



Nombre: Lizet Cristal Varela Coyotl

Matricula: 201637826

Materia: programación paralela y concurrente.



## Clase archivo

Realizar un programa que contenga la clase Archivo, con los métodos

1. Cuenta vocales
2. Cuenta consonantes
3. Cuenta signos de puntuación
4. Cuenta espacios
5. Cuenta palabras
6. Cuenta líneas
7. Mayúsculas
8. Minúsculas
9. Copia archivo
10. Convierte a mayúsculas
11. Convierte a minúsculas
12. Muestra en hexadecimal

### Desarrollo

```
In [7]: 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), end="")
                i+=1
            self.f.seek(0)
```

Como primer lugar creamos la clase principal, en este caso como trabajamos con un archivo, a la clase la nombramos como **class Archivo**, ahí es en donde se encontrará el texto para que se lleve a cabo la realización del programa.

La función **def muestra(self)** mostrará los datos que tiene el archivo que creamos.

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

    contador = 1
    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("bcd fghjklmnpqrstvwxyz"):
                contador += 1
        return contador

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

1. **def cuentaVocales:** esta función realizará el conteo de todas las vocales que se encuentren en el texto del archivo. Para eso indicamos en una sentencia

condicional **if** que evalué el **set** que es la estructura de datos que declaramos anteriormente en el código, para este caso las vocales. Se implementa un contador que inicie en cero, acumule el número de vocales que encontró en el texto y al final nos regrese el número total.

2. **def cuentaConsonantes:** esta función realizará el conteo de todas las consonantes que se encuentren en el texto del archivo. Para eso indicamos en una sentencia condicional **if** que evalué el **set** que es la estructura de datos que declaramos anteriormente en el código, para este caso, las consonantes. Se implementa un contador que inicie en cero, acumule el número de consonantes que encontró en el texto y al final nos regrese el número total.

```
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 cuentaEspacios(self):
    def espacio(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 += espacio(linea)
    self.f.seek(0)
    return contador
```

3. **def cuentaSignos:** esta función realizará el conteo de todos los signos de puntuación que se encuentren en el texto del archivo. Para eso indicamos en una sentencia condicional **if** que evalué el **set** que es la estructura de datos que declaramos anteriormente en el código, para este caso, los signos. Se implementa un contador que inicie en cero, acumule el número de puntos que encontró en el texto y al final nos regrese el número total.
4. **def cuentaEspacios:** esta función realizará el conteo de todos los espacios que se encuentren en el texto del archivo. Para eso indicamos en una sentencia condicional **if** que evalué el **set** que es la estructura de datos que declaramos anteriormente en el código, para este caso, los espacios. Se implementa un contador que inicie en cero, acumule el número de los espacios que encontró en el texto y al final nos regrese el número total.



```
def cuentaPalabras(self):
    def palabra(s):
        contador = 1
        for i in range(len(s)):
            if s[i] in set("([Aa-Zz-])"):
                contador += 1
        return contador

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

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

5. **def cuentaPalabras:** esta función realizará el conteo de todas las palabras que se encuentren en el texto del archivo. Para eso indicamos en una sentencia condicional **if** que evalúe el **set** que es la estructura de datos que declaramos anteriormente en el código, para este caso, las palabras. Se implementa un contador que inicie en cero, acumule el número de las palabras que encontró en el texto y al final nos regrese el número total.
6. **def cuentaLinea:** aquí lo único que hacemos es implementar un contador en cero para el inicio de las líneas se acumula el número de líneas que encontró en el texto y al final nos regrese el número total.

```
def cuentaMayusculas(self):
    def mayusculas(s):
        contador = 0
        for i in range(len(s)):
            if s[i] in set("AABCCDEFGHIJKLMNOPQRSTUVWXYZ"):
                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("aebouaéiôübcdfghjklmnpqrstvwxyz"):
                contador += 1
        return contador

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

7. **def cuentaMayusculas:** esta función realizará el conteo de todas las mayúsculas que se encuentren en el texto del archivo. Para eso indicamos en una sentencia condicional **if** que evalúe el **set** que es la estructura de datos que declaramos anteriormente en el código, para este caso, las letras mayúsculas. Se implementa un contador que inicie en cero, acumule el número de las mayúsculas que encontró en el texto y al final nos regrese el número total.
8. **def cuentaMinusculas:** esta función realizará el conteo de todas las letras minúsculas que se encuentren en el texto del archivo. Para eso indicamos en una sentencia condicional **if** que evalúe el **set** que es la estructura de datos que declaramos anteriormente en el código, para este caso, las minúsculas. Se



implementa un contador que inicie en cero, acumule el número de las letras que encontró en el texto y al final nos regrese el número total.

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

def convierteMayusculas(self):
    contador = ""
    for linea in self.f:
        contador +=linea.upper()
    self.f.seek(0)
    return contador

def convierteMinusculas(self):
    contador = ""
    for linea in self.f:
        contador += linea.lower()
    self.f.seek(0)
    return contador

def convierteHexadecimal(self):
    conv = []
    def hexa(s):
        for i in range(len(s)):
            conv.append(hex(ord(s[i])))
    for linea in self.f:
        hexa(linea)
    print(conv)
    self.f.seek(0)
```

9. **def copiaArchivo:** *shutil.copy()* es un método en Python se utiliza para copiar el contenido del archivo de *origen* en el archivo o directorio de *destino* al que llamaremos “Archivo copiado”.
10. **def convierteMayusculas:** en esta función como ya indicamos quien es el set de mayúsculas lo único que hacemos es que `contador +=linea.upper()` muestre todas letras del texto en mayúsculas.
11. **def convierteMinusculas:** en esta función como ya indicamos quien es el set de minúsculas lo único que hacemos es que `contador += linea.lower()` muestre todas letras del texto en minúsculas.
12. **def convierteHexadecimal:** en esta función únicamente indicamos `conv=[]`

```
#main
nomb = input('Nombre del archivo:')
archivo = Archivo(nomb)
archivo.muestra()
print("\n1-. El número de vocales es: ", archivo.cuentaVocales())
print("\n2-. El número de consonantes es: ", archivo.cuentaConsonantes())
print("\n3-. El número de signos de puntuación es: ", archivo.cuentaSignos())
print("\n4-. El número de espacios es: ", archivo.cuentaEspacios())
print("\n5-. El número de palabras es: ", archivo.cuentaPalabras())
print("\n6-. El número de líneas es: ", archivo.cuentaLinea())
print("\n7-. El número de mayúsculas es: ", archivo.cuentaMayusculas())
print("\n8-. El número de minúsculas es: ", archivo.cuentaMinusculas())
print("\n9-. El archivo copiado es : ", archivo.copiaArchivo("Archivo copiado.txt"))
print("\n10-. El texto en mayúsculas es : ", archivo.convierteMayusculas())
print("\n11-. El texto en minúsculas es : ", archivo.convierteMinusculas())
print("\n12-. El texto en hexadecimal es : ", archivo.convierteHexadecimal())
```

## Main

Aquí es en donde todos los printf del 1 hasta 12 muestran en pantalla un letrero que indica lo que se solicitó realizar en el código, luego del letrero lo que se hace es mandar a llamar cada función y se pone antes el nombre de “archivo.ejemplo()”, ya que todos los datos los estará leyendo del archivo que le indiquemos al programa.



## RESULTADOS

Nombre del archivo: texto.txt

1El hada más hermosa ha sonreído  
2al ver la lumbre de una estrella pálida,  
3que en hilo suave, blanco y silencioso  
4se enrosca al huso de su rubia hermana.  
5  
6¿Y vuelve a sonreír porque en su rueca  
7el hilo de los campos se enmaraña?  
8Tras la tenue cortina de la alcoba  
9está el jardín envuelto en luz dorada.  
10  
11La cuna, casi en sombra. ¡El niño duerme!  
12Dos hadas laboriosas lo acompañan,  
13hilando de los sueños los sutiles  
14copos en rucas de marfil y plata.  
1-. El número de vocales es: 160

2-. El número de consonantes es: 186

3-. El número de signos de puntuación es: 12

4-. El número de espacios es: 72

5-. El número de palabras es: 73

6-. El número de líneas es: 14

7-. El número de mayúsculas es: 6

8-. El número de minúsculas es: 340

9-. El archivo copiado es : None

10-. El texto en mayúsculas es : EL HADA MÁS HERMOSA HA SONREÍDO  
AL VER LA LUMBRE DE UNA ESTRELLA PÁLIDA,  
QUE EN HILO SUAVE, BLANCO Y SILENCIOSO  
SE ENROSCA AL HUSO DE SU RUBIA HERMANA.  
  
¿Y VUELVE A SONREÍR PORQUE EN SU RUECA  
EL HILO DE LOS CAMPOS SE ENMARAÑA?  
TRAS LA TENUE CORTINA DE LA ALCOBA  
ESTÁ EL JARDÍN ENVUELTO EN LUZ DORADA.  
  
LA CUNA, CASI EN SOMBRA. ¡EL NIÑO DUERME!  
DOS HADAS LABORIOSAS LO ACOMPAÑAN,  
HILANDO DE LOS SUEÑOS LOS SUTILES  
COPOS EN RUECAS DE MARFIL Y PLATA.

11-. El texto en minúsculas es: el hada más hermosa ha sonreído  
al ver la lumbre de una estrella pálida,  
que en hilo suave, blanco y silencioso  
se enrosca al huso de su rubia hermana.  
  
¿y vuelve a sonreír porque en su rueca  
el hilo de los campos se enmaraña?  
tras la tenue cortina de la alcoba  
está el jardín envuelto en luz dorada.

la cuna, casi en sombra. ¡el niño duerme!  
dos hadas laboriosas lo acompañan,  
hilando de los sueños los sutiles  
copos en rucas de marfil y plata.

[ '0x45', '0x6c', '0x20', '0x68', '0x61', '0x64', '0x61', '0x20', '0x6d', '0xe1', '0x73', '0x20', '0x68', '0x65', '0x72', '0x6d', '0x6f', '0x61', '0x61', '0x6c', '0x73', '0x61', '0x20', '0x68', '0x61', '0x20', '0x6f', '0xe6', '0x72', '0x65', '0x64', '0x6f', '0x6c', '0x20', '0x76', '0x65', '0x72', '0x20', '0x6c', '0xe1', '0x20', '0x65', '0x6d', '0x62', '0x72', '0x65', '0x20', '0x64', '0x65', '0x20', '0x75', '0x6e', '0x61', '0x20', '0x65', '0x73', '0x74', '0x72', '0x65', '0x6c', '0x6c', '0x61', '0x20', '0x70', '0xe1', '0x6c', '0x69', '0x64', '0xe1', '0x2c', '0xa', '0x71', '0x75', '0x65', '0x20', '0x65', '0x6e', '0x20', '0x68', '0xe9', '0x6c', '0x6f', '0x20', '0x7', '0x75', '0xe1', '0x76', '0xe5', '0x2c', '0x20', '0x62', '0x6c', '0xe1', '0x6e', '0x63', '0x6f', '0x20', '0x79', '0x20', '0x73', '0x69', '0x6c', '0xe5', '0x6e', '0x63', '0xe9', '0x6f', '0x73', '0xe6', '0xa', '0x73', '0x65', '0x20', '0xe5', '0x6e', '0x72', '0xe6', '0x7', '0x63', '0xe1', '0x20', '0xe1', '0x6c', '0x20', '0x68', '0x75', '0x73', '0x6f', '0x20', '0x64', '0x65', '0x20', '0x73', '0x75', '0x2', '0x72', '0x75', '0xe2', '0xe9', '0xe1', '0x20', '0xe8', '0xe5', '0x72', '0x6d', '0xe1', '0x6e', '0xe1', '0xe2', '0xa', '0xa', '0xe', '0x59', '0x20', '0x76', '0x75', '0x65', '0x6c', '0x76', '0x65', '0x20', '0xe1', '0x20', '0x73', '0x6f', '0x6e', '0x72', '0x65', '0xed', '0x72', '0x20', '0x70', '0x6f', '0x72', '0x71', '0x75', '0x65', '0x20', '0xe5', '0x6e', '0x20', '0x73', '0x75', '0x20', '0x72', '0x75', '0x65', '0x63', '0xe1', '0xa', '0xe5', '0x6c', '0x20', '0xe8', '0xe9', '0x6c', '0x6f', '0x20', '0x64', '0xe5', '0x20', '0x6c', '0x6f', '0x73', '0x20', '0xe3', '0xe1', '0x6d', '0x70', '0xe6', '0x73', '0xe5', '0x20', '0xe5', '0x6e', '0xe6', '0x6d', '0xe1', '0x72', '0xe61', '0x20', '0xe3', '0x3f', '0xa', '0x54', '0x72', '0xe1', '0x73', '0x20', '0x6c', '0xe1', '0x20', '0x74', '0xe5', '0xe6', '0x75', '0xe6', '0x5', '0x20', '0xe3', '0x6f', '0x72', '0x74', '0xe9', '0x6e', '0xe1', '0x20', '0xe4', '0xe5', '0x20', '0x6c', '0xe1', '0x20', '0xe1', '0xe6c', '0xe6ff', '0xe2', '0xe1', '0xa', '0xe5', '0x73', '0x74', '0xe1', '0x20', '0xe5', '0x6c', '0x20', '0xe6', '0xe1', '0x72', '0xe54', '0xed', '0xe6', '0x20', '0xe5', '0xe6', '0x76', '0xe5', '0x6f', '0x20', '0xe5', '0xe6', '0x7a', '0x20', '0xe4', '0xe6', '0x72', '0xe1', '0xe4', '0xe1', '0xe2e', '0xa', '0xa', '0x4c', '0xe1', '0x20', '0xe3', '0x75', '0xe6', '0xe1', '0x2c', '0x20', '0xe3', '0xe1', '0x73', '0xe9', '0x20', '0xe5', '0xe6', '0x20', '0x73', '0xe6f', '0xe6d', '0xe2', '0x72', '0xe1', '0xe2e', '0x20', '0xa1', '0x45', '0x6c', '0x20', '0xe6', '0xe9', '0xf1', '0xe6f', '0x20', '0xe4', '0x75', '0xe5', '0x72', '0xe6d', '0xe5', '0x21', '0xa', '0x44', '0x6f', '0x73', '0x20', '0xe8', '0xe1', '0xe4', '0xe1', '0x73', '0x20', '0x6c', '0xe1', '0xe2', '0x6f', '0x72', '0xe69', '0x6f', '0x73', '0xe1', '0x73', '0x20', '0x6c', '0x6f', '0x20', '0xe1', '0xe3', '0x6f', '0xe6d', '0x70', '0xe1', '0xf1', '0xe1', '0xe6e', '0x2c', '0xa', '0xe8', '0xe9', '0xe6c', '0xe1', '0xe6', '0xe4', '0xe6f', '0x20', '0xe4', '0xe5', '0x20', '0xe6c', '0xe6f', '0x73', '0xe20', '0x73', '0x75', '0xe5', '0xf1', '0xe6f', '0x73', '0x20', '0xe6c', '0xe6f', '0x73', '0x20', '0xe3', '0xe7', '0xe9', '0xe6c', '0xe65', '0x73', '0xa', '0xe3', '0xe6f', '0x70', '0xe6f', '0x73', '0x20', '0xe5', '0xe6', '0x20', '0x72', '0xe5', '0xe3', '0xe1', '0x7

<input type="radio"/>		Archivo copiado.txt	Text	Feb 8, 2020
<input type="radio"/>		Archivo.ipynb	Notebook	Feb 8, 2020

12-. El texto en hexadecimal es : None

LINK GITHUB: <https://github.com/lizetvarela/claseArchivo>