

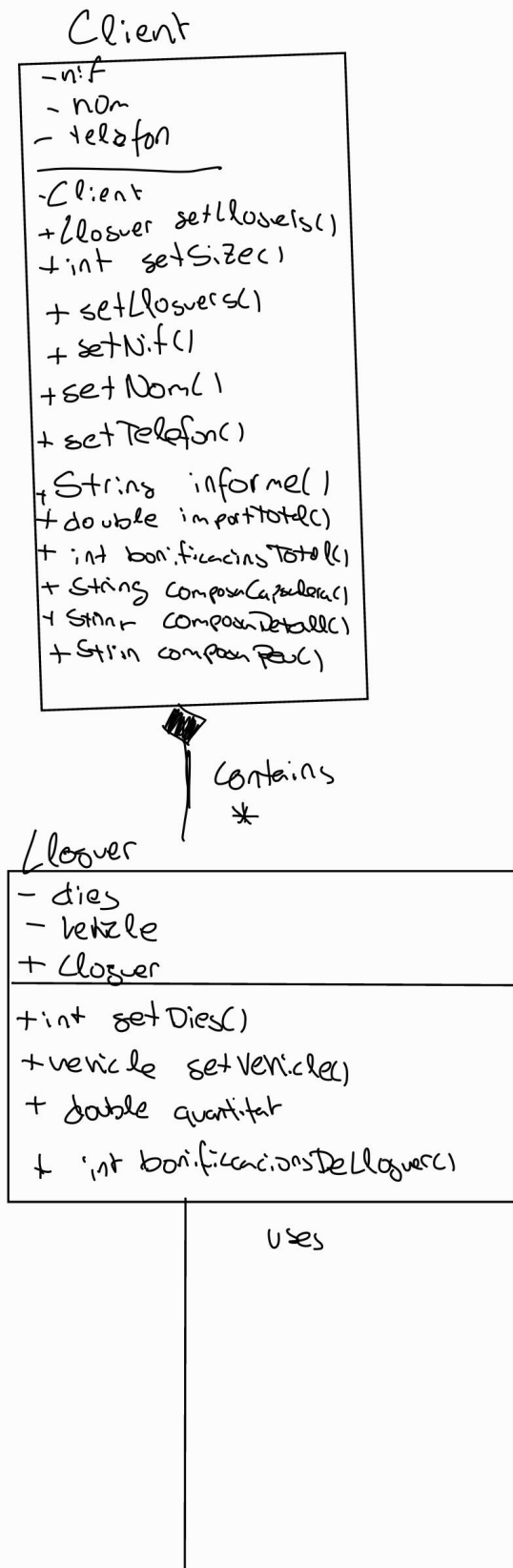
REFACCIÓ

Introducció

La refacció és la pràctica de millorar la llegibilitat i mantenibilitat del codi tot i que de vegades pugi fins i tot empitjorar el temps d'execució d'aquest.

És una part fonamental de la nostra feina ja que necessitem mantenir i millorar la llegibilitat del codi.

Diagrama de classes



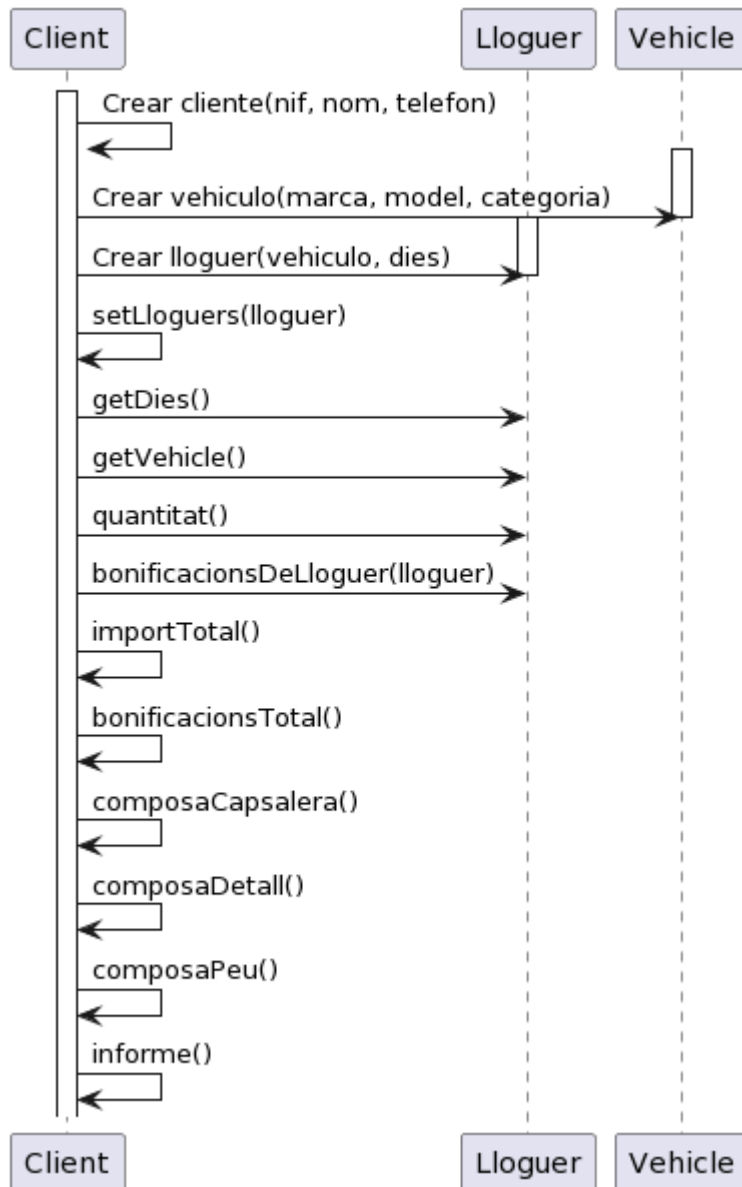
Vehicle

Enum Categoria

- marca
- model
- categoria

+ Vehicle
+ String getMarca()
+ String getModel()
+ Categoria getCategoria()
+ setMarca(String)
+ setModel(String)
+ setCategoria(int)

Diagrama de seqüència



Extracció

L'extracció es basa en canviar una variable per un mètode, cosa que fa que sigui més fàcil entendre el perquè d'aquesta.

Molts cops a l'hora de rellegir un codi és saber la raó per la qual el que hi al codi està al codi. El més difícil no es entendre el que fa sinó el motiu de que hi sigui, extraient el codi per tal de veure millor els passos que hi han a l'hora d'executar-lo i es pot reubicar a llocs on té més sentit que hi sigui.

Moviment

La base del moviment a la refacció és que un tros de codi pot tenir més sentit en una classe que en un altre, com passa a l'exercici 52_17 on movem el mètode de la classe Client a la classe Lloguer. Al fer això es veu més clarament que està relacionat amb la classe Lloguer, potser en un exercici amb poques classes com aquest és obvi que crida a dita classe, però en macroprojectes no ho és tant i el fet de saber a quina classe pertany facilita molt la vida del programador que haurà de mantenir i millorar el codi.

52_22

Als últims 2 exercicis el que he fet ha sigut fer un mètode que calculi les bonificacions en comptes de canviar el valor d'aquestes al informe. Posteriorment el que he fet ha sigut moure-ho a la classe Lloguer. Fer això millora la llegibilitat del codi i s'entén d'una manera molt més clara el càlcul de les bonificacions a part de veure que les bonificacions es calculen segons el lloguer.

52_25

A aquest exercici el que he fet ha sigut treure les variables total i bonificacions del mètode informe i les hem passat a ser un mètode per si mateixes. Passa el mateix que als exercicis anteriors ja que gràcies a aquests canvis el codi queda molt més clar, tant la forma de calcular les bonificacions com la de l'import total.

52_27

A aquest exercici he separat el mètode `informe()` en tres parts, la capselera, el detall i el peu. Amb això es redueix la quantitat de línees del mètode `informe()` i el fa molt més amè a l'hora de modificar-lo o entendre-ho.

Conclusions

Poc a poc vaig veient com millora la qualitat del codi, és més fàcil llegir, modificar i entendre el codi. Tot i que pot ser una mica tediós a l'hora de programar és molt fàcil veure per què és una pràctica tan estesa i important al nostre gremi.