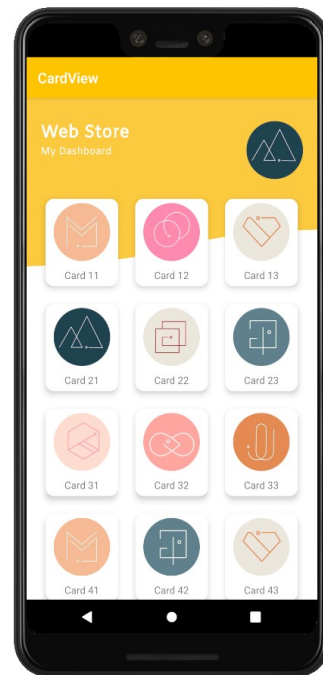
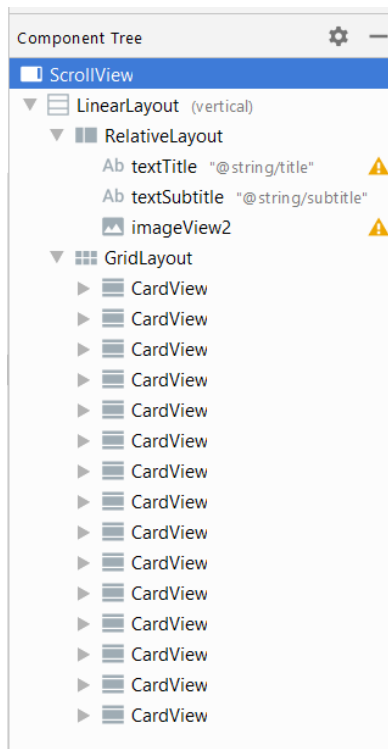


Exercici Layout 2

Per aquest projecte, has d'utilitzar les imatges el text, font i colors que tu prefereixis.

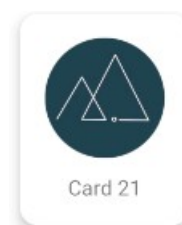
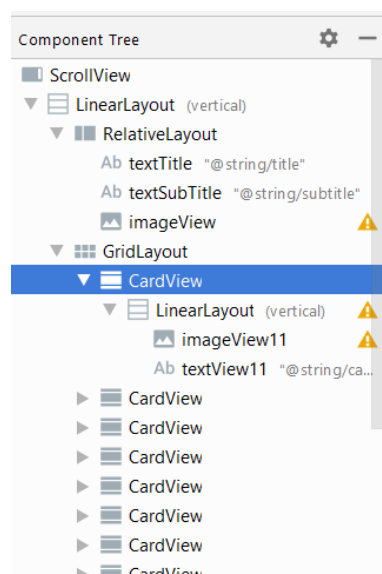
1. activity_main.xml

Realitzarem el següent:



2. CardView

Cada CardView tindrà un LinearLayout amb una imatge i un textView:



3. activity_card_view.xml

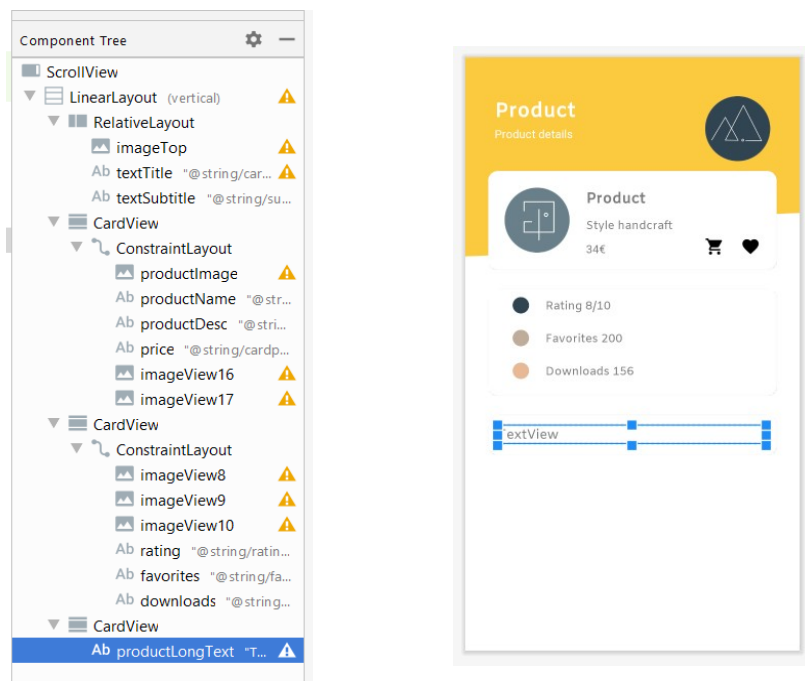
El layout contindrà el següent:

Un ScrollView per poder fer scroll per pantalla (verticalment)

Un LinearLayout on situarem (un RelativeLayout, 3 CardView's)

El RelativeLayout serà el mateix que en l'activitat anterior

Els CardViews contindran informació que es pot veure a la imatge



Fes servir `setOnClickListener` per 3 CardView's del MainActivity i crida l'activitat `CardViewActivity.java`, com que cada producte té informació diferent hauràs d'utilitzar `putExtra` i `getStringExtra` per cada un dels camps de la nova activitat, per exemple:

4. CardViewActivity.java

Aquesta vegada s'utilitzarà la mateixa activitat `CardViewActivity.java` per mostrar cada un dels productes.

A MainActivity.java:

- Defineix les constants:

```
public static String RATING = "com.example.edt16.RATING";
```

- Especifica el valor de la constant amb `putExtra` abans de cridar el intent:

```
intent.putExtra(RATING, rating);
```

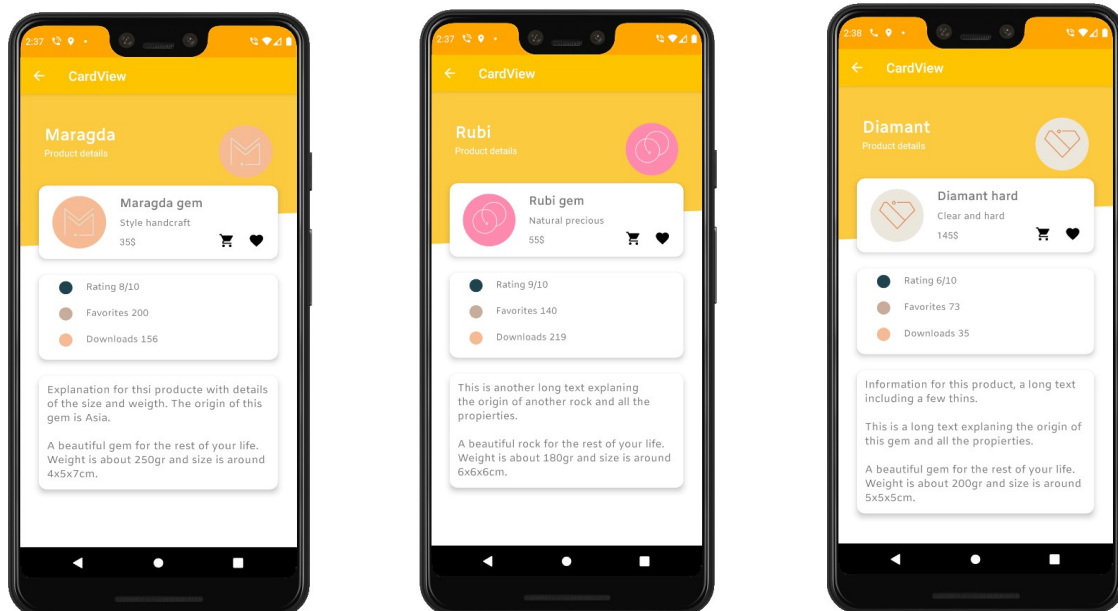
A CardViewActivity.java

- Recull el valor de la constant amb getStringExtra (per textos) o getIntExtra (per imatges)

```
String sRating = i.getStringExtra(MainActivity.RATING);
int iProductImage = i.getIntExtra(MainActivity.PRODUCT_IMAGE, 0);
```

- Mostra el valor amb setText (per textos) o setImageResource (per imatges):

```
rating.setText(sRating.toString());
productImage.setImageResource(iProductImage);
```



5. AndroidManifest.xml

Definirem l'activitat pare de l'activitat CardViewActivity, d'aquesta manera es mostra la fletxa de "back":

```
<activity android:name=".CardViewActivity"
    android:parentActivityName=".MainActivity">
</activity>
```