

Ayudantías

Fundamentos de Programación

Semana 11

Diccionarios y conjuntos

Ejercicio 1

Se necesita crear un programa para generar las estadísticas de las palabras en un texto. Por ejemplo, a partir del siguiente texto:

```
Con la ayuda de un grupo de amigos y de valientes aliados Frodo emprende un peligroso viaje con la
misión de destruir el Anillo Único Pero el Señor Oscuro Sauron quien creara el Anillo envía a sus
servidores para perseguir al grupo Si Sauron lograra recuperar el Anillo sería el final de la Tierra
Media Ganadora de cuatro Oscars este inmortal relato sobre el bien y el mal la amistad y el sacrificio
te transportará a un mundo más allá de tu imaginación
...
```

su programa generaría el siguiente diccionario, apoyándose en las cinco funciones que se detallan abajo. Asuma que **cada palabra tendrá 20 caracteres como máximo** y que el **número máximo de palabras por líneas es 30**. También tiene una lista **stopwords** = ['la', 'con', ...] con palabras que no agregan mayor significado al texto.

<pre>{ 'NTP': 83, 'NPC': 22, 'palabras': { 'Anillo': {'veces': 3, 'NL': (2,3,4)}, 'Sauron': {'veces': 2, 'NL': (2,3) }, ... } }</pre>	<p>Donde,</p> <p>NTP = Número total de palabras NPC = Número solo de palabras stopwords NL = Número de líneas</p>
---	--

Ahora implemente:

- 1) La función **cargarArchivo(nombre)**, que leerá el texto desde el archivo nombre y creará una matriz M de NumPy (**con dtype='U20'**) donde cada fila representa una línea, y cada columna, una palabra de dicha línea. Si la línea tiene menos de 30 palabras, las celdas restantes deben ser llenadas con un string vacío. Cada línea del archivo está limitada por '\n'. Cada palabra está separada por un espacio en blanco.
- 2) La función **ocurrencias(palabra, M)**, que devuelve el número de veces que el string palabra aparece en la matriz M.

- 3) La función **líneas(palabra, M)**, que devuelve una tupla con los números de fila donde aparece palabra en M.
- 4) La función **contarPalabras(M, stopwords)** que devuelve una tupla con el número total de palabras del texto (**incluyendo las palabras stopwords**) y el número **solo** de palabras stopwords. **Cada palabra (regular o stopwords) debe ser contada una sola vez** sin importar cuantas veces se repita en el texto.
- 5) Implemente la función **concordancia(M, stopwords)** que devuelve un diccionario con las estadísticas de las palabras (ver ejemplo arriba). El diccionario interno '**palabras**' **no debe incluir las palabras stopwords**.

Ejercicio 2

Para administrar la nueva matriz energética del Ecuador, Ud. tiene un diccionario con la información de las plantas de energía y a las ciudades que atienden. Cada ciudad tiene: una tupla con los consumos mensuales (12) del año en megavatios-hora (MWh) y la tarifa de consumo en dólares por megavatio-hora (MWh) que le cobra la planta eléctrica. Una ciudad puede estar servida por más de una planta eléctrica. No todas las ciudades son servidas por todas las plantas eléctricas.

```
consumo_energia = {
    'Coca Codo Sinclair': {
        'Quito': { 'consumos': (400, 432, ..., 213), 'tarifa': 65},
        'Guayaquil': { 'consumos': (120, 55, 32, ..., 70), 'tarifa': 84},
        ...
    },
    'Sopladora': {
        'Guayaquil': { 'consumos': (310, 220, 321, ..., 200), 'tarifa': 55},
        'Quito': { 'consumos': (400, 432, ..., 587), 'tarifa': 79},
        'Loja': { 'consumos': (50, 32, 32, ..., 40), 'tarifa': 32},
        ...
    },
    ...
}
```

Además, dispone del siguiente diccionario con información de ciudad por región:

```
informacion = {
    'costa': ('Guayaquil', 'Manta', ...)
    'sierra': ('Quito', 'Ambato', ...)
    'oriente': ('Tena', 'Nueva Loja', ...)
}
```

Implemente lo siguiente:

- 1) Una función **total_anual(consumo_energia, planta, ciudad)** que recibe el diccionario **consumo_energia**, el nombre de una planta y el nombre de una ciudad. La función debe calcular y retornar el total anual de megavatios-hora servido por planta a ciudad.
- 2) Una función **total_plantas_ciudad(consumo_energia, ciudad)** que recibe el diccionario **consumo_energia** y el nombre de una ciudad. La función debe devolver un diccionario cuyas claves corresponden a los nombres de las plantas generadoras que proveen energía a **ciudad** y los valores corresponden al total anual de megavatios-hora servido por cada planta a **ciudad**.
- 3) Una función **megavatios_hora(consumo_energia, informacion)** que recibe el diccionario **consumo_energia** y el diccionario **informacion**. La función retorna el total anual de megavatios-hora consumido por todas las ciudades de la región COSTA.
- 4) Una función **facturacion(consumo_energia)** que recibe el diccionario **consumo_energia** y genera un archivo con la facturación total en dólares de los seis primeros meses de cada planta generadora. El archivo resultante se llamará **facturacion.txt** y tendrá la siguiente estructura:

```
Planta,enero,febrero,marzo,...,junio
Coca Codo Sinclair,2903,2145,3010,...,2945
Sopladora,3102,3234,3223,...,3417
...
```

Ejercicio 3

Para el intercambio comercial entre países se tienen archivos con las transacciones de venta. Las transacciones están almacenadas en un archivo diferente por cada categoría (Flores, Frutas, Maderas, etc.) de producto. Por ejemplo, las ventas de claveles, rosas, tulipanes, girasoles, etc. están almacenadas en el archivo “Flores.txt”; las ventas de cacao, banana, etc. están almacenadas en el archivo “Frutas.txt”

Flores.txt

```
Comprador,Vendedor,Producto,UnidadesVendidas,VentasEn$,Fecha
Estados Unidos,Ecuador,rosas,59284,631432.21,2018-01-10
Holanda,Japon,tulipanes,2384,12434.87,2017-11-22
...
Estados Unidos,Ecuador,girasoles,38284,331432.75,2018-02-01
```

Note que un país puede vender el mismo producto al mismo comprador pero en una fecha diferente.

Desarrolle lo siguiente:

1. La función **calculaTotales(categoria)** que recibe el nombre de una categoría. La función deberá leer el archivo para esa categoría y retornar un diccionario de totales con la siguiente estructura:

```
totales = {(comprador,vendedor,producto):totalDolares}
```

Por ejemplo, si en la categoría “Flores” Estados Unidos le compró a Ecuador rosas en 12 fechas diferentes, en el diccionario deberá aparecer un solo item con las sumas totales de esas 12 transacciones.

```
{('Estados Unidos','Ecuador','rosas'):32645788}
```

2. La función **consolidado(nomArchivo, categorias)** que recibe una lista de categorías y genera un archivo con nombre **nomArchivo** en el que se listen todos los totales de ventas acumulados por Comprador, Vendedor, Producto. El archivo tendrá la siguiente estructura:

```
Comprador,Vendedor,Categoria,Producto,VentasTotalesDolares
```

Para el resto del ejercicio asuma que tiene una función crearMatriz que recibe el nombre del archivo consolidado y devuelve una tupla con tres elementos: (1) matriz M cuyas filas representan países compradores, columnas representan productos ordenados alfabéticamente dentro de cada categoría y las celdas representan ventas totales en dólares, (2) lista con las etiquetas de las filas y (3) lista con las etiquetas de las columnas. Las categorías no están ordenadas alfabéticamente.

3. La función **comprasCategorias(nomArchivo, dicCat)** que recibe el nombre del archivo consolidado y un diccionario donde las claves son las categorías y los valores son listas con todos los productos ordenados alfabéticamente dentro de cada categoría. La función deberá generar por cada categoría un archivo con el nombre de la categoría y extensión “.txt”. Cada archivo debe contener los **5 países que más han comprado para esa categoría**. Cada archivo tendrá la siguiente estructura: