

Nombre: Jared Valenzuela

```
1 PRIMERA PREGUNTA

In [ ]: 1

In [ ]: 1 SEGUNDA PREGUNTA

In [7]: 1 frutas={'manzana','pera','naranja','sandia','naranja','banana'}
2 print(frutas)
3
{'manzana', 'sandia', 'naranja', 'pera', 'banana'}

In [5]: 1 conjunto={1,2,3,4,5}
2 print(conjunto)
{1, 2, 3, 4, 5}

In [63]: 1 fre_palabras = {} # Diccionario vacío

1 TERCERA PREGUNTA

In [11]: 1 archivo = open('jhon.txt', 'r', encoding='ANSI') # Abrir archivo
2 mensaje = archivo.read().lower() # Convertir a minúsculas
3 archivo.close()

In [13]: 1 palabras = mensaje.split() # Separar el texto en palabras
2 palabras_unicas = set(palabras) # Obtener las palabras únicas utilizando un conjunto

In [14]: 1 fre_palabras = {} # Diccionario vacío
```

```
1 TERCERA PREGUNTA

In [11]: 1 archivo = open('jhon.txt', 'r', encoding='ANSI') # Abrir archivo
2 mensaje = archivo.read().lower() # Convertir a minúsculas
3 archivo.close()

In [13]: 1 palabras = mensaje.split() # Separar el texto en palabras
2 palabras_unicas = set(palabras) # Obtener las palabras únicas utilizando un conjunto

In [14]: 1 fre_palabras = {} # Diccionario vacío

In [ ]: 1

In [ ]: 1

In [15]: 1
2 >(".,[...]{...} +*-" ) # Convertir a minúsculas y eliminar signos de puntuación
3
4 # Si la palabra ya está en el diccionario, aumentar su contador para indicar que se encontró una ocurrencia más
5
6 # Si la palabra no está en el diccionario, agregarla con un contador inicial de 1 para indicar que se encontró una ocurrencia
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99

In [17]: 1 # Ordenar las palabras alfabéticamente mediante el método sorted
2 palabras_ordenadas = sorted(fre_palabras.items()) # El método items() devuelve una lista de tuplas (clave, valor)

In [19]: 1 # Imprimir los resultados
2 print("Listado de palabras y frecuencias:")
3 listado="" # Variable para almacenar el listado de palabras y frecuencias
4 for palabra, frecuencia in palabras_ordenadas: # Recorrer la lista de tuplas
5     print(f"[{palabra}] - {frecuencia}") # Formato de salida de cada palabra y su frecuencia
```

```

4 for palabra, frecuencia in palabras_ordenadas: # Recorrer la lista de tuplas
5     print(f"[{palabra}] - {fre_palabras[palabra]}") # Formato de salida de cada palabra y su frecuencia

```

```

[wired - 1]
[wisdom: - 1]
[wizards - 4]
[wolfgang - 1]
[women - 1]
[wood - 3]
[wooden - 8]
[wooden: - 1]
[woodrow - 2]
[wooten! - 1]
[wootten - 19]
[wootten? - 1]
[world - 2]
[worldwide - 1]
[worst - 1]
[worthy - 1]
[wright - 2]
[www.caribebetania.com - 1]
[wyatt - 1]
[xx - 8]

```

```

In [20]: 1 archivo = open('leyes.txt', 'r', encoding='ANSI') # Abrir archivo
2         mensaje = archivo.read().lower() # Convertir a minúsculas
3         archivo.close()

```

```

In [21]: 1 palabras = mensaje.split() # Separar el texto en palabras
2         palabras_unicas = set(palabras) # Obtener las palabras únicas utilizando un conjunto

```

```

In [22]: 1 fre_palabras = {} # Diccionario vacío

```

```

In [23]: 1
2         >(".,[...]()...() +*~") # Convertir a minúsculas y eliminar signos de puntuación
3
4         # Si la palabra va está en el diccionario, aumentar su contador para indicar que se encontró una ocurrencia más

```

```

In [24]: 1 # Ordenar las palabras alfabéticamente mediante el método sorted
2         palabras_ordenadas = sorted(fre_palabras.items()) # El método items() devuelve una lista de tuplas (clave, valor)

```

```

In [25]: 1 # Imprimir los resultados
2         print("Listado de palabras y frecuencias:")
3         listado="" # Variable para almacenar el listado de palabras y frecuencias
4         for palabra, frecuencia in palabras_ordenadas: # Recorrer la lista de tuplas
5             print(f"[{palabra}] - {fre_palabras[palabra]}") # Formato de salida de cada palabra y su frecuencia

```

```

[$300 - 1]
[$350.000 - 1]
[$4 - 2]
[$4,200 - 1]
[$4.000 - 1]
[$400.00 - 1]
[$47 - 1]
[$5 - 2]
[$5.000 - 1]
[$50.000 - 1]
[$500 - 1]
[$7 - 1]
[$7.5 - 1]
[$8 - 1]
[$80 - 1]
[$800.000 - 1]
[$925 - 1]
[8 - 2]
[0,535 - 1]
[0,759 - 1]

```

```
In [26]: 1 archivo = open('maneras.txt', 'r', encoding='ANSI') # Abrir archivo
2 mensaje = archivo.read().lower() # Convertir a minúsculas
3 archivo.close()
```

```
In [27]: 1 palabras = mensaje.split() # Separar el texto en palabras
2 palabras_unicas = set(palabras) # Obtener las palabras únicas utilizando un conjunto
```

```
In [28]: 1 fre_palabras = {} # Diccionario vacío
```

```
In [29]: 1
2 >(".,[...]{...} +*~") # Convertir a minúsculas y eliminar signos de puntuación
3
4 # Si la palabra ya está en el diccionario, aumentar su contador para indicar que se encontró una ocurrencia más
5
6 # Si la palabra no está en el diccionario, agregarla con un contador inicial de 1 para indicar que se encontró una ocurrencia
```

```
In [30]: 1 # Ordenar las palabras alfabéticamente mediante el método sorted
2 palabras_ordenadas = sorted(fre_palabras.items()) # El método items() devuelve una lista de tuplas (clave, valor)
```

```
In [31]: 1 # Imprimir los resultados
2 print("Listado de palabras y frecuencias:")
3 listado="" # Variable para almacenar el listado de palabras y frecuencias
4 for palabra, frecuencia in palabras_ordenadas: # Recorrer la lista de tuplas
5     print(f"[palabra] - {fre_palabras[palabra]}") # Formato de salida de cada palabra y su frecuencia
```

```
[141 - 1]
[142 - 1]
[143 - 1]
[144 - 1]
[145 - 1]
```

```
In [32]: 1 archivo = open('abe.txt', 'r', encoding='ANSI') # Abrir archivo
2 mensaje = archivo.read().lower() # Convertir a minúsculas
3 archivo.close()
```

```
In [33]: 1 palabras = mensaje.split() # Separar el texto en palabras
2 palabras_unicas = set(palabras) # Obtener las palabras únicas utilizando un conjunto
```

```
In [34]: 1 fre_palabras = {} # Diccionario vacío
```

```
In [35]: 1
2 >(".,[...]{...} +*~") # Convertir a minúsculas y eliminar signos de puntuación
3
4 # Si la palabra ya está en el diccionario, aumentar su contador para indicar que se encontró una ocurrencia más
5
6 # Si la palabra no está en el diccionario, agregarla con un contador inicial de 1 para indicar que se encontró una ocurrencia
```

```
In [36]: 1 # Ordenar las palabras alfabéticamente mediante el método sorted
2 palabras_ordenadas = sorted(fre_palabras.items()) # El método items() devuelve una lista de tuplas (clave, valor)
```

```
In [37]: 1 # Imprimir los resultados
2 print("Listado de palabras y frecuencias:")
3 listado="" # Variable para almacenar el listado de palabras y frecuencias
4 for palabra, frecuencia in palabras_ordenadas: # Recorrer la lista de tuplas
5     print(f"[palabra] - {fre_palabras[palabra]}") # Formato de salida de cada palabra y su frecuencia
```

```
[10 - 10]
[11 - 9]
[111 - 1]
[12 - 9]
[124 - 1]
[13 - 9]
```

```
In [38]: 1 archivo = open('actitud.txt', 'r', encoding='ANSI') # Abrir archivo
2 mensaje = archivo.read().lower() # Convertir a minúsculas
3 archivo.close()
```

```
In [39]: 1 palabras = mensaje.split() # Separar el texto en palabras
2 palabras_unicas = set(palabras) # Obtener las palabras únicas utilizando un conjunto
```

```
In [40]: 1 fre_palabras = {} # Diccionario vacío
```

```
In [41]: 1
2 >(".,[...]{...} +*~") # Convertir a minúsculas y eliminar signos de puntuación
3
4 # Si la palabra ya está en el diccionario, aumentar su contador para indicar que se encontró una ocurrencia más
5
6 # Si la palabra no está en el diccionario, agregarla con un contador inicial de 1 para indicar que se encontró una ocurrencia
```

```
In [42]: 1 # Ordenar las palabras alfabéticamente mediante el método sorted
2 palabras_ordenadas = sorted(fre_palabras.items()) # El método items() devuelve una lista de tuplas (clave, valor)
```

```
In [43]: 1 # Imprimir los resultados
2 print("Listado de palabras y frecuencias:")
3 listado="" # Variable para almacenar el listado de palabras y frecuencias
4 for palabra, frecuencia in palabras_ordenadas: # Recorrer la lista de tuplas
5     print(f"[palabra] - {fre_palabras[palabra]}") # Formato de salida de cada palabra y su frecuencia
```

```
[obtener - 1]
[ocurrencia - 1]
```

```

In [44]: 1 archivo = open('oro.txt', 'r', encoding='ANSI') # Abrir archivo
          2 mensaje = archivo.read().lower() # Convertir a minúsculas
          3 archivo.close()

In [45]: 1 palabras = mensaje.split() # Separar el texto en palabras
          2 palabras_unicas = set(palabras) # Obtener las palabras únicas utilizando un conjunto

In [47]: 1 fre_palabras = {} # Diccionario vacío

In [48]: 1
          2 >(".,[...]{...} +*-" # Convertir a minúsculas y eliminar signos de puntuación
          3
          4 # Si la palabra ya está en el diccionario, aumentar su contador para indicar que se encontró una ocurrencia más
          5
          6 # Si la palabra no está en el diccionario, agregarla con un contador inicial de 1 para indicar que se encontró una ocurrencia

In [49]: 1 # Ordenar las palabras alfabéticamente mediante el método sorted
          2 palabras_ordenadas = sorted(fre_palabras.items()) # El método items() devuelve una lista de tuplas (clave, valor)

In [50]: 1 # Imprimir los resultados
          2 print("Listado de palabras y frecuencias:")
          3 listado="" # Variable para almacenar el listado de palabras y frecuencias
          4 for palabra, frecuencia in palabras_ordenadas: # Recorrer la lista de tuplas
          5     print(f"[palabra] - {fre_palabras[palabra]}") # Formato de salida de cada palabra y su frecuencia

[12 - 4]
[12.884 - 2]
[123 - 1]
[124 - 1]
[1267 - 1]
[13 - 6]

```

```

In [52]: 1 archivo = open('personas.txt', 'r', encoding='ANSI') # Abrir archivo
          2 mensaje = archivo.read().lower() # Convertir a minúsculas
          3 archivo.close()

In [53]: 1 palabras = mensaje.split() # Separar el texto en palabras
          2 palabras_unicas = set(palabras) # Obtener las palabras únicas utilizando un conjunto

In [54]: 1 fre_palabras = {} # Diccionario vacío

In [55]: 1
          2 >(".,[...]{...} +*-" # Convertir a minúsculas y eliminar signos de puntuación
          3
          4 # Si la palabra ya está en el diccionario, aumentar su contador para indicar que se encontró una ocurrencia más
          5
          6 # Si la palabra no está en el diccionario, agregarla con un contador inicial de 1 para indicar que se encontró una ocurrencia

In [56]: 1 # Ordenar las palabras alfabéticamente mediante el método sorted
          2 palabras_ordenadas = sorted(fre_palabras.items()) # El método items() devuelve una lista de tuplas (clave, valor)

In [57]: 1 # Imprimir los resultados
          2 print("Listado de palabras y frecuencias:")
          3 listado="" # Variable para almacenar el listado de palabras y frecuencias
          4 for palabra, frecuencia in palabras_ordenadas: # Recorrer la lista de tuplas
          5     print(f"[palabra] - {fre_palabras[palabra]}") # Formato de salida de cada palabra y su frecuencia

[121 - 1]
[122 - 1]
[123 - 1]

```

```
In [58]: 1 archivo = open('vive.txt', 'r', encoding='ANSI') # Abrir archivo
2 mensaje = archivo.read().lower() # Convertir a minúsculas
3 archivo.close()
```

```
In [59]: 1 palabras = mensaje.split() # Separar el texto en palabras
2 palabras_unicas = set(palabras) # Obtener las palabras únicas utilizando un conjunto
```

```
In [60]: 1
2 o(",,[...]{...}() +*") # Convertir a minúsculas y eliminar signos de puntuación
3
4 # Si la palabra ya está en el diccionario, aumentar su contador para indicar que se encontró una ocurrencia más
5
6 # Si la palabra no está en el diccionario, agregarla con un contador inicial de 1 para indicar que se encontró una ocurrencia
```

```
In [61]: 1 # Ordenar las palabras alfabéticamente mediante el método sorted
2 palabras_ordenadas = sorted(fre_palabras.items()) # El método items() devuelve una lista de tuplas (clave, valor)
```

```
In [62]: 1 # Imprimir los resultados
2 print("Listado de palabras y frecuencias:")
3 listado="" # Variable para almacenar el listado de palabras y frecuencias
4 for palabra, frecuencia in palabras_ordenadas: # Recorrer la lista de tuplas
5     print(f"{palabra} - {fre_palabras[palabra]}") # Formato de salida de cada palabra y su frecuencia
```

Listado de palabras y frecuencias:

```
[ - 929]
[! - 11]
[!"i - 1]
```