

REPORTE DE PROGRAMA ARCHIVO.PY

CODIGO

```
1 from sys import exit #Se importa exit
2 import re #Se importa re
3 import shutil #Se importa shutil para poder copiar el archivo
4
5 class Archivo: #Se crea la clase archivo
6     def __init__(self,nombre): #Se define la clase
7         try: #Si no se puede abrir el archivo no lo hace
8             self.f = open(nombre, 'r') #Se abre el archivo con el modo solo lectura
9             self.nombre=nombre #Se le asigna un nombre
10        except: #Entra la excepcion en caso de que no se pueda abrir el archivo
11            print("No se puede abrir el archivo ", nombre)
12            exit() #Sale del programa
13
14    def muestra(self): #Se define la funcion para mostrar el texto del archivo
15        i = 1 #Variable i se iguala a 1
16        for linea in self.f: #Se usa para recorrer linea por linea todo el archivo
17            print(":{:3}.".format(i,linea)) #Imprime cada linea del archivo con un prefijo igual a el numero de la l
18            i+=1 #Se aumenta i
19        print() #Se imprime un salto de linea
20        self.f.seek(0) #Define a 0 la posicion en el texto del archivo
21
22    def cuentaVocales(self): #Se define la funcion para contar vocales
23    def vocales(s): #Se define la funcion vocales y se asigna a la variable s
24        contador = 0 #Contador igual a 0
25        for i in range(len(s)): #Para recorrer todo el texto del archivo
26            if s[i] in set("aeiouáéíóú"): #Compara si alguna vocal de las que se definen esta en la posicion i del
27                contador += 1 #Aumenta el contador en 1 en caso de que haya alguna vocal
28            print("Vocales: ", contador) #Imprime el numero de vocales que hay en el texto
29            return contador #Regresa el contador
30
31        contador = 0 #Contador igual a 0
32        for linea in self.f: #Recorre el texto linea por linea
33            contador += vocales(linea) #Aumenta el contador en 1
```

```
31        contador = 0 #Contador igual a 0
32        for linea in self.f: #Recorre el texto linea por linea
33            contador += vocales(linea) #Aumenta el contador en 1
34        self.f.seek(0) #Declara a 0 la posicion en el texto del archivo
35        return contador #Regresa el contador
36
37    def cuentaConsonantes(self): #Se define la funcion para contar consonantes
38    def consonantes(s): #Se define la funcion consonantes y se asigna a la variable s
39        contador = 0 #Contador igual a 0
40        for i in range(len(s)): #Para recorrer todo el texto del archivo
41            if s[i] in set("bcdfghjklmnpqrstvwxyzBCDFGHJKLMNPQRSTVWXYZ"): #Compara si alguna consonantes de las
42                contador += 1 #Aumenta el contador en 1 en caso de que haya alguna consonante
43            print("Consonantes: ", contador) #Imprime el numero de consonates que hay en el texto
44            return contador #Regresa el contador
45
46        contador = 0 #Contador igual a 0
47        for linea in self.f: #Recorre el texto linea por linea
48            contador += consonantes(linea) #Aumenta el contador en 1
49            self.f.seek(0) #Declara a 0 la posicion en el texto del archivo
50            return contador #Regresa el contador
51
52    def cuentaEspacios(self): #Se define la funcion para contar espacios
53    def espacios(s): #Se define la funcion espacios y se asigna a la variable s
54        contador = 0 #Contador igual a 0
55        for i in range(len(s)): #Para recorrer todo el texto del archivo
56            if s[i] in set(" "): #Compara si algun espacio esta en la posicion i del texto si es que si entra el i
57                contador += 1 #Aumenta el contador en 1 en caso de que haya algun espacio
58            print("Espacios: ", contador) #Imprime el numero de espacios que hay en el texto
59            return contador #Regresa el contador
60
61        contador = 0 #Contador igual a 0
62        for linea in self.f: #Recorre el texto linea por linea
63            contador += espacios(linea) #Aumenta el contador en 1
64        self.f.seek(0) #Declara a 0 la posicion en el texto del archivo
```

```
63         return contador #Regresa el contador
64
65     def cuentaSignos(self): #Se define la funcion para contar signos de puntuacion
66     def signos(s): #Se define la funcion signos y se asigna a la variable s
67         contador = 0 #Contador igual a 0
68         for i in range(len(s)): #Para recorrer todo el texto del archivo
69             if s[i] in set(".,;:(){}*?[]'\"'-"): #Compara si algun signo esta en la posicion i del texto si es que si
70                 contador += 1 #Aumenta el contador en 1
71             print("Signos de puntuacion: ", contador) #Imprime el numero de signos que hay en el texto
72         return contador #Regresa el contador
73     contador = 0 #Contador igual a 0
74     for linea in self.f: #Recorre el texto linea por linea
75         contador += signos(linea) #Aumenta el contador en 1
76     self.f.seek(0) #Declara a 0 la posicion en el texto del archivo
77     return contador #Regresa el contador
78
79     def cuentaPalabras(self): #Se define la funcion para contar palabras
80     cadena2 = self.f.read() #Se lee el texto del archivo
81     npalabras = 0 #Se define un contador igual a 0
82     np = cadena2.split() #Sirve para dividir el texto por palabras
83     npalabras = len(np) #Cuenta el numero de divisiones
84     print("Cantidad de palabras:", npalabras) #Imprime el numero de palabras que hay en el texto
85
86     def cuentaLineas(self): #Se define la funcion para contar lineas
87     i = 0 #Se crea un contador igual a 0
88     for linea in self.f: #Sirve para que recorra el texto linea por linea
89         i += 1 #Aumenta el contador si es que cambia una linea
90     print("Numero de lineas: ", i) #Imprime el numero de lineas que hay en el texto
91     self.f.seek(0) #Se define el inicio en del texto en la posicion 0
92
93     def cuentaMayusculas(self): #Se define la funcion que cuenta mayusculas
94     def mayusculas(s): #Se define la funcion mayusculas y se le asigna a la variable s
```

```
95     contador = 0 #Contador igual a 0
96     for i in range(len(s)): #Para recorrer todo el texto del archivo
97         if s[i] in set("ABCDEFGHIJKLMNOPQRSTUVWXYZ"): #Compara si alguna letra mayuscula esta en la posicion
98             contador += 1 #Aumenta el contador en 1 en caso de que haya alguna mayuscula
99     print("Mayusculas: ", contador) #Imprime cuantas mayusculas hay en el texto
100     return contador #Regresa el contador
101     contador = 0 #Contador igual a 0
102     for linea in self.f: #Recorre el texto linea por linea
103         contador += mayusculas(linea) #Aumenta el contador en 1
104     self.f.seek(0) #Define el inicio del texto en la posicion 0
105     return contador #Regresa el contador
106
107     def cuentaMinusculas(self): #Se define la funcion cuenta minusculas
108     def minusculas(s): #Se define la funcion minusculas y se le asigna a la variable s
109     contador = 0 #Contador igual a 0
110     for i in range(len(s)): #Para recorrer todo el texto del archivo
111         if s[i] in set("abcdefghijklmnopqrstuvwxyz"): #Compara si alguna letra minuscula esta en la posicion
112             contador += 1 #Aumenta el contador
113     print("Minusculas: ", contador) #Imprime cuantas minusculas hay en el texto
114     return contador #Regresa el contador
115     contador = 0 #Contador igual a 0
116     for linea in self.f: #Recorre el texto linea por linea
117         contador += minusculas(linea) #Aumenta el contador en 1
118     self.f.seek(0) #Define el inicio del texto en la posicion 0
119     return contador #Regresa el contador
120
121     def copyfile(self): #Se define la funcion copyfile
122     fuente = "h:\home\kayle\Escritorio\UNI\Boo\Concurrente\PROGRAMAS/prueba.txt" #Se pone la ruta del archivo a copiar
123     destino = "h:\home\kayle\Escritorio\UNI\Boo\Concurrente\PROGRAMAS/pruebaCOPIA.txt" #Se ingresa la ruta a donde se
124     try: #Si no encuentra el archivo no lo hace
125         shutil.copyfile(fuente, destino) #Se utiliza la funcion shutil con los parametros fuente y destino
126     print("\nARCHIVO COPIADO CORRECTAMENTE\n") #Arroja un mensaje que se a copiado correctamente
```

```
127     except: #Si no encuentra el archivo manda error
128     print("Error") #Imprime error
129
130     def convierteMayuscula(self): #Se define la funcion para convertir todo el texto a mayusculas
131     print("#####\nTEXTO EN MAYUSCULAS:")
132     for linea in self.f: #Recorre el texto linea por linea
133         print("{}\n".format(linea.upper())) #Imprime el texto ya en mayusculas utilizando la funcion upper
134     print() #Imprime un salto de linea
135     self.f.seek(0) #Define el inicio del texto en la posicion 0
136
137     def convierteMinuscula(self): #Se define la funcion para convertir todo el texto a minusculas
138     print("#####\nTEXTO EN MINUSCULAS:")
139     for linea in self.f: #Recorre el texto linea por linea
140         print("{}\n".format(linea.lower())) #Imprime el texto ya en minusculas utilizando la funcion lower
141     print() #Imprime un salto de linea
142     self.f.seek(0) #Define el inicio del texto en la posicion 0
143
144     def Hexadecimal(self): #Se define la funcion hexadecimal
145     s = self.f.read() #Se le asigna a la variable s el texto del archivo
146     print("Texto convertido a hexadecimal\n") #Se imprime un mensaje para saber desde donde convirtio a hexadecimal
147     for i in range(len(s)): #Para recorrer todo el texto
148         print(hex(ord(s[i]))) #Convierte la posicion i del texto a formato hexadecimal
149
150     nomb = input("Nombre del archivo \n") #Se pregunta que archivo se va a abrir
151     archivo = Archivo(nomb) #
152     archivo.muestra() #Muestra el texto del archivo
153     archivo.cuentaVocales() #Muestra el numero de vocales
154     archivo.cuentaConsonantes() #Muestra el numero de consonantes
155     archivo.cuentaEspacios() #Muestra el numero de espacios
156     archivo.cuentaSignos() #Muestra el numero de signos
157     archivo.cuentaPalabras() #Muestra el numero de palabras
158     archivo.cuentaLineas() #Muestra el numero de lineas
159     archivo.cuentaMayusculas() #Muestra cuantas mayusculas hay en el texto
```

```
144     def Hexadecimal(self): #Se define la funcion hexadecimal
145         s = self.f.read() #Se le asigna a la variable s el texto del archivo
146         print("Texto convertido a hexadecimal\n") #Se imprime un mensaje para saber desde donde convirtio a hexadecimal
147         for i in range(len(s)): #Para recorrer todo el texto
148             print(hex(ord(s[i]))) #Convierte la posicion i del texto a formato hexadecimal
149
150 nomb = input("Nombre del archivo \n") #Se pregunta que archivo se va a abrir
151 archivo = Archivo(nomb) #
152 archivo.muestra() #Muestra el texto del archivo
153 archivo.cuentaVocales() #Muestra el numero de vocales
154 archivo.cuentaConsonantes() #Muestra el numero de consonantes
155 archivo.cuentaEspacios() #Muestra el numero de espacios
156 archivo.cuentaSignos() #Muestra el numero de signos
157 archivo.cuentaPalabras() #Muestra el numero de palabras
158 archivo.cuentaLineas() #Muestra el numero de lineas
159 archivo.cuentaMayusculas() #Muestra cuantas mayusculas hay en el texto
160 archivo.cuentaMinusculas() #Muestra cuantas minusculas hay en el texto
161 archivo.copyfile() #Copea el archivo a otro
162 archivo.convierteMayuscula() #Convierte el texto del archivo a pura letra mayuscula
163 archivo.convierteMinuscula() #Convierte el texto del archivo a pura letra minuscula
164 archivo.Hexadecimal() #Convierte el texto del archivo a hexadecimal
```

CORRIDA

```
[kaylee@kayle-pc PROGRAMAS]$ python Archivo.py
Nombre del archivo
prueba.txt
1.-Tras más de 16 años estudiando el universo con luz infrarroja, revelando nuevas maravillas en nuestro sistema solar, nuestra galaxia e incluso más allá, la misión del Telescopio Espacial Spitzer de la NASA ha llegado a su fin.

2.-Los ingenieros de la misión confirmaron a las 2:30 p.m. PST (5:30 p.m. EST) del jueves, 30 de enero de 2020, que la nave había sido puesta en modo seguro, cesando todas las operaciones científicas. Tras la confirmación del cierre definitivo, Joseph Hunt, director de proyectos para Spitzer, declaró el cierre oficial de la misión.

Vocales: 79
Vocales: 108
Consonantes: 101
Consonantes: 139
Espacios: 37
Espacios: 54
Signos de puntuación: 4
Signos de puntuación: 16
Cantidad de palabras: 93
Numero de lineas: 0
Mayusculas: 9
Mayusculas: 11
Minusculas: 171
Minusculas: 231

ARCHIVO COPIADO CORRECTAMENTE

#####
TEXTO EN MAYUSCULAS:
TRAS MAS DE 16 AÑOS ESTUDIANDO EL UNIVERSO CON LUZ INFRARROJA, REVELANDO NUEVAS MARAVILLAS EN NUESTRO SISTEMA SOLAR, NUESTRA GALAXIA E INCLUSO MÁS ALLÁ, LA MISIÓN DEL TELESCOPIO ESPACIAL SPITZER DE LA NASA HA LLEGADO A SU FIN.

LOS INGENIEROS DE LA MISIÓN CONFIRMARON A LAS 2:30 P.M. PST (5:30 P.M. EST) DEL JUEVES, 30 DE ENERO DE 2020, QUE LA NAVE HABÍA SIDO PUESTA EN MODO SEGURO, CESANDO TODAS LAS OPERACIONES CIENTÍFICAS. TRAS LA CONFIRMACIÓN DEL CIERRE DEFINITIVO, JOSEPH HUNT, DIRECTOR DE PROYECTOS PARA SPITZER, DECLARÓ EL CIERRE OFICIAL DE LA MISIÓN.
```

```
#####
TEXTO EN MINUSCULAS:
tras más de 16 años estudiando el universo con luz infrarroja, revelando nuevas maravillas en nuestro sistema solar, nuestra galaxia e incluso más allá, la misión del telescopio espacial spitzer de la nasa ha llegado a su fin.

los ingenieros de la misión confirmaron a las 2:30 p.m. pst (5:30 p.m. est) del jueves, 30 de enero de 2020, que la nave había sido puesta en modo seguro, cesando todas las operaciones científicas. tras la confirmación del cierre definitivo, joseph hunt, director de proyectos para spitzer, declaró el cierre oficial de la misión.

Texto convertido a hexadecimal

0x54
0x72
0x61
0x73
0x20
0x6d
0xe1
0x73
0x20
0x64
0x65
0x20
0x31
0x36
0x20
0x61
0xf1
0x6f
0x73
0x20
0x65
0x73
0x74
0x75
```