

# Analizador de textos con Python

Equipo 5:

Méndez González, Romina Estephania

Moreno Aguilar, Diego Denilson

Rodríguez Castillo, Jordan Ricardo

# Sobre los analizadores de texto...



## ¿Qué es?

Programa que interactúa con un texto para proporcionar información de utilidad sobre éste mismo.



## Función

Recolecta datos provenientes de la información que introduzca el usuario.



## Aplicaciones

Brinda herramientas para la toma de decisiones que pueden ser de apoyo en las empresas o en el ámbito académico.

# Con este proyecto se pretende:

01

Entender la estructura sintáctica implementada en la elaboración de un analizador de textos con el lenguaje de Python.

02

Reconocer la importancia que tiene la creación de programas enfocados en el análisis de datos, tal como un analizador de textos.



## CUERPO

- La estructura `input()` nos permite guardar información para utilizarla después.
- El `.appen()` incluye elementos.
- El `.lower()` transforma un texto a minúsculas.

```
#-----1-----  
texto= input('Ingresa un texto:')  
texto = texto.lower() #lo sobreescribimos  
  
letras = [] #variable que usaremos abajo  
  
letras.append(input('Ingresa la primera letra de tu elección:').lower())  
letras.append(input('Ingresa la segunda letra de tu elección:').lower())  
letras.append(input('Ingresa la tercera letra de tu elección:').lower())
```

- El `.count()` cuenta las veces que aparece un elemento en una lista.
- `'\n'` permite dar un salto de línea.

```
print('\n')
print("Cantidad de letras")
Cantidad_letras1 = texto.count(letras[0])
Cantidad_letras2 = texto.count(letras[1])
Cantidad_letras3 = texto.count(letras[2])

print(f"Hemos encontrado la letra '{letras[0]}' repetida {Cantidad_letras1} veces.")
print(f"Hemos encontrado la letra '{letras[1]}' repetida {Cantidad_letras2} veces.")
print(f"Hemos encontrado la letra '{letras[2]}' repetida {Cantidad_letras3} veces.")
```



- El `.split()` separa los elementos de un string y los incluye en una lista.
- La función `len()` determina cuantos caracteres posee una lista.

```
print('\n')
print("CANTIDAD DE PALABRAS")
palabras = texto.split()
print(f"En el texto hay {len(palabras)} palabras.")
```

```
print('\n')
print("LETRAS DE INICIO Y FIN")
Letra_inicio = texto[0]
Letra_final = texto[-1]
print(f"La primera letra del texto es '{Letra_inicio}'.")
print(f"La última letra del texto es '{Letra_final}'.")
```

## RESULTADOS

↓

⌵

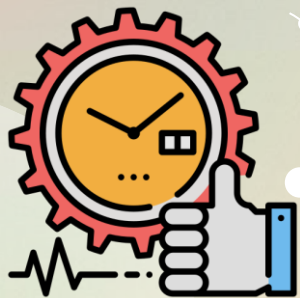
⌵

⌵

⌵

Ingresar un texto:

## CONCLUSIONES



### Eficiencia



El programa nos permite examinar datos de interés de manera eficiente, ya que reduce el tiempo considerablemente a comparación de que si lo hiciéramos manualmente.

### Precisión



Un lenguaje de programación no falla a menos que el código presente errores, por lo que la precisión de los datos que arroja el programa dota de gran confiabilidad a esta herramienta.





The background features a horizontal rainbow gradient from red on the left to purple on the right. Overlaid on this are several thin black lines forming circles and arcs. Two large circles intersect on the left and right sides of the frame. A vertical line runs down the left side, and a horizontal line runs across the bottom. Two solid black dots are placed at the intersections of the vertical line with the left circle and the horizontal line with the right circle.

# Preguntas