



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

<i>Profesor:</i>	Jorge Solano
<i>Asignatura:</i>	Estructura de Datos y Algoritmos II
<i>Grupo:</i>	2
<i>No de Práctica(s):</i>	10
<i>Integrante(s):</i>	Martínez Ostoa Néstor Iván
<i>Semestre:</i>	2019-1
<i>Fecha de entrega:</i>	24/Octubre/2018
<i>Observaciones:</i>	

CALIFICACIÓN: _____

Práctica 10: Archivos

Introducción:

Los archivos son la unidad mínima de almacenamiento de información a largo plazo. A pesar de ser un invento/concepto ya viejo, siguen siendo bastante útiles. Especialmente para la parte de parsear información en donde requerimos leer desde un archivo y manejar la información en clases.

Objetivos:

1. Crear una función que permita obtener los datos almacenados en un archivo XML con el siguiente formato para guardarlos dentro de una lista de diccionarios.

```
<record>
    <user>Matsoft</user>

    <password>Avery</password>

    <email>achapleo0@npr.org</email>

</record>
```

2. Una vez terminada la búsqueda, generar las siguientes estadísticas y guardarlas en disco:

- Número y lista de passwords iguales.
- Número y password con mayor incidencia.
- Número de empresas (dependiendo del correo).
- Número y lista de correos iguales.

3. Graficar el tiempo de ejecución de las funciones anteriores.

4. Generar una gráfica de tipo “Pie” con los números obtenidos de las funciones.

Desarrollo

1. Código en Python

```

def getElements(bSequence, eSequence, offSet):
    listToUse = []
    try:
        file = open(PATH, "r", encoding='utf-8')
        for line in file:
            beginning = line.find(bSequence)
            end = line.find(eSequence)
            if beginning >= 0:
                listToUse.append(line[beginning+offSet:end])
    except:
        print('No fue posible abrir el archivo -> ' + str(PATH))
    finally:
        if file:
            file.close()
    return(listToUse)
generateGraph(finalList)
generateGraphMail(finalList)

main()

```

```

def getListOfDictionaries(users,passwords,emails,keys):
    listOfDictionaries = []
    for i in range(len(users)):
        user = []
        user.append(users[i])
        user.append(passwords[i])
        user.append(emails[i])
        listOfDictionaries.append(dict(zip(keys, user)))
    return listOfDictionaries

def generateStatistics(ld):
    print("\n---STATISTICS---\n")

    equalP= getDicOfEquals('password',ld)
    repeatedElements,mostRepeatedKey,numberOfItems = getMostRepeatedItem(equalP)
    generatePieChart(equalP, 'password', mostRepeatedKey)
    printNicely(equalP, 'Password', repeatedElements, mostRepeatedKey, numberOfItems)

    equalM = getDicOfEquals('email',ld)
    repeatedElements,mostRepeatedKey,numberOfItems = getMostRepeatedItem(equalM)
    generatePieChart(equalM, 'email', mostRepeatedKey)
    printNicely(equalM, 'Email', repeatedElements, mostRepeatedKey, numberOfItems)

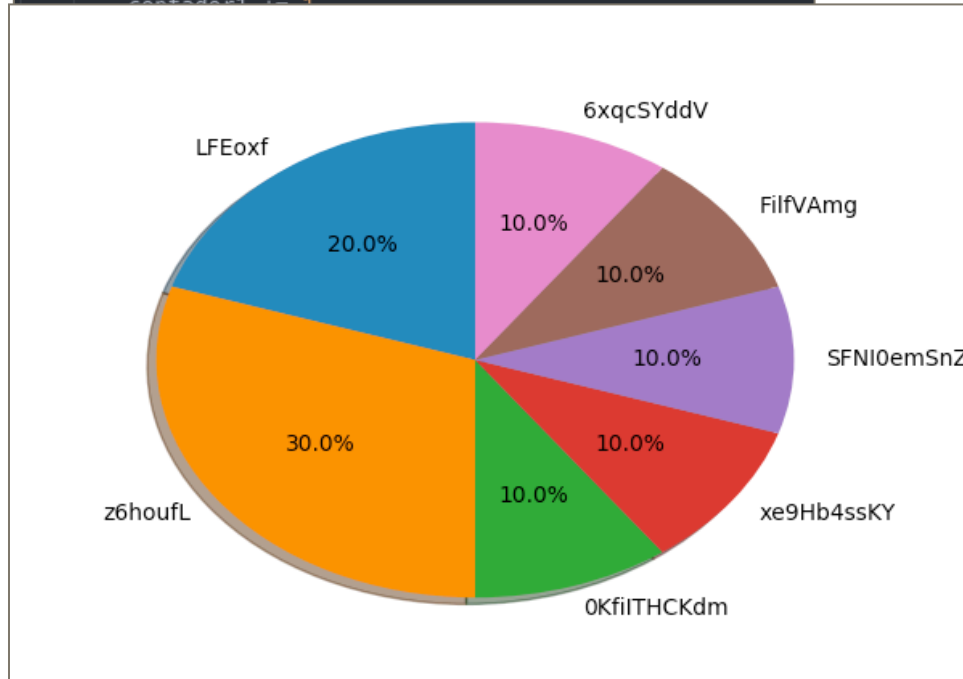
    numberOfCompanies,companies = getNumberCompanies()
    print("Number of companies: \n" + str(numberOfCompanies) + "\n")
    companiesPieChart(companies, 'Compañías')
    printCompanies(companies)

```

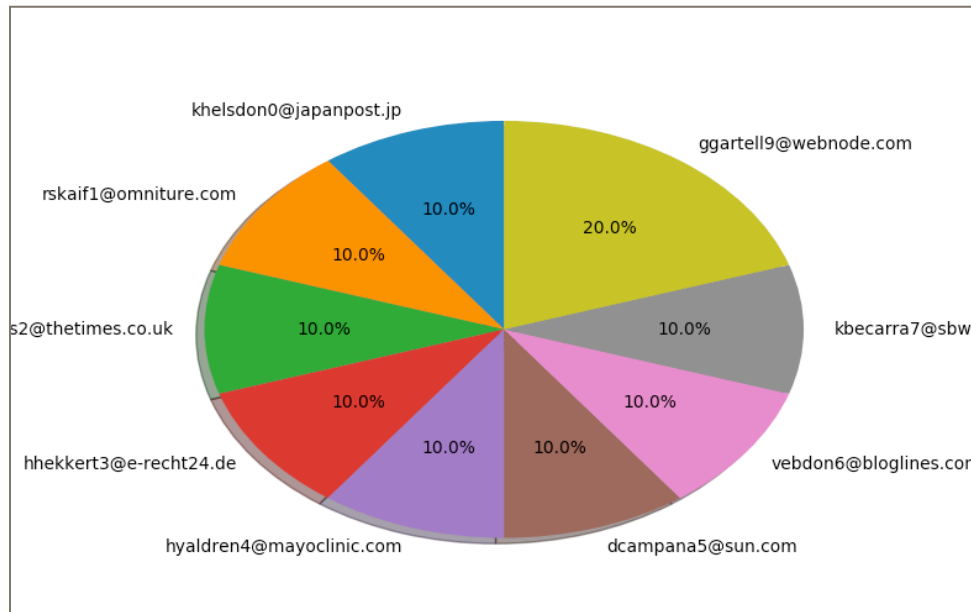
2. Gráficas de Pie

Passwords

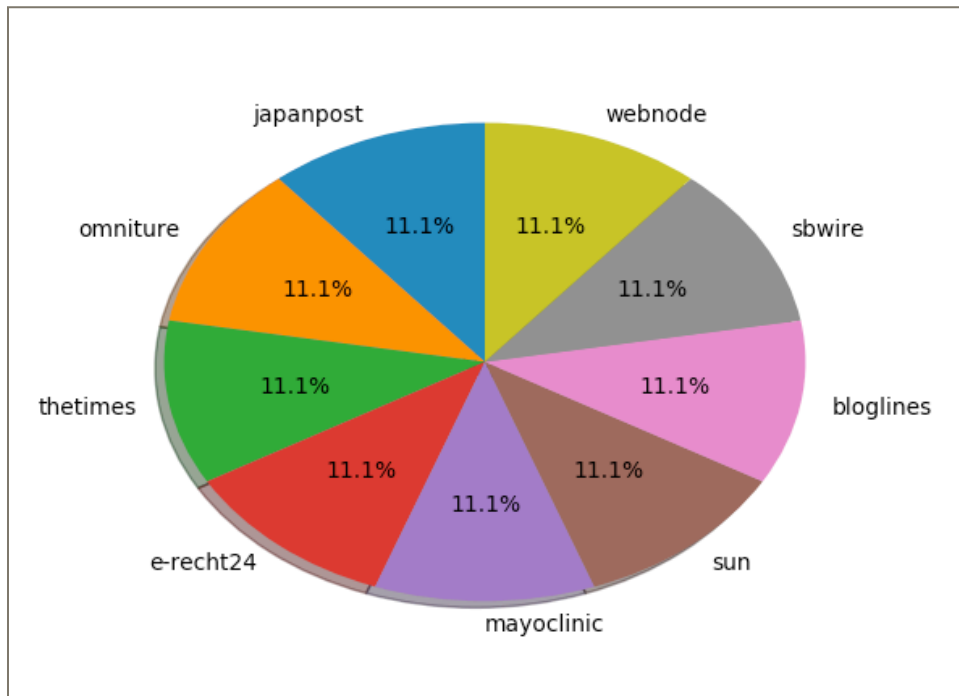
```
def getDicOfEquals(element,ld):  
    global contador1  
    listOfElements = {}  
    for u in ld:  
        contador1 += 1
```



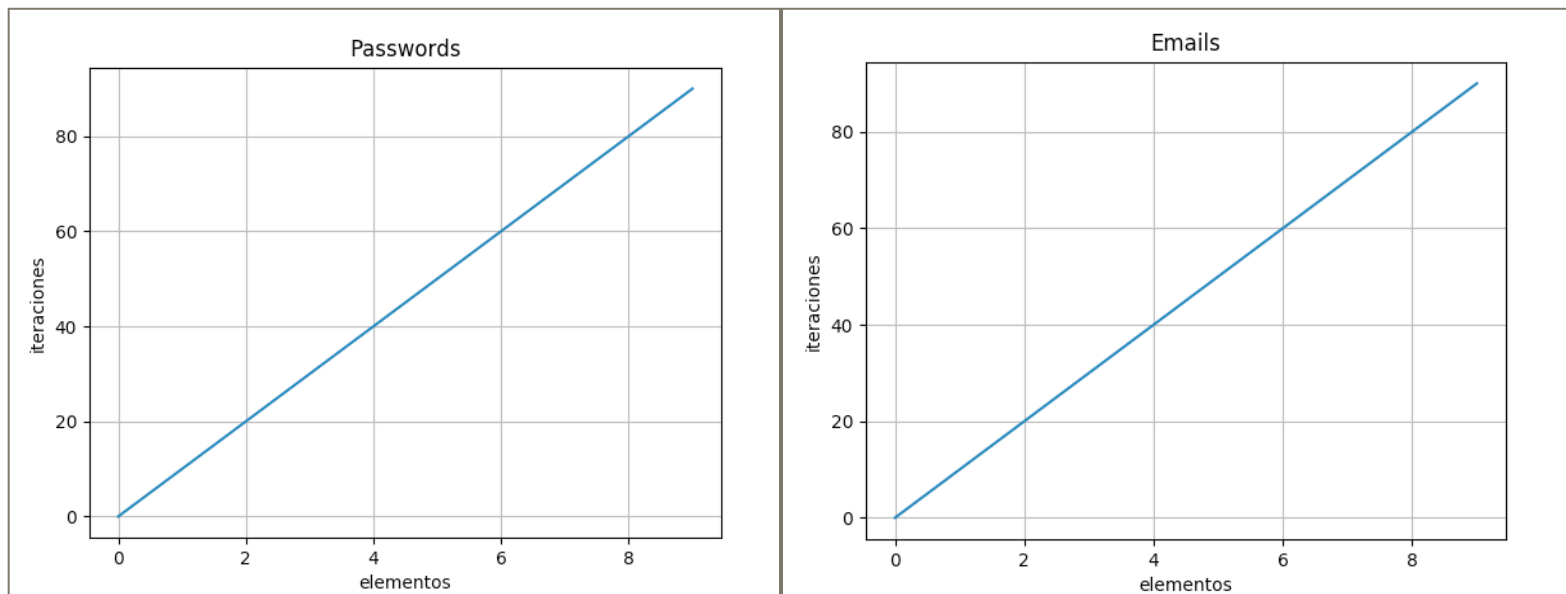
Emails



Compañías



3. Gráficas de funciones



4. Archivos

Password.txt

```
Most repeated Password: z6houfL
Times: 3
[(2, 'LFEoxf'), (3, 'z6houfL')]
```

Email.txt

```
Most repeated Email: ggartell9@webnode.com
Times: 2
[(2, 'ggartell9@webnode.com')]
```

Compañía.txt

```
Most repeated Email: ggartell9@webnode.com
Times: 2
[(2, 'ggartell9@webnode.com')]
```

5. Prueba de funcionamiento

Lista de diccionarios obtenidos a partir del archivo xml

```
List of dictionaries
[{'user': 'Karie', 'email': 'khelsdon0@japanpost.jp', 'password': 'LFEoxf'}, {'user': 'Renell', 'email': 'rskaif1@omniture.com', 'password': 'LFEoxf'}, {'user': 'Porter', 'email': 'pferres2@thetimes.co.uk', 'password': 'z6houfL'}, {'user': 'Hadlee', 'email': 'hhekkert3@e-recht24.de', 'password': 'z6houfL'}, {'user': 'Hillel', 'email': 'hyaldren4@mayoclinic.com', 'password': 'z6houfL'}, {'user': 'Dex', 'email': 'dcampana5@sun.com', 'password': '0KfiITHCKdm'}, {'user': 'Victoir', 'email': 'vebdon6@bloglines.com', 'password': 'xe9Hb4ssKY'}, {'user': 'Kaylyn', 'email': 'kbecarra7@sbwire.com', 'password': 'SFNI0emSnZJy'}, {'user': 'Conney', 'email': 'ggartell9@webnode.com', 'password': 'FilfVAmg'}, {'user': 'Garrard', 'email': 'ggartell9@webnode.com', 'password': '6xqcSYddV'}]
```

```
Number of users
10
```

Conclusiones

El pasado mes (Septiembre, 2018) durante el Hackaton de Bancomer era muy común encontrar retos que requerían del proceso de información partiendo de un archivo ya sea .csv o .xml. Esta práctica facilita mucho el manejo de archivos para tenerlos en un formato conveniente dependiendo de nuestra forma de procesarlo.