



3ra fecha 2024

Segundo recuperatorio - Orientación a Objetos I - 21/12/24

Una plataforma de streaming registra las actividades realizadas por los clientes para calcular los montos a cobrar a cada uno. De cada **cliente** se conoce el nombre y su fecha de alta. Los clientes pueden contratar un plan que puede ser individual o grupal. Al momento de contratar el **plan**, se establece una cantidad límite en las direcciones IP desde las cuales se permite acceder a las actividades, sin generar un cargo adicional. En el caso de los planes **individuales** se permite acceder desde una **única dirección IP**; mientras que en los planes **grupales** se permite acceder desde varias direcciones IP, con una cantidad máxima establecida al contratar el plan. En los planes **individuales** se registra además la cantidad de minutos contratados. Los clientes pueden cambiar el plan contratado (y no es necesario mantener el registro de planes anteriores). Un cliente tendrá siempre un solo plan. Hay dos tipos de actividades: reproducciones de video y sesiones de juegos.

- En las **reproducciones de video** se conoce la **fecha de inicio de la actividad y la dirección IP** desde la cual se accedió (almacenado como un String). Además, una reproducción de video conoce la duración total y la duración en publicidad (ambas expresadas en minutos). Se puede obtener la duración real utilizada calculada como la duración total menos la duración en publicidad.
- En el caso de las **sesiones de juegos** se conoce la **fecha de inicio de la actividad**, la duración (expresada en minutos) y la **dirección IP** desde la cual se accedió. Además, el usuario tiene la opción de utilizar diversos ítems que mejoran la experiencia o brindan ventajas dentro del juego. Cada uno de estos **ítems** tiene nombre, cantidad y precio unitario. El usuario deberá pagar, si corresponde, el precio final de cada ítem como cargo adicional (calculado como la cantidad multiplicada por precio unitario).

La empresa cobra un precio base según el plan contratado, al que le suma el monto correspondiente a las actividades registradas y un monto extra por penalización (si el cliente excede la cantidad de IPs según su plan).

Nos piden implementar la siguiente funcionalidad:

Calcular el precio base de un plan: Para los planes individuales, el precio base es \$20 por la cantidad de minutos contratados. Para los planes grupales, el precio base es de \$800 por la cantidad límite de **IPs contratadas**.

Monto total a cobrar: dado un cliente y un intervalo de tiempo (fecha inicial - fecha final), retorna el monto total a cobrar para ese intervalo. El monto total a cobrar es la suma del **precio base según el plan**, el **monto por las actividades registradas** y el **monto por penalización**.

Monto a cobrar por las actividades registradas: dado un cliente y un intervalo de tiempo (fecha inicial - fecha final), retorna el monto a cobrar en ese intervalo, según cada caso:

- **Video:** Se cobra \$10 por cada minuto de duración real de la reproducción del video.
- **Juegos:** Si la duración de la actividad supera 360 minutos, se cobra el precio final de todos los ítems utilizados durante la sesión. En caso contrario, no se cobra nada.

Monto por penalización por superar el límite contratado: dado un cliente y un intervalo de tiempo (fecha inicial - fecha final), retorna el monto por penalización en ese intervalo. El monto de penalización varía según el plan:

- Si el cliente tiene un plan individual será penalizado si utilizó más de una dirección IP en las actividades registradas durante ese intervalo. El costo por cada IP adicional que haya utilizado es de \$300.
- Si el cliente tiene un plan grupal será penalizado si el número de direcciones IP utilizadas en el intervalo supera la cantidad máxima establecida al contratar el plan. El costo por cada IP adicional es de \$500. En caso de que la antigüedad del cliente sea mayor a 10 años, no se cobra esta penalización.

Ayuda: Para conocer los años entre dos fechas: `Period.between(fechaInicio, fechaFin).getYears()`

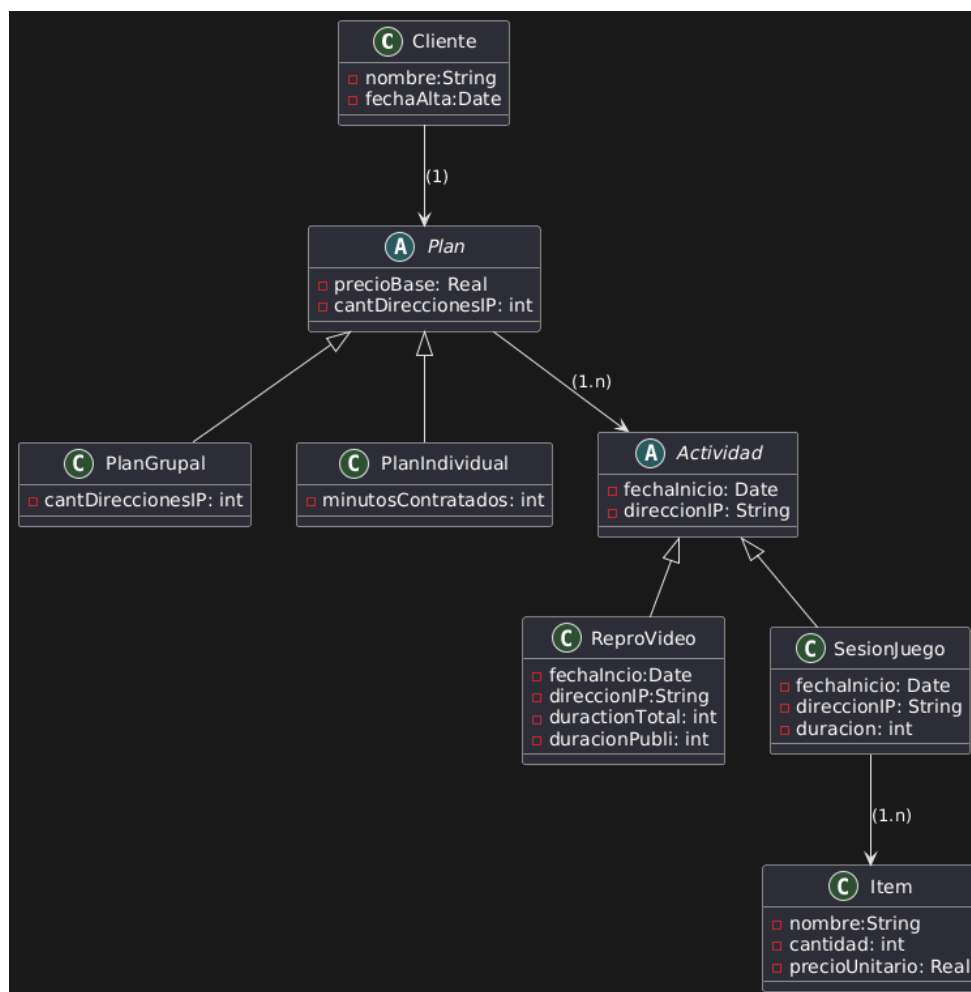
Tareas (debe realizar los cuatro ítems para aprobar):

1. Describa el diseño de su solución utilizando un diagrama de clases UML.
2. Implemente en Java toda la funcionalidad antes descripta. Puede usar las clases implementadas en los ejercicios de la práctica: `DateLapse`, `FilteredSet`, `Bag`, según se requiera.
3. Diseño de los casos de prueba: Enfocándose en la funcionalidad que permite calcular el monto por penalización por superar el límite contratado (todos los métodos de todas las clases involucradas en conseguir esa funcionalidad), determine y enumere **qué métodos** testear, **indicando clases** y casos de prueba (teniendo en cuenta los conceptos de valores de borde y particiones equivalentes). Identifique claramente las particiones que encuentra y los valores de borde para cada caso.
4. Un **ejemplo del código Java** necesario para crear un cliente con un plan grupal con una cantidad máxima de 6 direcciones IP, que accedió a dos actividades: una reproducción de video, y una sesión de juegos, donde se utilizó un ítem de nombre "daño aumentado", con 2 unidades y precio unitario de 1000. Luego el cliente cambia a un plan individual. Complete con los valores que necesite según corresponda.

1) Conceptos:

Actividades de clientes, clientes, calcular monto a cobrar, nombreCliente, fechaDeAlta, plan cantidadLimiteDirecciones, plan individual, plan grupal, reproducciones de video, sesiones de juego, fechaInicioActividad, direccionIp, duracionTotal, duracionPublicidad, items, nombreltem, cantidadItem, precioItem, calcularCargoAdicional, precioBase, montoExtraPorPenalizacion, calcularPrecioBase, montoTotalACobrar, montoACobrarPorActividades, montoPorPenalizacion

Modelo de dominio:



Modelo UML:

▼ PLANTUML

```

@startuml
class Cliente{
    -nombre:String
    -fechaAlta:Date
    +calcularBase(): Real
    +montoTotalACobrar(periodo Date): Real
}

abstract class Plan{
    -precioBase: Real
    -cantDireccionesIP: int
    +getCantIp(): int
    +<<abstract>> calcularPrecioBase():Real
    +calcularMontoActividades(periodo Date): Real
    +contarIpsEnPeriodo(periodo Date): int
    +<<abstract>> calcularPenalizacion(periodo Date): Real
}

class PlanGrupal{
    -cantDireccionesIP: int
    +calcularPrecioBase():Real
    +calcularPenalizacion(periodo Date): Real
}

class PlanIndividual{
    -minutosContratados: int
    +calcularPrecioBase():Real
    +calcularPenalizacion(periodo Date)
}

abstract class Actividad{
    -fechaInicio: Date
    -direccionIP: String
    +getFecha(): Date
    +getIP():String

```

```
+<<abstract>> calcularCosto():Real  
}
```

```
class ReproVideo{  
    -durationTotal: int  
    -duracionPubli: int  
    +calcularCosto():Real  
}
```

```
class SesionJuego{  
    -duracion: int  
    +calcularCosto():Real  
}
```

```
class Item{  
    -nombre:String  
    -cantidad: int  
    -precioUnitario: Real  
    +precioFinal():Reeal  
}
```

```
Cliente → Plan: (1)  
Plan <|-- PlanGrupal  
Plan <|-- PlanIndividual  
Plan → Actividad: --actividades  
Actividad <|-- ReproVideo  
Actividad <|-- SesionJuego  
SesionJuego → Item: --items  
@enduml
```

