



Ampliación

Ahora nos han solicitado ampliar la funcionalidad de la aplicación de tal manera que se haga un programa de fidelización y bonificación de clientes.

Los clientes podrán recomendar nuestro taller a otros clientes (amigos, familiares, etc.) y recibirán por ello unos bonos que podrán usar a la hora de pagar sus facturas. Un cliente únicamente podrá ser recomendado (a efectos de bonos) por un único cliente.

De igual modo y con el fin de premiar la fidelidad de los clientes se darán bonos según el número de reparaciones que haya tenido y/o por ciertos importes gastados en el Taller.

Los bonos pueden ser canjeados a la hora de pagar facturas en el Taller.

Actores del sistema y casos de uso

Administrador	<ul style="list-style-type: none">- Generación automática de bonos- Listados de importe de bonos sin consumir (por cliente, totales, etc.)
Caja	<ul style="list-style-type: none">- Gestión de Medios de Pago (CRUD)- Liquidar Factura
Jefe Taller	<ul style="list-style-type: none">- Gestión de Clientes (ampliación)<ul style="list-style-type: none">o CRUD de clienteso Cliente viene recomendado por....o Creación automática de medio de pago metálico al dar de alta un cliente- Listar clientes recomendados por un cliente

Descripción de los casos de uso

Caja. Gestión de medios de pago

El usuario que atiende la caja necesita poder gestionar los medios de pago de cada cliente en el Taller. Para ello, podrá dar de alta, baja y consultar los medios de pago del cliente. No se ha detectado la necesidad de modificar un medio de pago; si ha sido creado de forma errónea se elimina y se vuelve a crear. No se podrá eliminar un medio si ya se le han hecho cargos. Además, los medios de pago de tipo metálico no se eliminan.

Los datos a registrar dependen del tipo de medio de pago: metálico, tarjeta y bono.

- Metálico: todos los clientes tienen este medio de pago. Solo pueden tener un medio de pago de tipo “en metálico”, que será creado de forma automática por el sistema al dar de alta un cliente. Por lo tanto, el usuario no creará, modificará, ni eliminará ningún medio de pago de este tipo.
- Tarjeta: los clientes pueden tener varios medios de pago de este tipo. Se creará uno por cada tarjeta que el cliente facilite. Se deben pedir los datos de la tarjeta: tipo (Visa, MasterCard, etc.), número y fecha de caducidad. Al dar de alta es necesario verificar que



el número de la tarjeta no esté repetido en el sistema y que la fecha de caducidad no haya pasado.

- Pago mediante bonos: Los clientes pueden tener cualquier número de ellos. Cada bono se identifica de forma única mediante un código. El código es generado de forma automática por el sistema. Los bonos tienen además una descripción (motivo por el que se regala el bono) y la cantidad de dinero disponible, que es el valor actual del bono.

Todos los medios de pago tienen un valor *acumulado* que indica la cantidad de € que se ha pagado hasta ahora con ese medio de pago (el mismo medio se puede haber empleado para pagar diferentes facturas). En el caso de los bonos, el acumulado coincide con lo que se lleva gastado de éste y el campo *disponible* lo que todavía queda.

Administrador. Generación automática de bonos

Cada cierto tiempo el administrador lanzará el proceso de generación automática de bonos que seguirá las siguientes reglas:

- Por cada cliente que haya **recomendado a otros 3 clientes** (que hayan realizado alguna reparación en el Taller) se le emitirá un bono por valor de 25€, sin límite en el número de bonos que puede conseguir. Es necesario que el cliente al que se le vaya a emitir un bono haya hecho previamente alguna reparación en el Taller. Los bonos de este tipo llevan la descripción “Por recomendación”.
- Por cada **3 averías** reparadas en el Taller (y cobradas) se le emitirá un bono de 20€. Sin límite en el número de bonos que puede conseguir. La descripción de este tipo de bono será “Por tres averías”.
- Por cada **factura superior a 500€** (y ya cobrada) se le emitirá un bono de 30€. Sin límite en el número de bonos que puede conseguir. La descripción será “Por factura superior a 500€”.

Este proceso se ejecutaría automáticamente una vez por día por una tarea programada del sistema; pero en nuestro caso, y para facilitar la labor, se lanzará de forma manual desde el menú del administrador.

Administrador. Listados de Bonos

El administrador necesitará tener una serie de listados que le permitan conocer la situación de los bonos. Para ello tendrá:

- Listado de los bonos de un cliente dado. Se pedirá el identificador del cliente y se mostrará una lista con todos y cada uno de los bonos. Se mostrará de cada uno su código, descripción, disponible y acumulado. Además, al final del listado aparecerá una línea con información agregada de todos los bonos del cliente que indique: cuantos son, el importe total, importe ya consumido e importe por consumir.
- Listado de información agregada de los bonos de cada cliente con la siguiente información: dni, nombre cliente, nº total de bonos emitidos, importe total de los bonos emitidos, importe total canjeado de bonos, importe total que queda disponible.

Caja. Liquidar factura

Antes de recoger el vehículo el cliente se dirigirá a la caja a pagar. La persona al cargo buscará por número de factura la que corresponde al cliente. El sistema le mostrará los todos medios de pago



disponibles para ese cliente; podrán ser metálico, tarjetas de crédito o bonos. Puede fraccionarse el pago de una factura entre todos los medios disponibles para ese cliente. Por ejemplo, para una factura de 500€ se podrían utilizar 2 bonos de 100€ y el resto pagarlo con una tarjeta de crédito.

Se deben seguir las siguientes reglas:

- Todo cliente tendrá siempre disponible el pago en metálico.
- Si el cliente tiene tarjeta(s) registrada(s) en el sistema podrá emplearla si no está caducada.
- El sistema llevará cuenta, para cada medio de pago, del total pagado (acumulado) entre todas las facturas pagadas.
- Si el cliente tiene bono(s) podrá usarlo(s) sin pasarse de la cantidad máxima disponible. Al hacer el pago el sistema actualizará la cantidad disponible en el bono. Por ejemplo, si el cliente quiere pagar una factura de 50€ y tiene un bono con 90€ disponibles, el sistema aceptará el pago, incrementará el acumulado del bono en 50€ y dejará el disponible en 40€. Se pueden utilizar tantos bonos como se desee para pagar una factura.
- Una factura no puede ser pagada parcialmente. Es decir, no puede pagarse una parte la factura y dejar pendiente de pago otra cantidad. O se paga por completo o no se paga nada.
- Una vez la factura está pagada pasará al estado *ABONADA*.

Jefe de Taller. Gestión de clientes

El jefe de taller registrará un cliente la primera vez que venga al taller. De cada cliente se registrará dni, nombre, apellidos, dirección postal, teléfono y correo electrónico. Sobre cada cliente registrado se podrán aplicar las operaciones CRUD básicas: alta, modificación, baja, ver detalle y listado de todos. Para la modificación solo se cambiarán los datos básicos del cliente. Un cliente solo podrá ser eliminado si no tiene vehículos registrados y, en ese caso, se eliminará su metálico y la recomendación que haya podido recibir.

Para dar soporte a la nueva funcionalidad es necesario ampliar la gestión de clientes original especificada en el problema de referencia de la siguiente forma:

- Al dar de alta un cliente, el sistema, de forma automática, le creará y asociará un medio de pago de tipo “metálico”. Éste se eliminará en caso de que se borre el cliente.
- También en el proceso de dar de alta, se le preguntará si viene “recomendado” por algún otro cliente ya existente en el Taller y caso de ser así se registrará esa información. Un cliente nuevo solo puede tener una recomendación, y el cliente “recomendador” ya debe estar registrado en el sistema y tener al menos una factura pagada. En caso de que se borre el nuevo cliente (antes de que se le registren vehículos) se deberá borrar también la recomendación que recibió.
- Listar los clientes que han sido recomendados por un cliente especificado.

Criterios de Diseño

A continuación se mencionan una serie de criterios de diseño **obligatorios**. El no utilizarlos tendrá una penalización en la calificación.

- El diseño debe ajustarse a la **estructura de paquetes** vista en clase.
- Todo el código entregado pasará por un **detector de plagio**.



- Es necesario realizar correctamente el **tratamiento de errores** (excepciones) y hacerlo en el lugar correcto (no se puede interactuar con el usuario en ningún otro sitio que la capa de presentación). No es válido que la aplicación termine de forma abrupta con una traza de excepción en pantalla.
- Todas las operaciones deben mantener la **integridad referencial** de los datos. Cuidado con los borrados o modificaciones (ej: ¿qué pasa si intento borrar un bono que ha sido utilizado para pagar una factura? no debe permitirlo).
- El código debe ajustarse a las **Java Code Conventions**¹. No debe tener warnings (serán tratados por nuestra parte como errores de compilación).
- El **código** debe estar “limpio”, bien sangrado, legible y comentado correctamente (javadoc donde proceda). No debe haber código “muerto” ni código para depuración (println...).
- El sistema tiene que funcionar **al menos con el motor HSQLDB**.
- Todas las sentencias SQL deben estar **externalizadas**.
- Se debe entregar (a través del campus virtual) un único fichero comprimido con el siguiente contenido:
 - * Un fichero de texto cuyo nombre será tu código UO (*UOnnnnn.txt*) con tu **información personal** (Nombre, DNI), qué **casos de uso** te ha tocado implementar y comentarios que quieras hacer.
 - * **Proyecto Eclipse** completo (sin el directorio *bin* para aligerar)
 - * **Diagrama UML de clases** representando el nuevo modelo de dominio de la ampliación. Debe ser un diagrama de análisis (esto es, conceptual) y no debe incluir detalles técnicos propios de diseño o implementación. **No debe ser hecho con ingeniería inversa**. Y tampoco se admiten esquemas dibujados a mano y luego escaneados o fotografiados.
 - * **Diagrama de tablas** (modelo relacional) como en el enunciado original (no es un diagrama entidad-relación) con las nuevas tablas añadidas y aquellas que se hayan modificado o se relacionen con las nuevas. Este diagrama *debe* ser hecho con **herramientas de ingeniería inversa**.
 - * La **base de datos** debe estar rellena con datos suficientes para poder pasar pruebas (esto incluye las nuevas tablas creadas). **Si no se entregan datos para poder realizar las pruebas la entrega será nula**.
 - * No se debe modificar la biblioteca AlbUtil. Si consideras que algún método de esta librería se puede mejorar, o echas de menos alguno, hazlo en un nuevo paquete dentro del proyecto que se entrega. Si te parece que merece la pena añadirlo a la librería original puedes comentarlo en el fichero *UOnnnnn.txt*.

¿Qué casos de uso se deben implementar?

- Si tu UO módulo 3 es 0 (uo % 3 == 0):
 - * Administrador. Generación automática de bonos: por 3 recomendaciones.
 - * Caja. Gestión de Medios de Pago
 - * Administrador. Listado de Bonos

¹ <http://www.oracle.com/technetwork/java/codeconventions-150003.pdf>



- Si tu UO módulo 3 es 1 ($uo \% 3 == 1$):
 - * Administrador. Generación automática de bonos: por 3 averías.
 - * Jefe Taller. Gestión de Clientes.
- Si tu UO módulo 3 es 2 ($uo \% 3 == 2$):
 - * Administrador. Generación automática de bonos: por 3 averías y por facturas > 500€.
 - * Administrador. Listado de Bonos
 - * Caja. Liquidación de facturas.
- Independientemente de los casos de uso que te toquen es necesario que el modelo de dominio se implemente completo y correcto para todos ellos (como si tuvieras que implementarlos todos).

Recuerda

Además del profesor en las horas presenciales y las de tutorías, tienes a tu disposición, en el campus virtual, el foro para consultas sobre cualquier duda de la asignatura, del enunciado, posibles soluciones, etc. Usalo tanto como necesites.

El foro es una herramienta muy útil para que los profesores nos podamos dar cuenta de posibles problemas que os surjan y así poder ayudarlos. Os animamos a utilizarlo. Recordad que no hay preguntas tontas.

Esta práctica, como muchas, te va a llevar un tiempo. Los profesores han estimado, y contrastado después de varios cursos, que el tiempo a emplear está ajustado al programa y a la guía docente. Pero ten en cuenta que esas estimaciones se hacen contando con que la vas haciendo al ritmo que te indica el profesor, en paralelo con las clases presenciales de teoría y prácticas, y que le dedicas cada semana el tiempo de trabajo autónomo que se espera (1,5 horas por cada hora presencial).

Dejar todo el trabajo para la última semana, invalida por completo esa planificación y la metodología docente que los profesores están tratando de aplicar. El resultado probable, si ese es el caso, es que pierdas mucho tiempo tratando recuperar el hilo de las cosas, te encuentres con infinidad de problemas menores que no sabes ubicar (de los que casi seguro te advirtió el profesor en clase, pero no te acuerdas ya...), te hagas un lío leyendo por foros, la ansiedad por entregar te lleve a copiar fragmentos de código que encuentras por ahí, pero que no entiendes del todo, y que, al final, se te atragante y no llegues a captar la esencia conceptual del tema de la práctica.

Luego viene el examen de teoría, que suele consistir en preguntas cortas sobre conceptos con los que te tuviste que enfrentar en las prácticas, y, aunque hayas conseguido entregarlas, de nada te servirá el atragantón si al final no has entendido los conceptos básicos. Recuerda que debes sacar una nota mínima en el examen de teoría para que se aplique el cálculo de la nota final.