

Ciclo 1 Fundamentos de Programación

Reto 4: Toma de Decisiones Cajeros Automáticos



El *Banco Principal* recientemente realizó la compra de una firma bancaria menor, integrando las sucursales, sistemas de información y cajeros automáticos de esta. Posterior a la integración, el banco necesita observar el uso y estado de los cajeros automáticos, los cuales se encuentran distribuidos en zonas, algunos están fuera de servicio, otros están funcionando y otros fueron recientemente cerrados. La gerencia necesita consolidar entonces, indicadores de sus cajeros, para luego contrastarlos con los indicadores de los cajeros provenientes de la firma integrada, con el fin de iniciar el proceso de toma de decisiones sobre los equipos, sin embargo, no registra información actualizada de sus máquinas, previo a la comparación con los equipos de la *Firma Integrada*. Se busca observar la capacidad instalada de los equipos que se encuentran operando o simplemente pendientes de mantenimiento del *Banco Principal*, contra los cajeros adquiridos de la *Firma Integrada*. Puntualmente, se requiere satisfacer la carencia de información de los equipos del *Banco Principal*, obteniendo la cantidad de retiros, consignaciones y transferencias que han sido realizadas en sus equipos, junto con el monto promedio asociado a estos movimientos.

El sistema de información encargado de monitorear los dispositivos, cuenta con una base de datos que registra el estado actual de los cajeros, y el historial de transacciones del último año (junio de 2020 a junio de 2021). La información es extraída de la base de datos y entregada en un diccionario con la estructura que se presenta en la Figura 1.

▼ 5E2U6FI:	
estado:	"Operando"
modeloCajero:	101
▼ transacciones:	
▼ 0:	
fechaMovimiento:	"27-08-2020"
monto:	1000000
tipoCuenta:	"cuentaVirtual"
tipoMovimiento:	"retiro"
▼ 1:	
fechaMovimiento:	"25-07-2020"
monto:	300000
tipoCuenta:	"ahorros"
tipoMovimiento:	"transferencia"
zona:	2
▼ 67vr2FI:	
estado:	"Cerrado"
modeloCajero:	101

Figura 1. Fragmento del diccionario entregado por la base de datos.

El diccionario que debe ser procesado, está constituido naturalmente por parejas *llave:valor*, donde la llave es el código del equipo, compuesta por 7 caracteres: 5 valores alfanuméricos y 2 que indican si la máquina es del banco principal (BP), o es un equipo de la firma bancaria secundaria que acaba de comprar el banco (FI). La Figura 1 presenta el cajero con el identificador (llave) **5E2U6FI**, y el valor asociado son todos los atributos del equipo. A continuación se especifican:

- **estado:** valor tipo string, indica si el equipo actualmente se encuentra 'Fuera de Servicio', 'Operando' o 'Cerrado'.
- **modeloCajero:** cualquiera de los modelos que la empresa de soporte ha instalado para el banco. Podría ser: 100, 101, 2017 o 2020 (valores enteros).
- **transacciones:** historial de transacciones del equipo en formato de lista, donde cada posición de dicha lista aloja un diccionario con los atributos o detalles de la transacción. Se observa en la Figura 1 los atributos del diccionario que representa cada transacción.
- **zona:** código de la zona donde se encuentra ubicado el equipo. Codificación de las zonas: 1, 2, 3, 4, 5, 6, o 7 (valores enteros).

Se requiere entonces una función que procese la información que contiene el diccionario proveniente de la base de datos, y retorne los siguientes consolidados para los cajeros del *Banco Principal*:

- 1) Cantidad de retiros realizados en los cajeros que están operando o a la espera de mantenimiento para volver a su funcionamiento normal (Operando y Fuera de Servicio).



- 2) Promedio de los montos presentados en los retiros del requerimiento. En lugar de un redondeo, se espera el valor resultante de realizar un recast a tipo entero del promedio que se obtenga, es decir, int().
- 3) Cantidad de consignaciones realizadas en los cajeros que están operando o a la espera de mantenimiento para volver a su funcionamiento normal (Operando y Fuera de Servicio).
- 4) Promedio de los montos presentados en las consignaciones del requerimiento. Como en el numeral 2, se espera un recast a entero del valor.
- 5) Cantidad de transferencias realizadas en los cajeros que están operando o a la espera de mantenimiento para volver a su funcionamiento normal (Operando y Fuera de Servicio).
- 6) Promedio de los montos presentados en las transferencias del requerimiento. Como en el numeral 2, se espera un recast a entero del valor.

Partiendo de lo anterior, a continuación se especifican los detalles técnicos del requerimiento. El esqueleto de la función, indicado en la Figura 2, recibe el diccionario con la información de los cajeros del caso de estudio, y retorna un diccionario de diccionarios, que describen la información solicitada de retiros, consignaciones y transferencias.

```
#Esqueleto de la función
def detalleTransaccionesBP(caso: dict) -> dict:
    """Desarrollar aquí el requerimiento. Se recomienda el uso de las funciones
    map, filter, reduce y opcionalmente zip para resolverlo con mayor facilidad."""
    pass
```

Figura 2. Prototipo de la función

En la siguiente tabla, se especifican los parámetros (campos, atributos o ítems) que llegan encapsulados en el diccionario que contiene el **caso**, única entrada de la función solicitada. Los nombres de dichos parámetros y sus contenidos no tienen tildes, previniendo problemas de codificación.

Especificación del diccionario de entrada

En la siguiente tabla, se realiza la especificación con un posible elemento del diccionario de entrada que tendría la llave **5E2U6FI**. Recordar que cada cajero estará identificado por una llave diferente, siguiendo la estructura descrita en la Figura 1.

Ítem	Tipo de Dato	Descripción
caso	dict	Diccionario cuyos ítems corresponden a los cajeros.



caso['5E2U6FI']['estado']	str	Cadena con el estado actual del equipo. Las cadenas que podrían encontrarse en este campo (dominio) son: 'Fuera de Servicio', 'Operando' o 'Cerrado'.
caso['5E2U6FI']['modeloCajero']	int	Entero que tomaría alguno de los siguientes valores: 100, 101, 2017, o 2020.
caso['5E2U6FI']['transacciones']	list	Listado de diccionarios que contienen el detalle de cada una de las transacciones.
caso['5E2U6FI']['transacciones'][i]['fechaMovimiento']	str	Cadena que contiene la fecha de la transacción en formato DD-MM-AAAA.
caso['5E2U6FI']['transacciones'][i]['monto']	int	Valor de la transacción. Los cajeros están programados para realizar movimientos únicamente por los siguientes valores: 20000, 50000, 100000, 200000, 300000, 500000 y 1000000.
caso['5E2U6FI']['transacciones'][i]['tipoCuenta']	str	Cadena con el tipo de cuenta asociado a la transacción. Los posibles valores son: 'ahorros', 'corriente' o 'cuentaVirtual'.
caso['5E2U6FI']['transacciones'][i]['tipoMovimiento']	str	Cadena con el tipo de movimiento



		asociado a la transacción. Los posibles valores son: 'retiro', 'consignacion' o 'transferencia'.
caso['5E2U6FT']['zona']	int	Entero que tomaría alguno de los siguientes valores: 1,2,3,4,5,6 o 7

El requerimiento especificado en los numerales (1-6), puede satisfacerse utilizando condicionales y ciclos convencionales, sin embargo, podría tener un grado de dificultad mayor. Se recomienda trabajar el requerimiento con las funciones *map*, *filter*, *zip* y *reduce*, ofrecidas por Python, provenientes del paradigma de programación funcional. Se aclara que su uso no es obligatorio, y el reto puede realizarse con dichas estructuras cíclicas y condicionales básicas. En la siguiente tabla se presenta el diccionario esperado después del cómputo que debe realizar la función a desarrollar.

Para visualizar de una forma más clara los casos de prueba, han sido generados también en versión JSON. Se procede descargándolos y abriéndolos localmente en un navegador como Firefox, para obtener una visualización como la que se presenta en la Figura 1. Estos archivos pueden ser descargados desde:

https://github.com/luismescobarf/clasesCiclo1/tree/master/P61/archivosReto4/Formato_JSON

Estructura de la Salida

Tipo de Retorno	Descripción	Ejemplo
dict	Diccionario de diccionarios que describen la información solicitada de retiros, consignaciones y transferencias. Notar que todos los valores son tipo entero, las cantidades naturalmente, y los promedios al realizar un recast int() al cálculo realizado.	{'retiros': {'cantidad': 12, 'promedio': 555833}, 'consignaciones': {'cantidad': 15, 'promedio': 263333}, 'transferencias': {'cantidad': 9, 'promedio': 86666}}

Dado que algunos de los casos de prueba son extensos, han sido definidos y subidos al repositorio del grupo en GitHub. Las definiciones se encuentran en:

<https://github.com/luismescobarf/clasesCiclo1/blob/master/P61/archivosReto4/casosGeneradosP61.py>

Para utilizar los casos de prueba, basta tener descargado en la misma ubicación del archivo .py donde se desarrolle la función del reto, las definiciones que se encuentran en casosGeneradosP61.py (archivo que se encuentra en el enlace anterior). Luego, en la solución que se esté desarrollando, incorporar la siguiente línea antes de los llamados:



```
from casosGeneradosP61 import *
```

Partiendo de lo anterior, los casos de prueba son los siguientes:

Caso de Prueba 1:

Llamado:

```
print(detalleTransaccionesBP(caso1))
```

Resultado esperado:

```
{'retiros': {'cantidad': 12, 'promedio': 555833}, 'consignaciones': {'cantidad': 15, 'promedio': 263333}, 'transferencias': {'cantidad': 9, 'promedio': 86666}}
```

Caso de Prueba 2:

Llamado:

```
print(detalleTransaccionesBP(caso2))
```

Resultado esperado:

```
{'retiros': {'cantidad': 4, 'promedio': 230000}, 'consignaciones': {'cantidad': 5, 'promedio': 410000}, 'transferencias': {'cantidad': 4, 'promedio': 292500}}
```

Caso de Prueba 3:

Llamado:

```
print(detalleTransaccionesBP(caso3))
```

Resultado esperado:

```
{'retiros': {'cantidad': 2, 'promedio': 35000}, 'consignaciones': {'cantidad': 3, 'promedio': 416666}, 'transferencias': {'cantidad': 3, 'promedio': 180000}}
```

Caso de Prueba 4:

Llamado:

```
print(detalleTransaccionesBP(caso4))
```



Resultado esperado:

```
{'retiros': {'cantidad': 11, 'promedio': 451818}, 'consignaciones': {'cantidad': 9, 'promedio': 197777}, 'transferencias': {'cantidad': 8, 'promedio': 281250}}
```

Recordar: En la plataforma debe subirse una función con el mismo nombre, la misma cantidad de argumentos, y el retorno debe tener exactamente la estructura de la salida que se presenta en este documento.