

Instrucciones

- Lea con detenimiento cada una de las actividades a realizar durante la experiencia.
- Cree un archivo con extensión **.py** con lo desarrollado. El nombre del archivo debe tener el siguiente formato: **TEL101_C5_Nombre_Apellido.py** (Ej. TEL101_C5_Nicolas_Galvez.py), sin incluir tildes.
- Enviar el archivo a través de la página de aula del ramo, sección “Control 5” hasta las 11:39:59 del presente día 10/08/2020. **No se aceptarán entregas atrasadas.**
- Utilice solo herramientas aprendidas en clases y presentes en las diapositivas del curso. **No utilice códigos genéricos obtenidos por búsqueda en la web.**
- Sea riguroso con las instrucciones de desarrollo.
- ¡Éxito!

1 Elecciones CEETEL

1. El Centro de Estudiantes de Telemática (CEETEL) está ad portas de una nueva elección de los representantes de los alumnos de la Carrera de Ingeniería Civil Telemática de la UPyTFSM, la universidad más prestigiosa de Pyland. Para esta instancia, el TRICEL ha registrado las listas postulantes con el nombre de los candidatos a presidente de cada una de ellas. Esta información la manejan en un archivo llamado **candidatos.txt**.

candidatos.txt

```
SophyPeace  
ElNelson  
JoacoHastaLaProxima  
Dudu  
Oscar3Names  
...
```

Cada línea de este archivo representa a un candidato a presidente que encabeza una lista. El archivo contiene muchos más datos que los que se muestran. El CEETEL, además, ha implementado una novedosa forma de votación: utilizando los comentarios efectuados por los estudiantes en la red social PyBook, en la entrada (post, topic, publicación, etc.) de las elecciones. Los usuarios fueron previamente validados, y además los comentarios para ser considerados deben incluir el hashtag exacto: **#elecciones_ceetel**. Los comentarios emitidos en dicha publicación son únicos por usuario y el TRICEL los almacena en el archivo **pybook.txt**.

pybook.txt

```
grande ElNelson, para él mi voto #elecciones_ceetel
En #elecciones_ceetel voto por SophyPeace, por la peace en Telemática
Joaco, hasta la proximaaaaaa #elecciones_ceetel Dudu o SophyPeace que ganen
Vota Oscar3Names o ElNelson, votar por el dudú solo llevará al fracaso #elecciones_ceetel
vengo a probar su programa de elecciones #ELECCIONES_CEETEL
Dudu votó por si mismo jaja saludos #elecciones_ceetel
No se si votar por el Dudu o por SophyPeace, que dicen?
Vota por la SophyPeace poh
Nah, vota por el Dudu
Ya sé, votaré por JoacoHastaLaProxima #elecciones_ceetel
F los tres comentarios anteriores #elecciones_ceetel
#elecciones_ceetel Que gane la SophyPeace, por paridad de genero
Tienen razón cabros, SophyPeace o SophyPeace, si somos mamones #elecciones_ceetel
...
```

Cada línea representa el comentario de un estudiante en la publicación de la elección. Además, el archivo contiene muchos más datos que los que se muestran.

En esta peculiar elección, se le asignará un voto a una lista, si en **un mensaje válido** aparece el nombre **exacto** del candidato a presidente de la lista. Así, un comentario puede entregar varios votos a la vez, **pero solamente uno** a cada lista.

Considerando que pueden haber más comentarios y listas candidatas, **por lo tanto debe trabajar en genérico**, el CEETEL lo contrata como soporte tecnológico del TRICEL y le solicita:

- (a) (20pts) Desarrollar la función `initVotos(cand)`, la cual recibe como parámetro el string `cand`, que es el nombre de un archivo que contiene la información de los `candidatos` tal como `candidatos.txt`.

La función retorna un diccionario, que contiene como llave a cada uno de los candidatos, registrados por el TRICEL, y como valor el entero 0. Este diccionario funcionará como un contador de la votación. Maneje excepciones en caso que existan problemas con el nombre del archivo, `cand`.

```
>>>pizarra = initVotos("candidatos.txt")
>>>print(pizarra)
{'SophyPeace': 0, 'ElNelson': 0, 'JoacoHastaLaProxima': 0,
 'Dudu': 0, 'Oscar3Names': 0}

>>>pizarra = initVotos("candidatos2.txt")
El archivo candidatos2.txt no existe. Inténtelo denuevo.
```

- (b) (40pts) Implemente la función `cuentaVotos(pizarra, comentarios)`, la cual recibe dos parámetros: un diccionario `pizarra`, que contiene la información de los candidatos e inicializa los contadores de votos obtenidos, y el string `comentarios`, que representa el nombre del archivo que contiene la información de los `comentarios` efectuados por los estudiantes en la publicación de Pybook, como `pybook.txt`.

La función retorna un diccionario, como `pizarra`, pero contiene como valor la cantidad de votos obtenidos por los candidatos a partir de los comentarios **válidos** existentes en `comentarios`

Obligatoriamente, `pizarra` debe ser el diccionario generado por la función `initVotos(cand)`. No olvide manejar las excepciones correspondientes por el uso del archivo con nombre `comentarios`.

```
>>>pizarra = initVotos("candidatos.txt")
>>>pizarra = cuentaVotos(pizarra, "pybook.txt")
>>>print(pizarra)
{'SophyPeace': 4, 'ElNelson': 2, 'JoacoHastaLaProxima': 1,
'Dudu': 2, 'Oscar3Names': 1}

>>>pizarra = initVotos("candidatos.txt")
>>>pizarra = cuentaVotos(pizarra, "pystagram.txt")
El archivo pystagram.txt no existe. Inténtelo denuevo.
```

- (c) (40pts) Cree la función `resultadosVotos(pizarra)`, la cual recibe como parámetro un diccionario, `pizarra`, que contiene los candidatos y los votos que estos obtuvieron. La función debe generar un gráfico de torta que muestre los resultados de la elección. El título de la figura debe incluir la cantidad total de votos, el nombre de la lista ganadora y su presidente. Además, debe destacar la porción de la torta referente al ganador. Preocúpese de la estética de su gráfico. Tome el ejemplo como referencia. No olvide trabajar en genérico.

Obligatoriamente, `pizarra` debe ser el diccionario generado por la función `cuentaVotos(pizarra, comentarios)`.

```
>>>pizarra = initVotos("candidatos.txt")
>>>pizarra = cuentaVotos(pizarra, "pybook.txt")
>>>resultadosVotos(pizarra)
```

