

## Tarea 2 - Datos compuestos y Listas recursivas:

### "Estadísticas Plebiscito Constituyente"

Profesores: Bárbara Poblete y Alejandro Hevia. CC1002, Secciones: 5 y 7, Primavera 2020

Lea el enunciado completo antes de comenzar a trabajar

Un conocido suyo del Servicio Electoral le pide ayuda para realizar estadísticas de los resultados del Plebiscito constituyente del 2020, a lo cual Ud. acepta pues le interesa aportar a este proceso democrático. Se le pide implementar un programa que lea los resultados de las distintas mesas del plebiscito y entregue algunas estadísticas (cómputos) siguiendo un diálogo como se indica en los siguientes ejemplos de uso (en rojo lo que ingresa el usuario):

#### Ejemplo:

Bienvenido al sistema de estadísticas Plebiscito Constituyente 2020.

Ingrese los resultados por mesa

Circunscripción (o 'fin' para terminar): **Nunoa**

Número de la mesa: 1

Número de votos opción 'Apruebo': **200**

Número de votos opción 'Rechazo': **40**

Circunscripción (o 'fin' para terminar): **Santiago**

Número de la mesa: 5

Número de votos opción 'Apruebo': **500**

Número de votos opción 'Rechazo': **100**

Circunscripción (o 'fin' para terminar): **Nunoa**

Número de la mesa: 3

Número de votos opción 'Apruebo': **300**

Número de votos opción 'Rechazo': **100**

Circunscripción (o 'fin' para terminar): **Putre**

Número de la mesa: 3

Número de votos opción 'Apruebo': **150**

Número de votos opción 'Rechazo': **40**

Circunscripción (o 'fin' para terminar): **Nunoa**

Número de la mesa: 1

Número de votos opción 'Apruebo': **200**

Número de votos opción 'Rechazo': **40**

Circunscripción (o 'fin' para terminar): **fin**

Estadísticas Básicas:

Totales Plebiscito Constituyente

Total Opción Apruebo: 1150

Total Opción Rechazo: 280

Estadísticas Avanzadas:

Mesa con más concurrencia: Santiago, mesa 5, con 600 votantes.

Circunscripción de su interés: **Nunoa**

Votación en Circunscripción Nunoa:

Apruebo: 500, Rechazo: 140.

**Nota:** este es un ejemplo de ejecución. Su programa debe poder funcionar con otros valores (circunscripciones y números devotos) también.

Para implementar su programa, usará una base de datos implementada como lista recursiva, cuyos elementos serán estructuras del tipo resultadoMesa. La estructura resultadoMesa tiene la siguiente forma:

```
# resultadoMesa: circunscripcion (str), mesa (int), apruebo (int), rechazo (int)
estructura.crear("resultadoMesa", "circunscripcion mesa apruebo rechazo")
```

**Ejemplo:** si

```
r1 = resultadoMesa("Santiago", 3, 100, 20)
```

y

```
r2 = resultadoMesa("San Miguel", 2, 200, 60)
```

Entonces un ejemplo de lista recursiva de resultados es

```
L = crearLista(r1, crearLista(r2, listaVacía))
```

**Su tarea:** Para escribir su programa Ud. **debe implementar** las funciones indicadas más abajo dentro de un módulo llamado **estadisticas.py**, incluyendo recetas de diseño y test. Luego, deberá escribir el programa principal llamado **tarea2.py** que importe el modulo anterior y *usando esas funciones* implemente un programa para tener diálogos como el descrito anteriormente.

**Funciones que debe escribir para el módulo estadisticas.py:**

1. **buscaMesa:** esta función recibe una lista de resultados L, una circunscripción y un número de mesa, y retorna un dato del tipo resultadoMesa con el resultado correspondiente a la mesa de la circunscripción indicada. Si la mesa no está retorna None. Recuerde que None es un valor especial de Python. Nota: Suponga que el par (circunscripción, número de mesa) siempre representa un único resultado en la lista L (si es que existe).
2. **agregaMesa:** esta función recibe una lista de resultados L, un resultadoMesa r, y retorna una nueva lista con r agregado al principio de la lista. Esta función debe verificar que el resultado agregado no haya sido agregado anteriormente. Si no es así (esto es, el resultado a agregar ya existe en la lista), debe retornar la lista original.
3. **mesaConMasVotos:** esta función recibe una lista de resultados L y retorna un dato del tipo resultadoMesa que representa la mesa con mayor número de votos totales de la lista. El número de votos totales de una mesa es la suma de los votos de las dos opciones más blancos y nulos. Si la lista L es vacía, esta función retorna None.
4. **resultadosCircunscripcion:** Para tener sacar una estadística para una circunscripción, se le pide implementar una función que reciba una lista de resultados L, y un string con una circunscripción, y retorne la lista de resultados pero sólo asociados a la circunscripción indicada. Por ejemplo, si la lista L contiene resultados para 4 mesas y 2 de ellas son de "Paine", entonces resultadosCircunscripcion(L, "Paine") debiera retornar una lista de largo 2 con los dos resultados asociados a "Paine".
5. **totalesPorCircunscripción:** esta función recibe una lista de resultados L, una circunscripción, y un string con la palabra "Apruebo", "Rechazo", "Blancos", o "Nulos", y entrega el número de votos totales en la lista L por la opción indicada en el string pero, contando sólo los resultados de la circunscripción indicada.
6. **totalVotosFinales:** esta función recibe una lista de resultados L y un string con la palabra "Apruebo", "Rechazo", "Blancos", o "Nulos", y retorna el número de votos totales obtenidos por la opción indicada.

### Indicaciones importantes:

1. **Comience por implementar el módulo estadísticas.py con las funciones que se indican**, receta de diseño, y sus respectivos tests. No olvide además incluir una verificación de los parámetros dentro de la función misma (al menos el tipo de cada parámetro usando assert). Hágalas tal y cual como se pide en el enunciado. Si este módulo funciona de manera correcta Ud. ya podrá optar a una nota máxima de 5.0
2. **Luego implemente el programa principal tarea2.py sólo con las estadísticas básicas**. Si su módulo y el programa principal hasta acá funcionan de manera correcta, podrá optar a una nota máxima 6.0.
3. **Una vez que funcione lo anterior implemente las estadísticas avanzadas**. Si Ud. logra todos los objetivos podrá optar a una nota máxima de 7.0
4. **Bonus track (+0.5 pts):** Opcionalmente, se bonificarán las tareas que agreguen la siguiente funcionalidad adicional al programa principal: informar el número de circunscripciones distintas ingresadas. Esto debe mostrarse en la sección estadísticas avanzadas al final.

### Recuerde que:

- Para hacer esta tarea puede usar todos los módulos visto en clases hasta la semana 6 (inclusive).
- El trabajo es **individual**, preocúpese de leer las indicaciones del curso con respecto a *política de copias y formas permitidas de colaboración*. Se verificarán de forma automatizada que no exista plagio en las tareas.
- Debe enviar su tarea en la fecha de entrega indicada en u-cursos, puede hacer sus consultas de enunciado en las sesiones de consulta con los profes y en el foro. En la sesión de consultas con los profes se explicará el enunciado.

**Fecha de entrega:** Viernes 6 de Noviembre 2020, 23:59 hrs. en el módulo Tareas de u-cursos. Se aplicará un descuento de 1 punto por día hábil de retraso en entregar. Es decir, el descuento se aplicará para las tareas que se entreguen desde el día Lunes 9 de Noviembre en adelante.