

Reporte: Clase Archivo.

Oswaldo Córdoba Jimenez 201627085.

El siguiente código genera una clase archivo la cual abre un archivo de texto y hace un conteo de cuantas vocales y consonantes tiene, cuantas líneas, espacios y palabras, entre otros.

Código:

```
import shutil, os

copia = "/home/zentinel/Documents/Programacion Concurrente y Paralela/CodeGit"

class Archivo:
    def __init__(self, nombre):
        try:
            self.f = open(nombre, 'r')
            self.nombre = nombre
        except:
            print('No se puede abrir el archivo ', nombre)
            exit()

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

    def cuentaVocales(self):
        def vocales(s):
            contador = 0
            for i in range(len(s)):
                if s[i].lower() in set("aeiouáéíóú"):
                    contador += 1
            return contador
```

```

contador = 0
for linea in self.f:
    contador += vocales(linea)
self.f.seek(0)
return contador

def cuentaConsonantes(self):
    def consonantes(s):
        contador = 0
        for i in range(len(s)):
            if s[i].lower() in set("bcdfghjklmnpqrstvwxyz"):
                contador += 1
        return contador
    contador = 0
    for linea in self.f:
        contador += consonantes(linea)
    self.f.seek(0)
    return contador

def cuentaMayusculas(self):
    def mayusculas(s):
        contador = 0
        for i in range(len(s)):
            if s[i] in set("AÁBCDEÉFGHIÍJKLMNOÓPQRSTUÚVWXYZ"):
                contador += 1
        return contador
    contador = 0
    for linea in self.f:
        contador += mayusculas(linea)
    self.f.seek(0)
    return contador

```

```
def cuentaMinusculas(self):
def minusculas(s):
contador = 0
for i in range(len(s)):
if s[i] in set("aeiouáéíóúbcdfghjklmnpqrstvwxyz"):
contador += 1
return contador
contador = 0
for linea in self.f:
contador += minusculas(linea)
self.f.seek(0)
return contador

def cuentaEspacios(self):
def espacios(s):
contador = 0
for i in range(len(s)):
if s[i] in set(" "):
contador += 1
return contador
contador = 0
for linea in self.f:
contador += espacios(linea)
self.f.seek(0)
return contador

def cuentaPalabras(self):
text = self.f.read()
pal = text.split()
i = 0
while i < len(pal):
palabra=pal[i]
```

```

if palabra == 'Fin':
    break
i+=1
return i

def cuentaSignos(self):
    def signos(s):
        contador = 0
        for i in range(len(s)):
            if s[i] in set(".,,:;¿?¡!"):
                contador += 1
        return contador
    contador = 0
    for linea in self.f:
        contador += signos(linea)
    self.f.seek(0)
    return contador

def copiaArchivo(self, nombreCopia):
    shutil.copy(self.f.name, nombreCopia)

def copiaMinus(self):
    msm = self.f.read()
    print("Mensaje en minusculas:\n",msm.lower())
    self.f.seek(0)

def copiaMayus(self):
    msm = self.f.read()
    print("Mensaje en mayusculas:\n",msm.upper())
    self.f.seek(0)

def cuentaLineas(self):
    contador = 0

```

```

for linea in self.f:
    contador += 1
    self.f.seek(0)
    return contador

def toHex(self):
    lt = []
    def th(s):
        for i in range(len(s)):
            lt.append(hex(ord(s[i])))

    for linea in self.f:
        th(linea)
    print(lt)
    self.f.seek(0)

if __name__ == '__main__':
    nob = input('que archivo deseas abrir?\n')
    arc = Archivo('/home/zentinel/Documents/Programacion Concurrente y
    Paralela/CodeGit/'+nob)
    arc.muestra()
    print('el texto tiene:', arc.cuentaVocales(), 'vocales')
    print('el texto tiene:', arc.cuentaConsonantes(), 'consonantes')
    print('el texto tiene:', arc.cuentaSignos(), 'signos de puntuacion')
    print('el texto tiene:', arc.cuentaEspacios(), 'espacios')
    print('el texto tiene:', arc.cuentaPalabras(), 'palabras')
    print('el texto tiene:', arc.cuentaLineas(), 'lineas')
    print('el texto tiene:', arc.cuentaMayusculas(), 'mayusculas')
    print('el texto tiene:', arc.cuentaMinusculas(), 'minusculas')
    print('el archivo ha sido copado', arc.copiaArchivo("textoCopia.txt"))
    arc.copiaMayus()
    arc.copiaMinus()
    arc.toHex()

```

La función **“Init()”** es el constructor de la clase el cual abre el archivo en una variable.

La función **“Muestra()”** recorre el texto del archivo e imprime el número de línea y lo que esta escrito en esa línea del texto.

Las funciones **“cuentaVocales()”, “cuentaConsontanes()”, “cuentaSignos()”, “cuentaEspacios()”, “cuentaMayusculas()”** y **“cuentaMinusculas()”** recorren línea por línea y a cada una de estas la recorre carácter por carácter comparando con el conjunto definido dentro de una función **“set()”** y si algún carácter coincide con algún elemento dentro de las búsqueda, se suma un 1 al contador.

La función **“cuentaLineas()”** hace algo parecido a las funciones anteriores solo que lo unico es que recorre todas las líneas del texto y devuelve el índice en el que se quedó, es decir hasta que línea dejo de leer y ese será el número de líneas que tiene nuestro texto.

La función **“cuentaPalabras()”** utiliza la función **“split()”** la cual separa el contenido en palabras, entonces al recorrer el texto, cada posición de este corresponde a una palabra, entcones para saber cuantas palabras se tiene, solo que hay que recorrer el texto separado en palabras y devolver el índice en el que se quedó al final.

La función **“copiaArchivo()”** genera una copia exacta del archivo original con ayuda de la función **“shutil.copy()”**.

Las funciones **“copiaMayus()”** y **“copiaMinus()”** leen el texto y usan la función **“upper()”** para convertir el texto en mayúsculas y la función **“lower()”** para volverlo en minúsculas, y se imprime el resultado de la conversión en pantalla.

Por último, la función **“toHex()”** recorre el texto por línea y a cada línea la recorre por carácter, a cada uno de los caracteres los transforma a su código en ascii con la función **“ord()”** y posteriormente convierte su código a hexadecimal con la función **“hex()”** después de cada transformación, el resultado se agrega en una lista la cual se imprime en pantalla.

Liga del repositorio en GitHub:

<https://github.com/Oswaldcj/CodeConcurrenteYParalela>