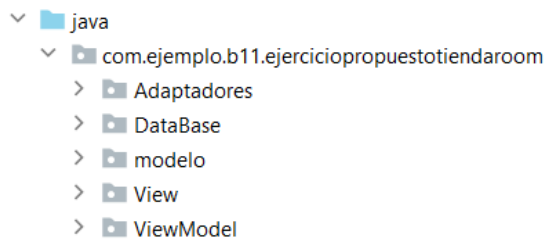


Exercici Proposat Tienda Room

[Descargar estos apunts](#)

Crearem una aplicació per a aprofundir més en la **gestió de Bases de dades amb Room**. Es tractarà d'implementar una mini botiga perquè els clients puguin fer comandes d'una sèrie d'articles que ofereix aquesta. En la següent imatge es mostrarà l'estructura de paquets que podeu seguir, perquè els components de l'aplicació estiguin més organitzats:



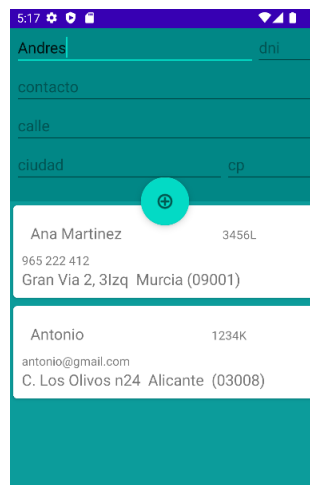
A més per a no perdre temps en la implementació de les vistes i els adaptadors, se us passen com a recursos adjunts tots els fitxers necessaris, encara que podeu personalitzar l'aplicació al vostre gust. *Important allotjar els adaptadors en la carpeta corresponent.*

Important: Per a millorar la independència del codi, en el paquet *ViewModel s'allotjarà:

- una classe object que contindrà la inicialització dels articles (es comentarà després de la primer ús per a evitar duplicats).
- la instanciació de la BD per a poder realitzar el seu ús.
- a més dels mètodes necessaris per a comunicar-se amb les funcionalitats de Dao, de manera que està serà la classe que té relació directa amb les incloses en el paquet de l'arquitectura Room, database.

FragmentCliente

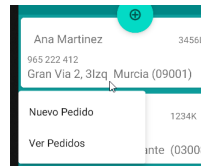
L'aplicació es presentarà amb un fragment principal que permetrà la gestió de clients: afegir, mostrar i eliminar. L'eliminació es farà amb el clic llarg, afegir mitjançant el FAB i els elements afegits es mostraran amb el recycler.



Tots els fragments i les activitats s'allotjaran en el paquet View.

Per a la gestió anterior només necessitarem l'entitat Client amb els camps que es visualitzen en la imatge i amb el dni com a clau primària.

Si premem de manera curta en un dels elements del recycler, es mostrarà un popup menú amb el següent aspecte:



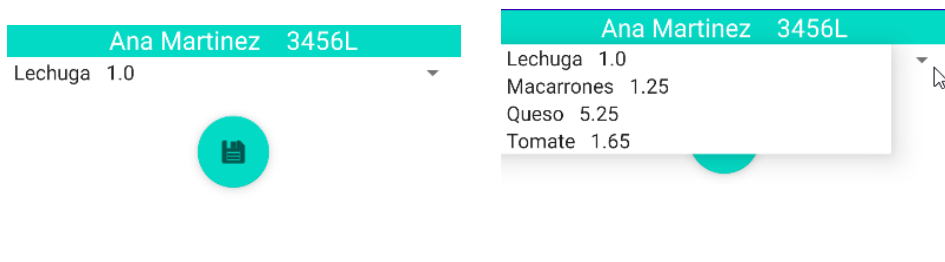
La selecció de cada opció del menú, carregarà el fragment corresponent.

FragmentPedido

Servirà per a realitzar la comanda del client premut, el primer que farà aquest fragment és carregar en un spinner els articulos a la venda de la botiga. Aquests articles es carregaran de manera automàtica en crear l'aplicació. Podem usar una classe object allotjada en ViewModel que retorne la següent llista d'articles, per exemple:

```
object InicializarArticulos {  
    fun listaArticulos():MutableList<ArticuloEntity>  
    {  
        var lista= mutableListOf<ArticuloEntity>()  
        lista.add(ArticuloEntity(0,"Macarrones",1.25f))  
        lista.add(ArticuloEntity(0,"Queso",5.25f))  
        lista.add(ArticuloEntity(0,"Tomate",1.65f))  
        lista.add(ArticuloEntity(0,"Lechuga",1.00f))  
        return lista  
    }  
}
```

Els elements d'aquesta llista s'inseriran en l'entitat Articulos per a posteriorment relacionar-los amb les comandes. L'aneu de l'article serà autoincrementable, per això s'insereix com 0 en el constructor.



En seleccionar un article del Spinner es mostrarà en el recycler inferior, podrem incrementar o decrementar la quantitat amb les icones corresponents, quan s'arribi a 0 l'article deixarà de mostrar-se en el recycler. Per a això utilitzarem una llista de ArticulosPedido, formada per una data classe amb els mateixos camps que la de articulos entity, més la quantitat que només serà necessària a l'hora de realitzar les comandes. *Aquesta data class s'allotjarà en el paquet model..*

Ana Martinez 3456L	
Queso 5.25	
2 Macarrones	+ X
1.25	
1 Tomate	+ X
1.65	
1 Queso	+ X
5.25	

En prémer sobre el FAB es procedirà a guardar les dades generals de la comanda en la taula Demanat, per a això crearem una entitat com la següent:

```
data class PedidoEntity(
    @PrimaryKey(autoGenerate = true) // El id será autogenerado.
    @ColumnInfo(name = "pedidoId")
    val pedidoId: Long,
    @ColumnInfo(name = "dniCliente")
    val dniCliente: String,
    @ColumnInfo(name = "total")
    val total: Float,
    @NonNull
    @ColumnInfo(name = "fecha")
    // Indicamos que siempre debe tener una fecha.
    val fecha: Long
)
```

En aquesta taula no es guardaran els articles, aquest procés el farem posteriorment, una vegada que ens funcione bé la inserció de la comanda.

Entre Client i Comanda hi ha una relació un a molts que hauràs de crear, tens la informació en les anotacions i en la pàgina oficial. Pots comprovar que la BD està guardant les dades correctament usant App Inspection.

Una vegada que tinguem les dades de la comanda guardada, passarem a guardar els articles que el componen. Per a això haurem de crear una relació **molts a molts entre Comanda i Article**, trobarem la informació en la pàgina principal. En aquest cas es crea una entitat intermitja ArticulosConPedido amb els dos aneu com a clau primària, i a més afegirem el camp quantitat. Per tant, en prémer sobre el FAB de guardar comanda, haurem d'inserir les dades en la table Comanda i a més inserir en la taula ArticulosConPedido l'aneu dels Articulos de la llista de la compra amb les seues quantitats, juntament amb l'aneu de la comanda.

FragmentListaPedidos

Aquest fragment es carregarà amb l'altra opció del menú Popup, i en aquest cas es visualitzarà la data i el cost de totes les comandes del client seleccionat.

Ana Martinez 3456L	
11/01/2023 18:19	9.25
11/01/2023 18:18	14.35

En prémer sobre una comanda, es mostrarà un dialogue amb els articles (quantitat, nom i preu) que componien la comanda.

En el dialogue es pot mostrar la informació com una cadena en la descripció.

Ana Martinez 3456L	
11/01/2023 18:19	9.25
11/01/2023 18:18	14.35
<div>CONTENIDO DEL PEDIDO</div> <div>2 Macarrones 1.25 1 Queso 5.25 4 Tomate 1.65</div> <div>OK</div>	