be removed from", "claim your prize", "limited time offer", "unsubscribe now", "click here", "congratulations"}
if any(keyword in cuerpo.lower() for keyword in keywords_spam_cuerpo):
    return 1

# Regla 4: Remitente con un dominio sospechoso o genérico
if re.search(r"@([mail|web|promo|free])[0-9]+\.\com$", remitente, re.IGNORECASE):
    return 1

# Regla 5: Exceso de signos de exclamación (indicador de urgencia o spam)
if cuerpo.count("!") > 10:
    return 1

# Regla 6: Uso de frases en mayúsculas típicas de spam
if re.search(r"(WIN|FREE|CLICK NOW|URGENT|LIMITED OFFER|BUY NOW|MONEY BACK GUARANTEE)", cuerpo):
    return 1
```

Y para finalizar, primero se aplica la función **reglas\_de\_spam** en cada fila del DataFrame y se almacena lo obtenido en una columna llamada **“Spam”** luego importamos funciones de **scikit-learn** para medir el rendimiento de la detección de spam, después se calcula la precisión del detecto comparando la columna **“spam”** con la columna **“esperado”** y finalmente se muestran los resultados.

```
#Se aplican las reglas al dataset
df['Spam'] = df.apply(reglas_de_spam, axis=1)

#Se evalua el rendimiento/precision
from sklearn.metrics import accuracy_score, classification_report

Precision = accuracy_score(df['Esperado'], df['Spam'])
print(f"\nPrecisión del detector de spam basado en reglas: {Precision:.2f}")
print("\nReporte de Clasificación:\n", classification_report(df['Esperado'], df['Spam']))

#Mostramos algunos correos evaluados y su clasificaciones
print("\nPrimeros 50 correos electrónicos con sus predicciones:")
print(df[['subject', 'Esperado', 'Spam']].head(50))
```

Y al final se añadió un fragmento de código donde se nos pide que creemos uno de manera manual para que el detector de spam lo clasifique según su contenido, simplemente se piden los datos de las columnas, remitente, asunto y cuerpo del correo escritos manualmente, se hace el análisis con las mismas reglas que se utilizaron con el dataset y nos da un resultado dependiendo del contenido

```
# Solicitar datos de un correo electrónico manualmente
print("\nIngrese los datos de un correo para analizar si es spam:")
remitente_manual = input("Remitente: ")
asunto_manual = input("Asunto: ")
cuerpo_manual = input("Cuerpo: ")

# Crear un diccionario con los datos ingresados
✓ correo_manual = {
    'sender': remitente_manual,
    'subject': asunto_manual,
    'body': cuerpo_manual,
    'text': f"From: {remitente_manual}\nSubject: {asunto_manual}\n\n{cuerpo_manual}"
}

# Evaluar el correo ingresado
resultado_spam = reglas_de_spam(correo_manual)

✓ if resultado_spam == 1:
    print("\nEl correo ingresado es SPAM.")
✓ else:
    print("\nEl correo ingresado NO es spam.")
```