

Programa de Archivo

Fernando Ochoa Torres

Matricula:

201649279

Professor:

Adolfo Aguilar Rico

Clase:

<u>Programación Concurrente y Paralela</u>
--

```
class
Archivo:
def
```



```
init(self,nombre): self.f = open("C:\Users\Fernando\Desktop\Prueba2.txt", "rt")
self.nombre="C:\Users\Fernando\Desktop\Prueba2.txt"
```

Se abre el archivo con la ruta deseada, e igualmente se le asigna la variable nombre

Funcion para mostrar archivo

```
def muestra(self): i = 1 for linea in self.f: s = "{:3} {}".format(i,linea) print(s) i += 1 self.f.seek(0)
```

La primera función para mostrar la información del archivo. Usamos el for para recorrer el archivo y asignamos a "s" las líneas para que las imprima con el print.

Funcion de conteo de vocales

```
def cuentaVocales(self): cadena1 = self.f.read() contadorvocales = 0 for i in cadena1: if(i=='A' or
i=='a' or i=='á' or i=='E' or i=='e' or i=='é' or i=='I' or i=='i' or i=='í' or i=='O' or i=='o' or i=='ó' or
```

```
i=='U' or i=='u' or i=='ú'): contadorvocales+=1 print("La cantidad de vocales es :",contadorvocales)
```

Aqui asignamos a cadena1 la el archivo cual leemos con la funcion read().Integramos una variable para que lleve el conteo e incorporamos un for para recorrer la cadena1 que establecimos anteriormente. El if es puesto para identificar las vocales, si es que encuentra una de ellas, el contadorvocales es incrementado. El final solo es imprimir el contador que llevamos.

Funcion de conteo de puntuacion

```
def cuentaPuntuacion(self):    self.f =
open("C:\\Users\\Fernando\\Desktop\\Prueba2.txt", "rt")    cadena5 = self.f.read()
contadorpuntuacion = 0    for i in cadena5:        if i in set(',;.!?:¡'):
contadorpuntuacion += 1    print("La cantidad de puntuaciones es :",
contadorpuntuacion)
```

Abrimos el archivo y asignamos la lectura a cadena5.Inicializamos el contadorpuntuacion para contar la cantidad de puntuaciones en el archivo. Con el for recorreremos la cadena5 ,el texto del archivo, y el if es incluido para identificar si es que existe alguna de las puntuaciones que fueron puestas en el set. Set es otra manera de expresar lo que hicimos anteriormente con los or. Si una es encontrada el contador es incrementado.

Funcion de conteo de palabras

```
def cuentaPalabras(self): self.f=open("C:\\Users\\Fernando\\Desktop\\Prueba2.txt", "rt")
cadena2=self.f.read() np=cadena2.split() npalabras=len(np) print("Cantidad de
palabras:",npalabras)
```

En este caso asignamos la lectura a la cadena2 y usamos la funcion split() para asignar una nueva interpretacion del texto con divisores por cada espacio que es asignado a np. Inicializamos las variable npalabras para el conteo con la funcion len(np) e imprimimos el resultado.

Funcion de conteo de consonantes

```
def cuentaConsonantes(self): self.f=open("C:\\Users\\Fernando\\Desktop\\Prueba2.txt", "rt")
cadena3=self.f.read() nc=0 for i in cadena3: if i in
set("bcd fghjklmnpqrstvwxyzBCDFGHJKLMNPQRSTVWXYZ"): nc+=1 print("La cantidad de
consonantes es:",nc)
```

Esta funcion es algo similar a la de puntuacion,excepto que ahora nuestro set esta hecho por las consonantes del alfabeto en mayuscula y minuscula para evitar tener que convertir el texto a todo mayuscula o todo minuscula.

Funcion de conteo de lineas

```
def cuentaLineas(self): self.f=open("C:\Users\Fernando\Desktop\Prueba2.txt", "rt")
cadena4=self.f.readlines() nl=0 for linea in cadena4: nl+=1 print("La cantidad de lineas es:",nl)
```

Aqui asignamos a la cadena4 la informacion del texto del archivo. Con la funcion readlines() , va leyendo linea por linea. Por eso se inicializa un for para ir contando por cuantas lineas pasa para terminar de recorrer el texto.

Funcion de conteo de espacios

```
def cuentaEspacios(self): cuenta_espacio= 0 ruta_de_archivo=
"C:\Users\Fernando\Desktop\Prueba2.txt" with open(ruta_de_archivo, "r") as self.f: for linea in
self.f: words = linea.split() for word in words: for char in word: if char.isspace(): cuenta_espacio
+=1 print("Cantidad de espacios blancos es:",cuenta_espacio)
```

Aqui tome un acercamiento diferente dado que era necesario contar la cantidad de espacios. Nuevamente usamos el split() para la identificacion de palabras luego integramos un dos for para que vaya revisando por cada caracter en si, y con la funcion isspace() revisa si es un espacio e incrementamos el contador.

Funcion de conteo de mayusculas

```
def cuentaMayusculas(self): cuenta_mayuscula = 0 ruta_de_archivo =
"C:\Users\Fernando\Desktop\Prueba2.txt" with open(ruta_de_archivo, "r") as self.f: for linea in
self.f: words = linea.split() for word in words: for char in word: if char.isupper():
cuenta_mayuscula += 1 print("Cantidad de Mayusculas es:",cuenta_mayuscula)
```

Nuevamente tomas la forma anterior , pero esta vez se usa la funcion isupper() para verificar si es mayuscula e incrementar el contador.

Funcion de conteo de minusculas

```
def cuentaMinusculas(self): cuenta_minuscula = 0 ruta_de_archivo =
"C:\Users\Fernando\Desktop\Prueba2.txt" with open(ruta_de_archivo, "r") as self.f: for linea in
self.f: words = linea.split() for word in words: for char in word: if char.islower():
cuenta_minuscula += 1
    print("Cantidad de Minusculas es:", cuenta_minuscula)
```

Muy similar al conteo de mayusculas pero esta vez se incorpora la funcion islower() para verificar si es minuscula e incrementar el contador.

Funcion de convertir a mayuscula

```
def hacer_Mayusculas(self): ruta_de_archivo = "C:\Users\Fernando\Desktop\Prueba2.txt" with
open(ruta_de_archivo, "r") as self.f: i = 1 for linea in self.f: s = "{:3} {}".format(i, linea)
print(s.upper()) i += 1 self.f.seek(0)
```

Aqui retomas la forma de las primeras funciones, pero en vez de imprimir a s se le incorpora la funcion upper() para convertir la impresion a mayusculas.

Funcion de convertir a minuscula

```
def hacer_Minusculas(self): ruta_de_archivo = "C:\Users\Fernando\Desktop\Prueba2.txt" with
open(ruta_de_archivo, "r") as self.f: i = 1 for linea in self.f: s = "{:3} {}".format(i, linea)
print(s.lower()) i += 1 self.f.seek(0)
```

Similar a convertir a mayuscula pero se usa la funcion lower() para en vez convertir a minuscula.

Funcion de convertir a hexadecimal

```
def convertir_Hexadecimal(self): ruta_de_archivo = "C:\Users\Fernando\Desktop\Prueba2.txt"
with open(ruta_de_archivo, "rb") as self.f: s= self.f.read() print(binascii.hexlify(s))
```

Esta fue algo mas complejo , tuve que investigar e encontre una libreria ,binascii , que me permitio usar el metodo hexlify() para que lo convirtiera.

Funcion para copiar archivo

```
def copiarArchivo(self): fuente= "C:\Users\Fernando\Desktop\Prueba2.txt" destino =
"C:\Users\Fernando\Desktop\Prueba\Prueba2.txt" shutil.copyfile(fuente,destino)
```

Con la libreria shutil, me permitio usar el metodo copyfile() para lograr copiar el archivo de la fuente al destino donde yo lo deseaba.

Llamados

```
x=Archivo("Prueba2.txt")
x.muestra()
x.cuentaVocales()
x.cuentaConsonantes()
x.cuentaPuntuacion()
x.cuentaEspacios()
x.cuentaPalabras()
x.cuentaLineas()
x.cuentaMayusculas()
x.cuentaMinusculas()
x.copiarArchivo()
```

```
x.hacer_Mayusculas()  
x.hacer_Minusculas()  
x.convertir_Hexadecimal()  
print(x.nombre)  
print("Fernando Ochoa Torres")
```

Aqui asignamos al valor x la clase Archivo, y llamamos a las funciones que creamos anteriormente.