

TRABAJO PRÁCTICO FINAL

UCA, Programación Orientada a Objetos II, segundo cuatrimestre de 2019.

INTRODUCCIÓN

Este trabajo, que comienza durante el último mes de cursada y finaliza con una defensa en las instancias de examen final, define la aprobación de la materia y tiene como objetivo poner en práctica y ejercitar todos los conceptos vistos de manera integradora.

PRIMERA PARTE

Se deberá implementar en Javascript, de manera iterativa e incremental realizando Test-Driven Development lo siguiente:

Queremos realizar el diseño con objetos relacionado a la venta de paquetes de una compañía telefónica. Lo que se necesita es poder vender paquetes a clientes que estén registrados como clientes de la compañía, y llevar registro de la validez de los mismos, como así también de los consumos que los clientes vayan realizando.

Un paquete contiene: una cantidad de datos móviles (expresada en GB), una cantidad de tiempo para realizar llamadas (expresada en minutos), una duración en días (luego de la misma, se vence), y un costo (expresado en pesos). Ejemplos de paquetes serían los siguientes:

- 2.5 GB y 1000 minutos durante 30 días, por \$400 pesos
- 1GB y 100 minutos durante 7 días, por \$150 pesos

Del cliente conocemos su nombre completo y número de línea, con esto basta para identificarlo. Además tiene asociada una cuenta, de la cual se va a debitar el monto de los paquetes que vaya comprando. Este es un sistema prepago, los clientes deben cargar dinero a su cuenta antes de comprar un paquete. Si el cliente no tiene suficiente saldo para comprar un paquete, el sistema debe lanzar un error.

El cliente va a poder adquirir todos los paquetes que estén disponibles. No va a poder adquirir dos paquetes al mismo tiempo, debe esperar que se termine uno para adquirir otro. No obstante, puede configurar para que sus paquetes se renueven automáticamente, para tener cobertura en todo momento.

Vamos a querer registrar cada consumo realizado por el cliente. Podemos asumir que un sistema externo nos va a proveer de la siguiente información para los diferentes tipos de consumo:

- Consumo de internet: cantidad de MB, fecha/hora de inicio y fin
- Consumo de minutos: cantidad de minutos, fecha/hora de inicio y fin.

Cada vez que se efectúa un consumo, se “descuenta” del paquete actual que el cliente tenga, queremos saber en todo momento cuánto queda disponible (de datos y minutos de llamadas). No

se puede efectuar un consumo mayor a lo que el paquete tenga disponible, se debe lanzar un error en ese caso. Queremos saber el historial de consumos ordenado por fecha, y con un filtro opcional por fecha y hora de inicio y fin.

El paquete se considera agotado cuando no quedan datos ni minutos de llamadas por consumir. Y se considera vencido cuando la fecha de expiración (determinada por la fecha de compra del paquete + la duración del mismo) es mayor a la fecha de hoy. Un cliente puede comprar un paquete nuevo si el último paquete está vencido o agotado.

SEGUNDA PARTE

Cada uno tendrá una modificación/extensión a realizar en la implementación de la primera parte. Dicho cambio será uno asignado aleatoriamente de la siguiente lista:

1. Poder configurar los paquetes de manera que los datos o los minutos para llamar sean ilimitados. De esta manera podemos tener un paquete como: 2.5GB + llamadas ilimitadas, entonces no importa la cantidad de consumos de llamadas que tenga el cliente, no hay chances de que se agote.
2. Poder agregar a los paquetes la posibilidad uso ilimitado de una app en particular, por ejemplo WhatsApp. Esto supone también agregar un nuevo tipo de consumo, en el cual sabemos cuál fue la aplicación.
3. Identificar a los clientes más "gastadores", que son aquellos que tienen por lo menos dos o más combos que fueron consumidos antes de tiempo. Ordenarlos del más gastador al menos gastador: contar la cantidad promedio de días que consumieron su paquete antes de que se venza. Por ejemplo, si un cliente consumió 3 paquetes, 2 de los cuales los terminó 4 y 8 días antes de que se venzan, y el otro no lo consumió antes de su vencimiento, su cantidad promedio de días es 6.
4. Implementar una promoción automática del 25% de descuento en la compra de nuevos paquetes a todos los clientes que hayan adquirido más de 5 paquetes en su vida.
5. Agregar promociones de consumo al sistema. Una promoción de consumo consiste en reducir a la mitad a todos los consumos realizados durante una hora estipulada (por ejemplo, si se consumieron 100MB, se cobra un consumo de 50MB). Fuera de esa hora, los consumos valen lo normal. Al ver el detalle de consumos de un cliente, queremos saber cuál fue el consumo original vs. el consumo con descuento.
6. Hacer que los clientes se puedan prestar datos y/o minutos. Esto se tiene que registrar en el detalle de consumos de cada cliente (del que recibe y del que lo otorga).

PAUTAS

METODOLOGÍA

Se deberá seguir la metodología de Test-Driven Development para asegurar la calidad del trabajo. En caso de no saber qué decisión tomar ante alguna parte del enunciado, consultar con el docente para definir un acuerdo y documentarlo (ver próxima sección).

FORMATO DE LA ENTREGA

Se deberá enviar un archivo .zip, usando la convención de nombres

TrabajoFinal-NombreApellido, conteniendo lo siguiente:

- El código del proyecto Node, completo, incluyendo tests.
- Un breve documento (no más de 4 páginas, en PDF preferentemente) que explique:
 - Los conceptos aplicados (de aquellos que vimos, pueden ser de los más básicos a los más avanzados), y en qué partes específicas del código se ven reflejados.
 - Explicar la evolución del trabajo: por dónde comenzaron, qué dificultades tuvieron, si hubo cambios en la definición del alcance, diferentes interpretaciones del enunciado, en conclusión cualquier detalle que sea relevante para explicar mejor el resultado final.

Durante la instancia de examen final, se deberá realizar una defensa del ejercicio mostrando la resolución y respondiendo preguntas sobre diseño de parte del docente.