

## PAC1

### Presentació

En aquesta primera PAC es crearà l'estructura de l'aplicació amb la que treballarem durant el curs. Es treballaran els 3 primers mòduls dels apunts de l'assignatura.

### Competències

En aquesta PAC es treballaran les següents competències del Màster universitari en desenvolupament d'aplicacions per a dispositius mòbils:

- Utilitzar de manera efectiva els llenguatges de programació de les plataformes mòbils més representatives del mercat.
- Usar les eines i entorns de desenvolupament disponibles per a les plataformes mòbils més representatives del mercat.

### Objectius

En el desenvolupament de la PAC es persegueixen aconseguir els següents objectius:

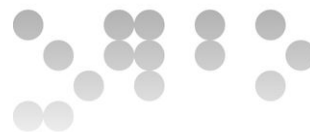
- Aprendre a desenvolupar una aplicació per a dispositius mòbils i tauletes amb l'estructura master-detail.
- Aprendre a utilitzar activitats i fragments en una aplicació.
- Realitzar llistes complexes amb RecyclerView.
- Mostrar targetes amb la vista CardView.
- Aprendre a utilitzar el ConstraintLayout.

### Descripció de la PAC/pràctica a realitzar

#### PART BÀSICA

#### Exercici 1 (25%): Estructura de l'aplicació

El primer exercici consisteix en crear una aplicació Android amb l'estructura master-detail. Aquesta estructura es caracteritza per estar composta de dues pantalles: una màster amb una llista d'ítems i una detall amb la informació específica de cada ítem. La característica especial d'aquest tipus d'estructura es que en pantalles petites es veurà primer la llista



d'ítems i en prémer sobre un d'ells, el seu detall; en canvi, en pantalles grans es veurà directament la llista d'ítems i el detall de l'ítem seleccionat a la mateixa pantalla.

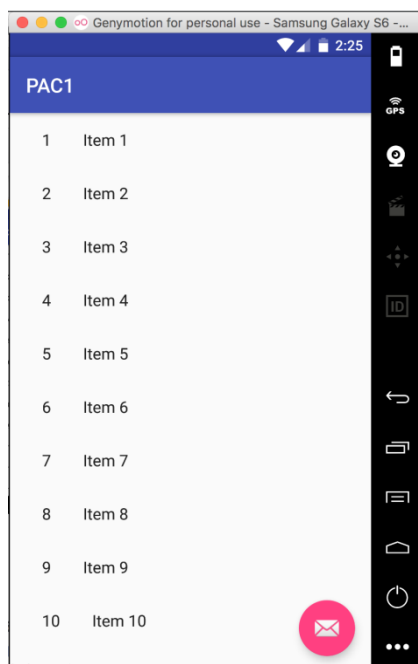
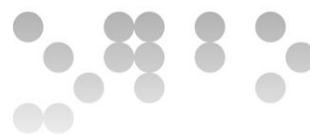
Obre l'Android Studio, crea una nova aplicació amb el nom i domini que vulguis i selecciona una aplicació per a mòbil i tauleta amb la versió mínima d'Android 4.0.

En la següent pantalla es selecciona el tipus d'activitat principal del projecte, en aquest cas s'utilitzarà "Add no activity". Això farà que Android Studio implementi en el nostre projecte buit.

Primerament, crea les dues activitats principals del projecte: **BookListActivity** i **BookDetailActivity**, per a la llista i el detall respectivament, i registra-les al fitxer Manifest.

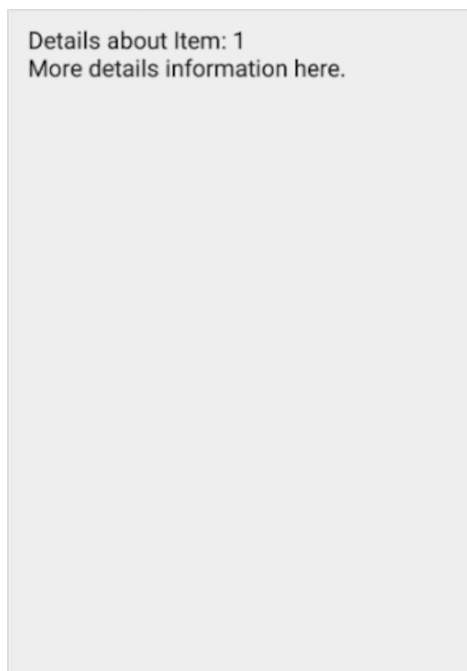
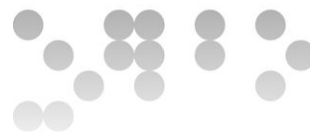
Per tal que es mostri un tipus de layout o un altre segons si estem en mòbils o tauletes serà necessari crear un layout XML diferent segons el tipus de resolució de pantalla per a l'activitat **BookListActivity**.

Crea el layout **book\_list.xml** per a la resolució w900dp amb una llista simple d'ítems i un frame layout per al detall, i un altre layout **book\_list.xml** a la carpeta de la resolució per defecte amb la llista de ítems només. Posa el mateix identificador per a la llista d'ítems, ja que aquesta llista es gestionarà des de la mateixa activitat. A **BookListActivity**, ompli aquest layout amb una llista per defecte (amb valors Ítem 1, Ítem 2, Ítem 3...), tal i com es mostra a la següent imatge:



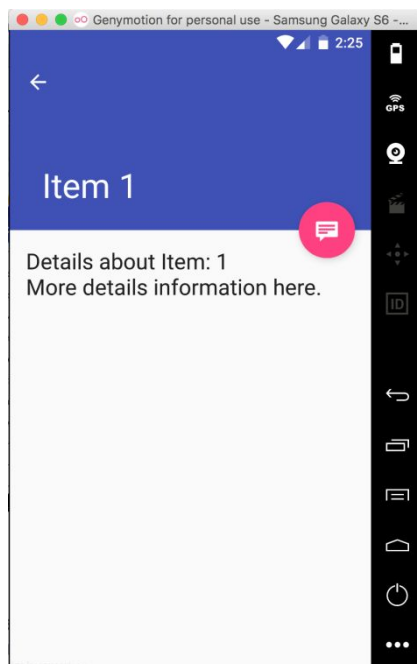
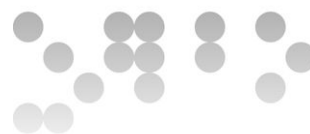
Des de l'activitat **BookListActivity**, detecta si el frame layout és present a l'activitat. En cas que sigui present significa que estem en una tauleta, per tant, s'haurà de mostrar el detall de la llista dins d'aquest frame.

Per a mostrar el detall de la llista tant en tauleta com en telèfon mòbil, serà necessari utilitzar un fragment. Aquest fragment s'afegirà al frame de **BookListActivity** o directament a **BookDetailActivity**. Per tant, crea un fragment **BookDetailFragment** on es mostraran els detalls del llibre, de moment crea uns detalls per defecte com es mostra a la següent imatge:



A **BookListActivity** afegeix aquest fragment al frame layout creat anteriorment, en cas que estem en una tauleta.

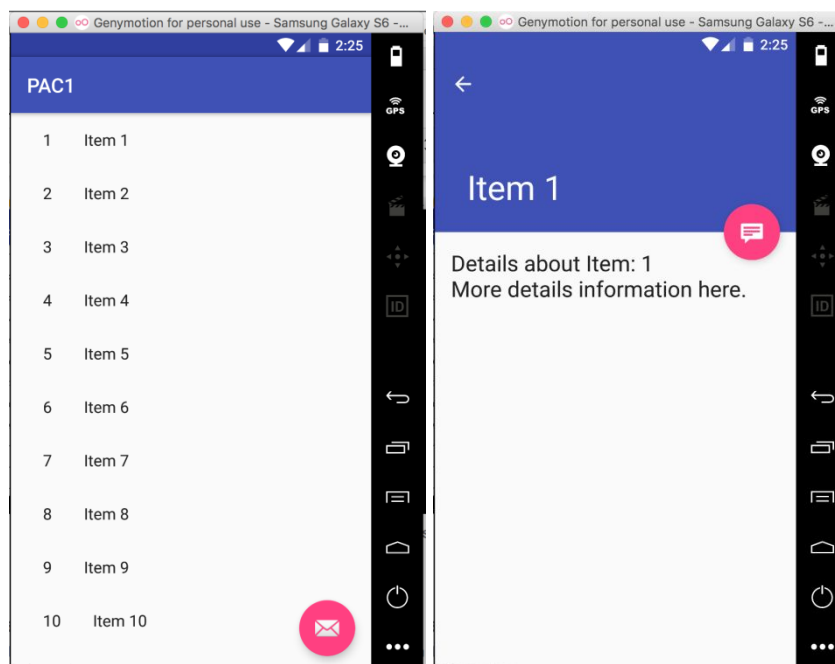
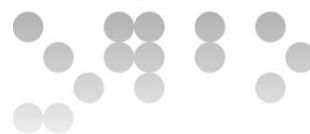
**BookDetailActivity** és l'activitat on es mostrarà el detall del llibre. Aquesta activitat s'utilitzarà només en cas de que estem en un telèfon mòbil, per tant, modifica el seu layout perquè tingui un frame on es pugui afegir el fragment anterior. Pots customitzar aquesta activitat tal i com es mostra a la següent imatge, perquè tingui una barra superior i el detall del fragment estigui baix:



Una vegada preparades aquestes activitats i el fragment, si executem el codi en una tauleta en horitzontal veurem que en la mateixa pantalla està la llista i el detall:



En canvi, si executem el codi en un telèfon tenim dos pantalles, la de la llista i la del detall:



## Exercici 2 (25%): Implementar una llista amb RecyclerView

Al segon exercici es crearà una llista de tipus RecyclerView per mostrar la llista de llibres en el primer fragment de l'aplicació i una vista de tipus CoordinatorLayout per mostrar els detalls dels llibres.

Primerament, fes una còpia del projecte creat a l'exercici anterior, ja que en els següents exercicis treballarem sobre l'estructura creada a l'exercici 1 i al final de la pràctica s'entregaran dos projectes: un projecte amb el codi de l'exercici 1 i un altre projecte amb la resta d'exercicis.

Abre el nuevo proyecto copiado en Android Studio para continuar.

Ara crearem el model del llibre: crea un nou paquet amb el nom "model" i la classe a "BookModel" amb una llista estàtica d'items de tipus "BookItem". Els nostres llibres tindran els següents detalls:

- Identificador (int)
- Títol (String)
- Autor (String)
- Data de publicació (Date)



- Descripció (String)
- URL imatge de portada (String)

Actualitza el model “BookItem” amb aquests elements.

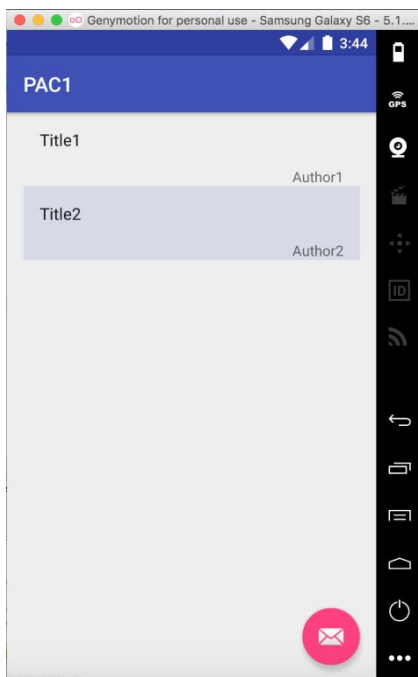
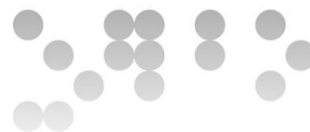
A continuació crea un parell de llibres de prova per poder veure els resultats:

```
/**  
 * An array of sample book items.  
 */  
public static final List<BookItem> ITEMS = new ArrayList<>();  
  
static {  
    BookItem book1 = new BookItem(0, "Title1", "Author1", new Date(), "Description", null);  
    BookItem book2 = new BookItem(1, "Title2", "Author2", new Date(), "Description 2", null);  
    ITEMS.add(book1);  
    ITEMS.add(book2);  
}
```

Una vegada preparat el contingut de prova, haurem de crear el RecyclerView que mostrarà la llista de llibres.

A l'activitat “BookListActivity.java” crearem el nostre RecyclerView per a mostrar els nostres llibres.

Crea dos layout XML per a mostrar els elements de la llista parells i imparells mostrant el següent disseny:



Crea un adaptador per a la llista que extendeixi de:

`RecyclerView.Adapter<SimpleItemRecyclerViewAdapter.ViewHolder>`

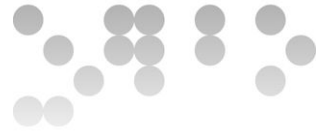
Al mètode “onCreateViewHolder”, carrega un layout per als elements parells de la llista i un altre per als elements imparells, segons el “viewType”:

`@Override`

```
public ViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {  
    View view = null;  
    // ===== INICI CODI A COMPLETAR =====  
  
    // ===== FI CODI A COMPLETAR =====  
    return new ViewHolder(view);  
}
```

Afegeix el mètode “getItemViewType” i retorna el tipus parell o imparell segons la seva posició:





```
private final static int EVEN = 0;
private final static int ODD = 1;

@Override
public int getItemViewType(int position) {
    int type;
    // ===== INICI CODI A COMPLETAR =====

    // ===== FI CODI A COMPLETAR =====
    return type;
}
```

Una vegada ja està la llista preparada, faltaria modificar la pantalla del detall. Com que ara no tenim un identificador entre els paràmetres dels llibres, utilitzarem la posició en la llista per a identificar el llibre. Al mètode “onBindViewHolder” modifica el contingut per a obtenir la posició i passar-la a la classe del detall:

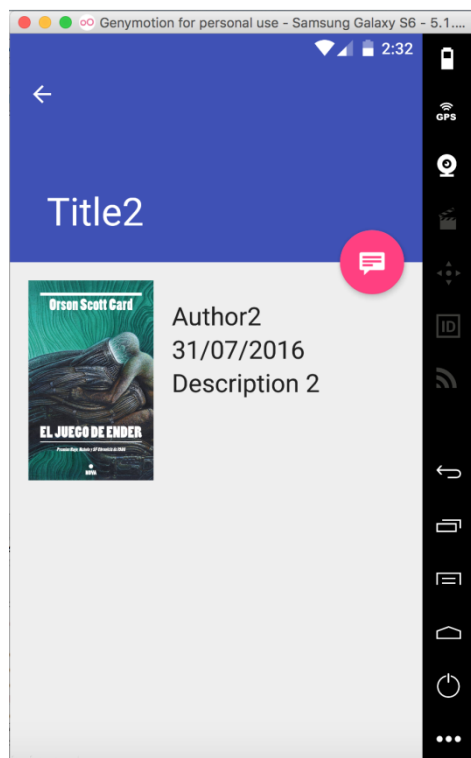
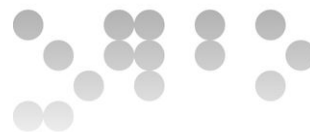
```
@Override
public void onBindViewHolder(final ViewHolder holder, final int position) {
    holder.mItem = mValues.get(position);
    holder.mTitleView.setText(mValues.get(position).title);
    holder.mAuthorView.setText(mValues.get(position).author);
    // ===== INICI CODI A COMPLETAR =====
    // Guarda la posició actual a la vista del holder
    // ===== FI CODI A COMPLETAR =====
    holder.mView.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            int currentPos;
            // ===== INICI CODI A COMPLETAR =====
            // Obtindre la posició del llibre on s'ha fet click de la vista
        }
    });
}
```



```
// ===== FI CODI A COMPLETAR =====  
if (mTwoPane) {  
    // ===== INICI CODI A COMPLETAR =====  
    // Iniciar el fragment corresponent a tablet, enviant l'argument de la posició seleccionada  
    // ===== FI CODI A COMPLETAR =====  
  
} else {  
    // ===== INICI CODI A COMPLETAR =====  
    // Iniciar l'activitat corresponent a mòbil, enviant l'argument de la posició seleccionada  
    // ===== FI CODI A COMPLETAR =====  
}  
}  
});  
}
```

Modifica els llocs on s'obté aquest argument per a obtenir un "int" en comptes d'un "string".

Per últim, modifica la vista del detall per a mostrar tots els detalls del llibre: l'autor, la data de publicació, la descripció i la imatge:



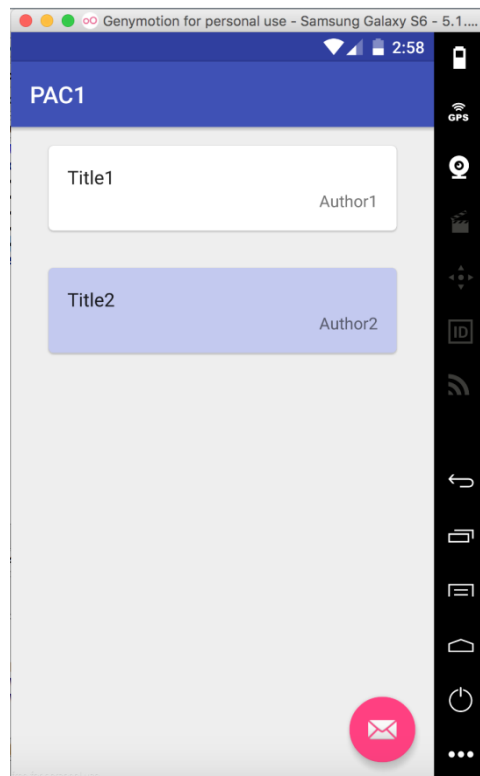
Com a imatge del llibre, de moment, utilitza una imatge estàtica.

### Exercici 3 (10%): Implementar targetes en una llista

Modificar els elements de la llista RecyclerView per a ser del tipus CardView.

Primer, inclou la llibreria de support d'Android que inclou les classes de CardView, per a poder utilitzar-la al projecte.

Després modifica els dos fitxers creats a l'exercici anterior (per als llibres en posició parell i els imparells), perquè utilitzin una vista de tipus "CardView":



Canvia l'elevació de la targeta a 2dp, un radi del cantó de la targeta de 4dp i color de fons blanc per a les posicions parelles i “#C3C9EF” per a les posicions imparelles.

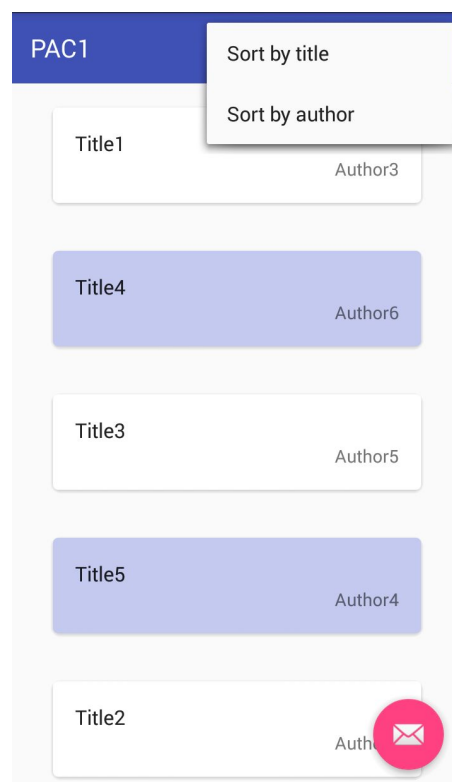


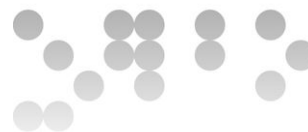
#### Exercici 4 (10%): Implementar mètodes d'ordenació

Implementa l'ordenació de la llista i crea un menú a la vista amb el mètode `onCreateOptionsMenu`. Afegeix dues opcions al menú que permetin ordenar la llista per títol o autor.

@Override

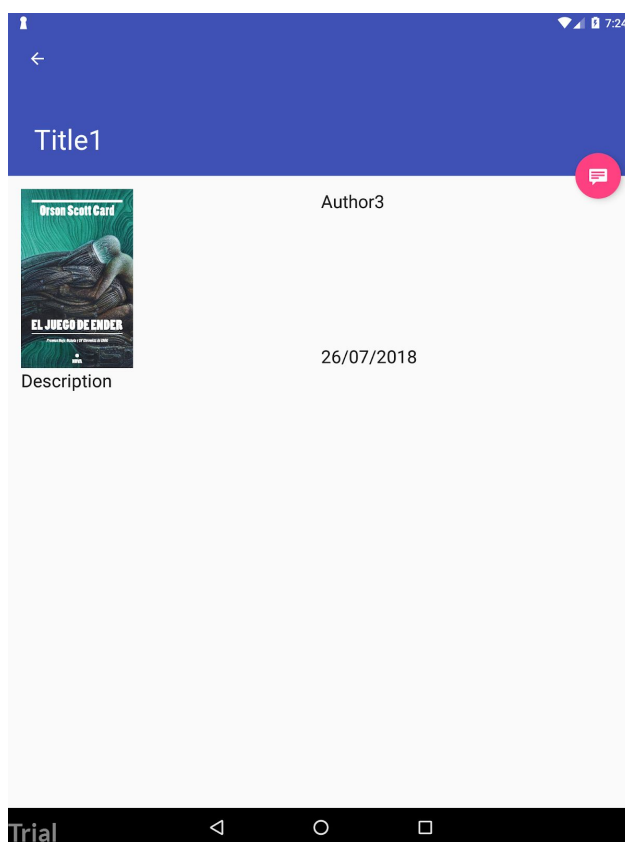
```
public boolean onCreateOptionsMenu(Menu menu) {  
    MenuInflater inflater = getMenuInflater();  
    inflater.inflate(R.menu.menu_list, menu);  
    return true;  
}
```





### Exercici 5 (10%): Modificar el detall utilitzant ConstraintLayout

Modifica el detall del llibre, utilitzant una ConstraintLayout, per aconseguir una estructura semblant a aquesta:



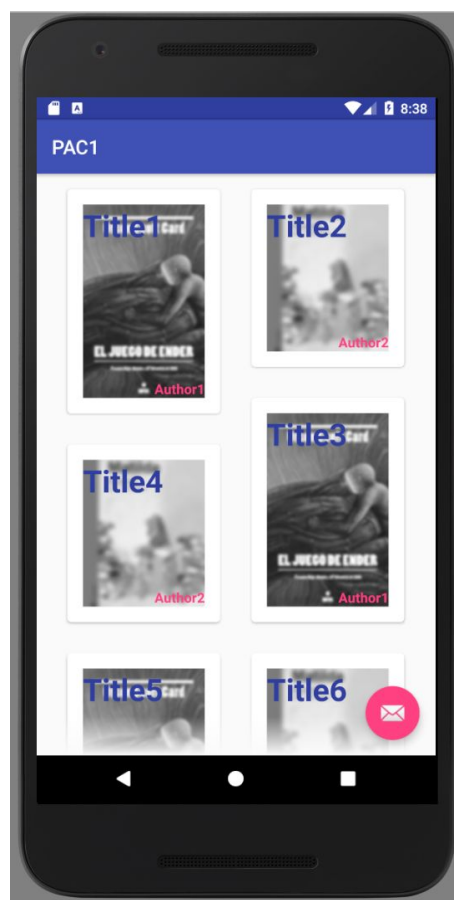
La descripció ha d'estar sota la imatge. La part d'imatge i descripció han d'ocupar un 50% de la pantalla. A la dreta, han d'aparèixer autor (a la part superior), i data (a la part inferior).

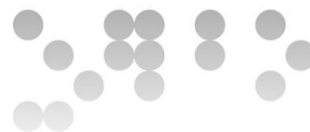


## PART AVANÇADA

### Exercici 6 (10%): Implementar una llista amb elements de diferent mida

Modifica els elements de la llista RecyclerView per a mostrar una llista com la que es mostra a continuació:





La mida dels elements ha de ser depenent de la mida de la imatge del llibre i l'última filera de llibres ha de tindre la part inferior descolorida per a indicar a l'usuari que la llista és més gran. Aplica un degradat a les imatges perquè sigui més fàcil llegir el text.

Mostra aquest tipus de llista només en telèfon i afegeix més llibres a la llista per a poder veure l'efecte.

\* Afegeix uns quants llibre de prova amb imatges de diferents mides per a comprovar-ho.

## PART TEÒRICA

### Exercici 7 (10%): Java o Kotlin

Explica per què escollir Java o Kotlin per començar un projecte nou a dia d'avui. Avantatges i possibles inconvenients de cada llenguatge.

## Recursos

### Bàsics

- Mòdul 1: Estructura d'una aplicació d'Android
- Mòdul 2: Controladors: activitats, fragments i serveis
- Mòdul 3: Interfície gràfica: layouts

### Complementaris

- Documentació oficial d'Android

## Criteris de valoració

- La PAC s'ha de resoldre de forma individual
- Cada exercici té el següent pes:

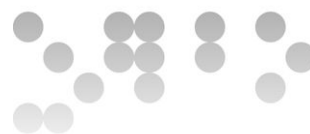
### PART BÀSICA

- Exercici 1: 25%
- Exercici 2: 25%
- Exercici 3: 10%
- Exercici 4: 10%
- Exercici 5: 10%

### PART AVANÇADA

- Exercici 6: 10%





## PART TEÒRICA

- Exercici 7: 10%
- El codi font s'ha de **comentar** i l'estil de programació ha de ser net i pulit.
- L'entrega són dos projectes: entrega un ZIP amb el projecte l'exercici 1 i un ZIP amb el projecte dels d'exercicis 2 al 6.

## Format i data de lliurament

S'ha de lliurar un arxiu ZIP amb nom **PAC1\_Cognom1Cognom2Nom.zip** amb el següents fitxers:

- Aplicació en format APK del projecte realitzats als exercicis 2 al 6
- Codi font de tot el projecte de l'exercici 1 (menys la carpeta de debug que no s'ha d'entregar)
- Codi font de tot el projecte del exercicis 2 al 6 (menys la carpeta de debug que no s'ha d'entregar)
- Document amb la resposta a la pregunta teòrica

Aquest arxiu s'ha de lliurar a l'espai Lliurament i registre d'activitats de l'aula **abans de les 23:59 del dia 24/03/2020. No s'acceptaran entregues fora de plaç.**

### Nota: Propietat intel·lectual

Sovint és inevitable, en produir una obra, fer ús de recursos creats per terceres persones. És per tant comprensible fer-ho

en el marc d'una activitat dels Estudis d'Informàtica, Multimèdia i Telecomunicació, sempre que això es documenti clarament i no suposi plagi en la pràctica.

Per tant, en presentar una activitat que faci ús de recursos aliens, s'ha de presentar juntament amb ella un document en

què es detallin tots ells, especificant el nom de cada recurs, el seu autor, el lloc on es va obtenir i el seu estatus

legal: si l'obra està protegida pel copyright o s'acull a alguna altra llicència d'ús (Creative Commons, llicència GNU,

GPL ...). L'estudiant haurà d'assegurar-se que la llicència que sigui no impedeix específicament seu ús en el marc de

l'activitat. En cas de no trobar la informació corresponent, haurà d'assumir que l'obra està protegida pel copyright.



S'hauran, a més, d'adjuntar els fitxers originals quan les obres utilitzades siguin digitals, i el seu codi font si correspon.