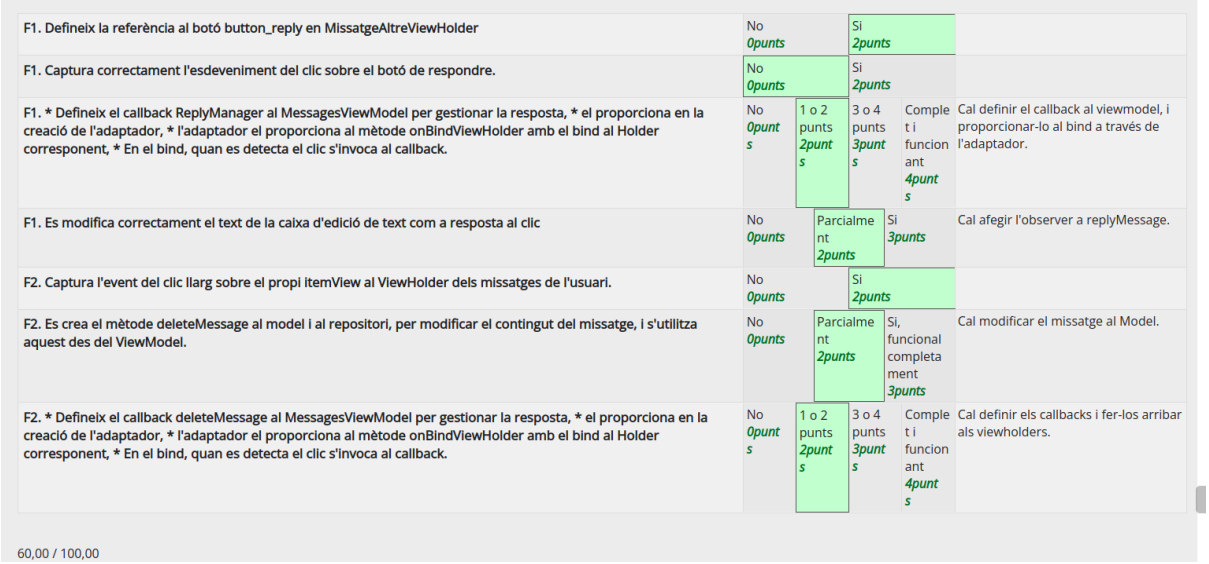
OLDER Exam:
Tenint en compte aquesta rubrica anem a realitzar les modificacions necesaries



F1. Defineix la referència al botó button_reply en MissatgeAltreViewHolder	No *0punts*		Si *2punts*		
F1. Captura correctament l'esdeveniment del clic sobre el botó de respondre.	No *0punts*		Si *2punts*		
F1. * Defineix el callback ReplyManager al MessagesViewModel per gestionar la resposta, * el proporciona en la creació de l'adaptador, * l'adaptador el proporciona al mètode onBindViewHolder amb el bind al Holder corresponent, * En el bind, quan es detecta el clic s'invoca al callback.	No *0punts*	1 o 2 punts *2punts*	3 o 4 punts *3punts*	Complet i funcionant *4punts*	Cal definir el callback al viewmodel, i proporcionar-lo al bind a través de l'adaptador.
F1. Es modifica correctament el text de la caixa d'edició de text com a resposta al clic	No *0punts*	Parcialment *2punts*	Si *3punts*		Cal afegir l'observer a replyMessage.
F2. Captura l'event del clic llarg sobre el propi itemView al ViewHolder dels missatges de l'usuari.	No *0punts*		Si *2punts*		
F2. Es crea el mètode deleteMessage al model i al repositori, per modificar el contingut del missatge, i s'utilitza aquest des del ViewModel.	No *0punts*	Parcialment *2punts*	Si, funcional completament *3punts*		Cal modificar el missatge al Model.
F2. * Defineix el callback deleteMessage al MessagesViewModel per gestionar la resposta, * el proporciona en la creació de l'adaptador, * l'adaptador el proporciona al mètode onBindViewHolder amb el bind al Holder corresponent, * En el bind, quan es detecta el clic s'invoca al callback.	No *0punts*	1 o 2 punts *2punts*	3 o 4 punts *3punts*	Complet i funcionant *4punts*	Cal definir els callbacks i fer-los arribar als viewholders.

60,00 / 100,00

CORRECTED Exam:
Model:

metode deleteMessage que no tenia fet a el model, a aquest i treballem la llista de missatges, busca el missatge en la llista i el usuari, si concorda amb la llista el misatge canvia, Sino ens retorna una errada. També hi habia probat a utilitzar try catch per a controlar excepcions per fer una bona practica, pero sembla que eu deixe així.

Repository ara:

```
fun deleteMessage(msg: Message) {
      Messages.deleteMessage(msg)
  }
```

Repository abans:

```
fun deleteMessage(msg: Message){msg.text = "mmssg deleted"};
```

He canviat la forma de fer el supostBorrat , ja que abans sols hi canviaba ek text de el missatge, ara utilitze el metode deleteMessage que he creat a el model, que aquest el elimina, sustituint el missatge. Anteriorment modificaba el missatge pero no estaba implementant-ho correctament, ja que no eliminaba el missatge de la llista .

MessagesWindow:
Hi tenia ubicada la funcio de click de respondre, on no hi debia, hi de una forma incorrecta.
Simplement he realitzat el canvi de :

```
viewModel.resposta.observe(this){
        binding.MessageText.setText(it)
```

} aquest canvi es realitzat degut a que al treballar amb M-V-V-M estem llegint constantment utilitzant el observer de el livedata per a llegir i actualizar el contingut

MessagesViewModel Actual:
canvis realitzats->

Creació de `resposta` com a `LiveData` per a observar les modificacions.

Creació de `_resposta` com a `MutableLiveData` per a guardar el missatge.

El mateix procés s'aplica a `adaptador` i `_adaptador`. Aquestes propietats gestionen l'adaptador utilitzat per al `RecyclerView` que mostra els missatges.

Canvi dels noms de les funcions i afegit dels logs requerits, que segons la rúbrica estaven marcats com a no realitzats.

Funcions de callback per a esborrar i respondre (l'únic canvi visible és a la funció `respondre`, amb el missatge que es mostrarà, guardat a `resposta`, que és un `LiveData`).

MissatgeAltreViewHolder actual:

s'ha afegit el click per a respondre dins de la funcio bind per la cual será l'utilitzada per a respondre .Aquesta funcio crida a la funció `gestorClick` amb el missatge i la vista com a paràmetres. Per a permetre respondre a un missatge quan es fa clic en el botó de resposta.

MissatgesAltreViewHolder: afegim a el missatges adapter el listener reply y listener remove per a actuar com a callbacks per a implementar la funcionalitat de borrar i editar segons la funcionalitat que vulgam. l'altra modificacio es invocarles dins del onbindViewholder segons el que vulgam depenent del missatge.

MissatgeViewHolder: El `MissatgeViewHolder` s'utilitza per a enllaçar les dades del missatge amb la vista. Dins del mètode `bind`, es posa el text del missatge i l'hora actual a la vista.

A més, s'ha afegit un gestor de clics de llarga durada (`setOnLongClickListener`) a la vista. Quan l'usuari fa una pulsació llarga en un missatge, es crida a la funció `gestorClick`, passant el missatge i la vista (`it`). Aquesta funció `gestorClick` és el `listenerRemove` que es va passar a l'adaptador, i es pot utilitzar per a implementar la funcionalitat d'eliminació del missatge.

Això permet que l'adaptador interactue amb altres parts de l'aplicació, com ara la interfície d'usuari i la lògica de l'aplicació, sense haver de gestionar directament aquestes interaccions

RESUM:

Mala colocació de callbacks, falta de funció a el model per a el borrat. Faltaban les funcions de els clicks tant el llarg com el de respondre a els bind’s corresponents. ús incorrecte de mutableliveData y livedata per a treballar amb la resposta i el adaptador. Falta d’utilitzacio de listenerreply y listeneremove per a gestionar les llistes.