

Una reconocida empresa de servicio de telefonía celular nos está pidiendo realizar el sistema que maneja las líneas y packs de sus clientes. No vamos a revelar cuál es, pero hay una pista en la imagen.



Una línea tiene un número de teléfono, y puede tener varios packs activos, que sirven para que la línea pueda realizar consumos. Por ahora Pdepfoni permite hacer dos tipos de consumo:

- Consumo de Internet (cada consumo es por una cantidad de MB)
- Consumo de Llamadas (cada consumo es por una cantidad de segundos)

Para poder realizar esos consumos, se pueden agregar packs a la línea. Algunos packs que hoy existen son:

- Cierta cantidad de crédito disponible.
- Una cant de MB libres para navegar por Internet.
- Llamadas gratis.
- Internet ilimitado los fines (*)

Agregar packs a futuro debe ser sencillo (como Internet gratis los martes, segundos de llamada libres, etc., aunque no hace falta agregarlos ahora).

Además algunos packs pueden tener vigencia (una fecha de vencimiento). Puede entonces existir un pack de llamadas gratis ó uno con 1000 MB que si están vencidos, no podrán utilizarse más.

(*) Ver al final del enunciado el uso de la clase Date.

Se pide la codificación de la solución en WolloK para:

1. Conocer el costo de un consumo realizado.

Se sabe que la empresa de telefonía dispone de precios por MB y por segundo de llamada. Además, se sabe que siempre se cobra un precio fijo por los primeros 30 segundos de llamada y luego se cobra por segundo pasado de los 30.

Por ejemplo, si el precio por segundo es \$0.05, con un precio fijo de \$1; y el precio por MB es \$0.10, entonces

*- un consumo de llamada de 40 segundos vale ($\$1 + 10 * \$0.05 = \$1.50$),*

*- y un consumo de internet de 5 MB vale ($5 * \$0.10 = \0.50).*

Recomendación: leer el punto siguiente para ayudarse con el modelado.

2. Sacar información de los consumos realizados por una línea:

- a. Conocer el costo promedio de todos los consumos realizados dentro de un rango de fechas inicial y final.
- b. Conocer el costo total que gastó la línea entre todos los consumos de los últimos 30 días.

3. Saber si un pack puede satisfacer un consumo.

Por ejemplo, un pack de 100MB libres satisface un consumo de internet de 5 MB, pero una llamada de 60 segundos no puede ser satisfecha por ese pack. Por otro lado, un pack de \$100 de crédito satisface ambos consumos (con los precios del ejemplo del punto 1).

4. Poder agregar algún pack de los mencionados a una línea.

5. Saber si una línea puede hacer cierto consumo.

Por ejemplo, la línea 1566666666 tiene los siguientes packs: 10 MB libres, \$50 de crédito, 200MB libres vencidos el 13/10/2019; 5 MB libres, y uno de 15MB libres vigentes hasta el 12/12/2019.

Esta línea **no** puede hacer un consumo de Internet de 20MB porque el consumo debe poderse satisfacer completamente por un pack.

6. Realizar un **consumo** en la línea. Esto debería, además de registrar el consumo en la línea (punto 2), producir el gasto del pack que lo puede satisfacer: Por ahora los packs de MB libres y crédito se van gastando, el resto no. Cuando se hace un consumo, se consume el último pack agregado que puede satisfacerlo. (Sí, si primero agregaste un pack con llamadas gratis y después otro con crédito, primero se te descuenta el crédito. Por cualquier queja nuestro [buzón de quejas](#)).

Si no se puede realizar el consumo, porque no me alcanzan los packs, debe lanzarse un error.

7. Más sobre packs:

- Realizar una limpieza de packs, que elimine los packs vencidos o completamente acabados.
- Agregar los packs "MB libres++", que son como los packs de MB libres pero que cuando se gastan los MB libres igual te siguen sirviendo... Pero sólo para consumos de 0.1 MB o menos.

8. Agregar las líneas **black** y las líneas **platinum**. Las líneas black, cuando ningún pack deja realizar el consumo, permiten realizar de todas formas el consumo, simplemente suman a un registro de deuda el costo del consumo. Las líneas platinum permiten realizar cualquier consumo y no suma ninguna deuda. Las líneas deberían poder cambiar entre común, black y platinum sin perder registro de los packs ni las estadísticas de consumo.

9. En base a los ejemplos, hacer 3 tests:

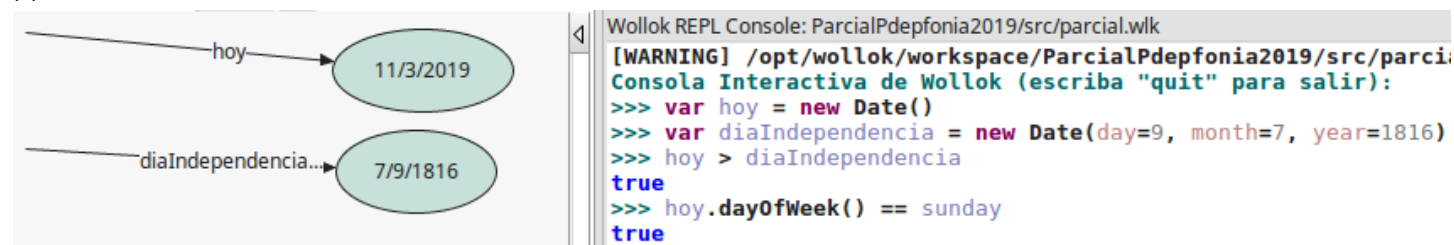
- uno que verifique que se haya producido efecto correctamente,
- uno que verifique que no se pueda hacer algo,
- y uno que verifique que se devuelva bien algo.

10. Describir y **justificar** qué es lo mínimo necesario que hay que hacer para agregar:

- una nuevo pack (como los mencionados al ppio)
- un nuevo consumo (como la posibilidad de enviar sms)

Debe usarse un diagrama que acompañe la explicación.

(*) Uso de la clase Date:



Las fechas también entienden mensajes que permiten restar días y saber si una fecha está entre otras dos:

```

>>> var ayer = hoy.minusDays(1)
>>> var maniana = hoy.plusDays(1)
>>> hoy.between(ayer,maniana)
true
  
```