



Universidad Nacional de La Matanza
Departamento de Ingeniería e
Investigaciones Tecnológicas

Tecnicatura Universitaria en Desarrollo de Aplicaciones para Dispositivos Móviles

Programación Móvil 1

TP “Taller de reparación”

Profesores:

- Pardo, Sebastian
- Bayarri, Brian
- Hoz, Aylen

Un taller de autos quiere implementar un sistema de generación de facturas a partir del historial de reparaciones.

El taller clasifica los autos en 2 categorías:

- Con seguro: Los autos que poseen seguro presentan: un tope de cobertura del costo de reparación (en pesos), el cual es aplicado cuando el costo de la reparación sea igual o mayor al monto indicado por el seguro. Solo se aplica 1 vez al mes, y en la primera reparación del mes realizada al vehículo.
- Sin seguro: No poseen un tope de cobertura.

Existen dos tipos de clientes que el taller maneja:

- Clientes regulares: Son aquellos clientes cuyo vehículo ha sido reparado al menos una vez en el taller. A este tipo de clientes se les ofrece un descuento del 15% si la reparación supera los \$15.000.
- Clientes nuevos: Son aquellos clientes cuyo vehículo ha sido reparado solo una vez en el taller. A este tipo de clientes se les ofrece un descuento del 40% en la primera y única reparación, sin importar el costo de reparación.

El costo de reparación se calcula a partir de 2 factores:

- Mano de obra: Se cobrará \$500 por cada hora invertida en la reparación.
- Repuestos: Cada repuesto utilizado en la reparación presenta un costo por unidad. Puede usarse más de un repuesto y más de una unidad de repuesto en una reparación.

Recordar aplicar los descuentos correspondientes y, en caso de aplicar ambos descuentos (por seguro y por tipo de cliente), primero aplicar el descuento por cliente y luego por seguro. En ambos casos, se compara las condiciones de descuento con el monto total sin descuentos aplicados.

Además, los repuestos presentan stock disponible, por lo que es importante validar que existan las unidades suficientes para la reparación antes de generar la factura.

El objetivo de usted como desarrollador es calcular el monto total a pagar por un cliente por una reparación, dado la información provista por el operador del sistema, para así generar la factura correspondiente.

Para generar la factura de una reparación, el operador del sistema tendrá la opción de ingresar el código de cliente y código de reparación o solo código de reparación.

Los datos de un cliente (client) se almacenan de la siguiente forma:

```
{  
    "code": Int,  
    "name": String,  
    "surname": String  
}
```

Siendo:

- *'code'* el código del cliente, que lo identifica.
- *'name'* el nombre del cliente.
- *'surname'* el apellido del cliente.

Los datos de un vehículo (vehicle) se almacenan de la siguiente forma:

```
{  
    "numberPlate": String,  
    "clientCode": Int,  
    "insuranceCap": Double,  
    "insuranceCoverageFrom": Double (monto de reparación a partir el cual aplica el seguro)  
}
```

Siendo:

- *'numberPlate'* la patente del vehículo, cuyo formato es de 3 letras y 3 números. Ejemplo: AYH 458
- *'clientCode'* el código del cliente y dueño del vehículo.
- *'insuranceCap'* el tope de reintegro del seguro.
- *'insuranceCoverageFrom'* el monto de reparación a partir del cual aplica el seguro.

Utilizamos la patente para identificar el vehículo, ya que en este sistema consideramos que es única e irrepetible.

Los datos de un repuesto (sparePart) se almacenan de la siguiente forma:

```
{  
    "code": Int,  
    "name": String,  
    "price": Double,  
    "unitsInStock": Int  
}
```

Siendo:

- *'code'* el código del repuesto, que lo identifica.
- *'name'* el nombre del repuesto.
- *'price'* el costo del repuesto por unidad.
- *'unitsInStock'* las unidades en stock disponibles del repuesto.

Los datos de una reparación (repair) se almacenan de la siguiente forma:

```
{  
    "code": Int,  
    "clientCode": Int,  
    "completionDate": LocalDate,  
    "sparePartsUsed": Map<Int, Int>,  
    "hoursWorked": Int
```

}

Siendo:

- *'code'* el código de la reparación, que la identifica.
- *'clientCode'* el código del cliente y dueño del vehículo reparado.
- *'completionDate'* la fecha en la que se efectuó la reparación.
- *'sparePartsUsed'* los repuestos utilizados en la reparación. Se indica de cada repuesto: código y unidades utilizadas.
- *'hoursWorked'* las horas de mano de obra invertidas en la reparación.

A la hora de solicitar la factura, el sistema deberá mostrar el siguiente mensaje:

Fecha de emisión: *[fecha de hoy]*, fecha de reparación: *[repair.completionDate]*

Reparación #*[repair.code]*

La siguiente factura corresponde al cliente #*[repair.code]*: *[client.name]* *[clientsurname]*

El vehículo ingresado tiene la patente *[vehicle.numberPlate]*.

¿Tiene seguro? *[Mostrar 'SI' o 'NO' dependiendo si el vehículo tiene seguro o no]*

Los repuestos utilizados en la reparación fueron:

#*[sparePartsUsed.code]*: *[spareParte.name]*

Precio por unidad: *[sparePart.price]*

Unidades utilizadas: *[sparePartsUsed.unitsUsed]*

Subtotal: *[monto total por unidades de repuesto]*

....

Subtotal de repuestos: *[monto total por respuestos utilizados]*

La mano de obra:

Horas trabajadas: *[repair.hoursWorked]*

Monto por hora: *[precio de mano de obra por hora]*

Subtotal: *[monto total por mano de obra]*

Los descuentos aplicados en la reparación fueron:

Por cliente: *[descuento por tipo de cliente en pesos]*

Por seguro: *[descuento por seguro]*

Monto total: *[monto total de la factura]*

Además de permitir consultar una factura por código de cliente y código de reparación o solo código de reparación, el sistema debe permitirle a un administrador consultar el costo total en reparaciones de un cliente a partir de su historial a lo largo de toda su vida y consultar la ganancia total en reparaciones del taller a lo largo de toda su vida.

Para eso, debe realizar un menú de opciones donde el administrador pueda seleccionar las acciones a realizar y, en consecuencia, realizar la acción.

Notas:

- No necesita hacer uso de una base de datos para este ejercicio.
- Todas las reglas que el taller usa para calcular el costo total fueron dadas anteriormente. No necesita aplicar ninguna otra regla.

Requisitos:

- Realizar una UI con constraint layout.
- Hacer uso de recycler view.
- Hacer uso de navigation components.
- Hacer uso diferentes tipos de clases.
- Hacer uso de herencia y/o polimorfismo.
- Hacer uso de funciones lambdas.
- Tener tratamiento de excepciones.

Ejemplo de factura:

Para el repositorio dado, si el operador ingresa el código de cliente 1 y el código de reparación 1 para generar una factura, la salida será:

Fecha de emisión: **17/04/2022**, fecha de reparación: **15/04/2021**

Reparación **#1**

La siguiente factura corresponde al cliente **#1: Juan Dominguez**

El vehículo ingresado tiene la patente **FGH 698**.

¿Tiene seguro? **SI**

Los repuestos utilizados en la reparación fueron:

#1: Batería

Precio por unidad: **8000.0**

Unidades utilizadas: **1**

Subtotal: **8000.0**

#2: Bujía

Precio por unidad: **200.0**

Unidades utilizadas: **2**

Subtotal: **400.0**

Subtotal de repuestos: **8400.0**

La mano de obra:

Horas trabajadas: **10**

Monto por hora: **500**

Subtotal: **5000**

Los descuentos aplicados en la reparación fueron:

Por cliente: **0**

Por seguro: **10000.0**

Monto total: **3400.0**