

## PRA1

### Presentació

En aquest primer lliurament de la pràctica es desenvoluparà una aplicació mòbil utilitzant Android Studio, l'entorn de desenvolupament oficial per a la plataforma Android. El treball es realitzarà de forma guiada a partir d'un projecte existent que haurà de completar-se segons les indicacions d'aquest enunciat.

### Competències

En aquest lliurament de la Pràctica es treballen les següents competències del Màster Universitari en Enginyeria Informàtica:

- Capacitat per a projectar, calcular i dissenyar productes, processos i instal·lacions en tots els àmbits de l'enginyeria informàtica.
- Capacitat per a modelar, dissenyar, definir l'arquitectura, implantar, gestionar, operar, administrar i mantenir aplicacions, xarxes, sistemes, serveis i continguts informàtics.
- Capacitat per assegurar, gestionar, auditar i certificar la qualitat dels desenvolupaments, processos, sistemes, serveis, aplicacions i productes informàtics.
- Capacitat de dissenyar i desenvolupar sistemes, aplicacions i serveis informàtics en sistemes encastrats i ubics.

També es treballen les següents competències del Màster Universitari en Desenvolupament d'Aplicacions per a Dispositius Mòbils::

- Identificar les peculiaritats del disseny d'aplicacions per a dispositius mòbils, especialment aquelles derivades de les limitacions del dispositiu.
- Utilitzar de forma efectiva els llenguatges de programació de les plataformes mòbils més representatives del mercat.
- Planificar i dirigir projectes tecnològics relacionats amb tecnologies mòbils.

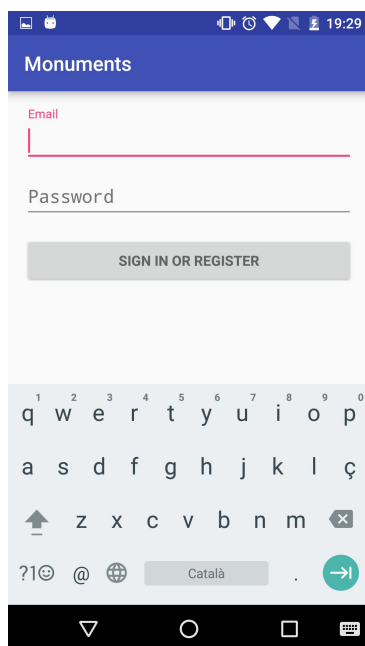
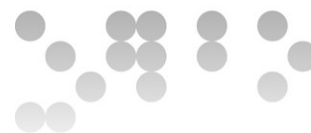
### Objectius

Els objectius concrets d'aquest lliurament de la Pràctica són que l'estudiant aprengui a :

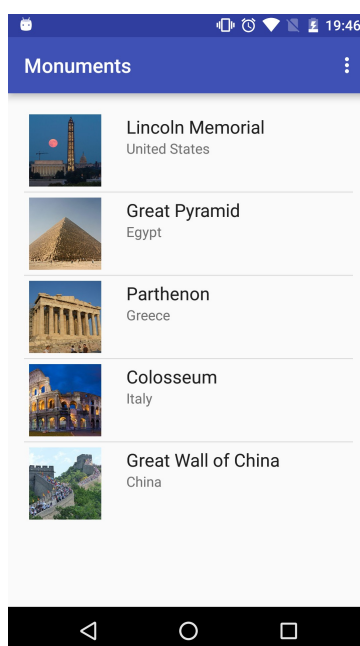
- Ser capaçs de dirigir un projecte relacionat amb les tecnologies de desenvolupament sobre mòbil, sabent que fer en cadascuna de les seves fases i proporcionant les eines necessàries per afrontar-lo amb garanties.
- Tenir un coneixement ampli i variat de les alternatives per al desenvolupament d'aplicacions mòbils.

### Descripció de la pràctica a realitzar

Moltes aplicacions comparteixen una arquitectura similar: una pantalla d'autenticació que dóna accés a una llista d'elements. Fent clic en un element accedim al detall del mateix



Pantalla de login



Llistat d'elements



Detall d'un element

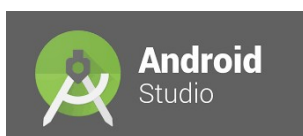
En aquesta pràctica, treballarem sobre aquest tipus d'aplicació. Concretament:

- En el primer exercici instal·larem Android Studio i coneixerem els seus diversos elements (entorn, SDK, simulador, eines de debug).
- En el segon exercici, estudiarem aspectes relacionats amb la programació concurrent (threads) dins d'una app. Si teniu alguna dificultat amb aquest exercici, podeu passar als següents ja que es poden resoldre de forma independent.
- El tercer i quart exercici es centren en les modificacions del disseny de les pantalles (layout) i en la comunicació de paràmetres entre les activitats.

### **Exercici 1. L'entorn (valoració d'un 20%)**

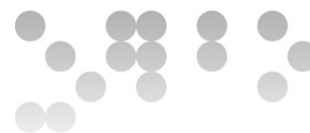
El 2013, Google va llançar la primera versió d'Android Studio, basat en IntelliJ IDEA, com IDE oficial per desenvolupament d'aplicacions per Android. Està disponible per a les plataformes Microsoft Windows, Mac OS X i GNU / Linux.

#### **Pas 1: Instal·lació d'Android Studio**



Descarrega Android Studio des de la següent URL:  
<https://developer.android.com/studio/index.html>

A l'enllaç trobareu informació sobre l'IDE i els requisits del sistema.

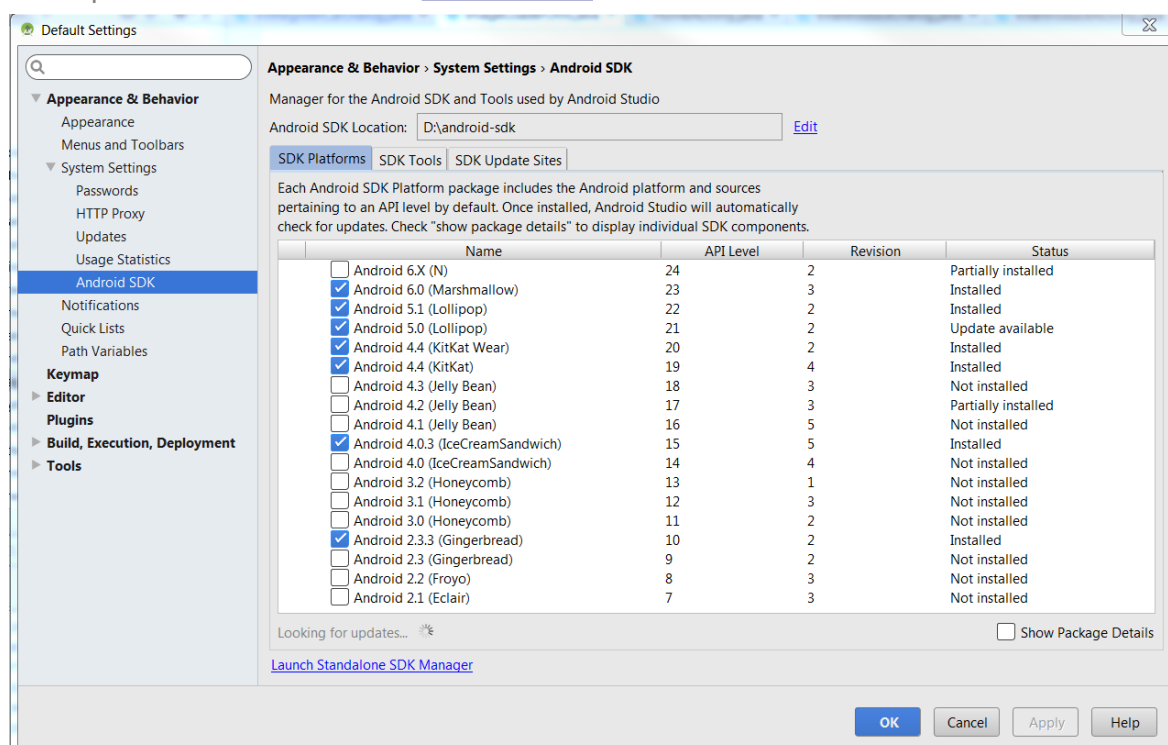


Si ja el teniu descarregat: Comproveu que el teniu actualitzat a la versió 2.2.0. Al Windows aquesta actualització de vegades consisteix en descarregar de forma manual la nova versió i descomprimir en un altre directori. A la carpeta bin podeu trobar els nous executables. A l'executar-los per primera vegada us demanaran accedir a la configuració de la versió antiga i heu d'acceptar aquesta petició.

## Pas 2: Instal·lació d'Android Studio

Hi ha múltiples versions de la plataforma Android, i cadascuna proporciona diferents APIs i funcionalitats als desenvolupadors a través d'un SDK (Software Development Kit). El SDK Manager és una eina d'Android Studio que ens permet configurar quines versions del SDK estan disponibles en el nostre entorn de desenvolupament.

Fent servir el SDK Manager, descarregueu la **versió 6.0, API Level 23**. Aquesta versió es correspon a la versió d'Android Marshmallow.



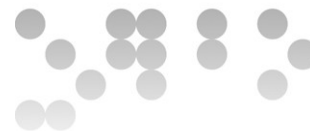
Respongueu a les següents preguntes:

**Pregunta 1.1.** ¿Quina versió mínima seria raonable utilitzar en una aplicació com la plantejada en aquest exercici? Justificar la resposta.

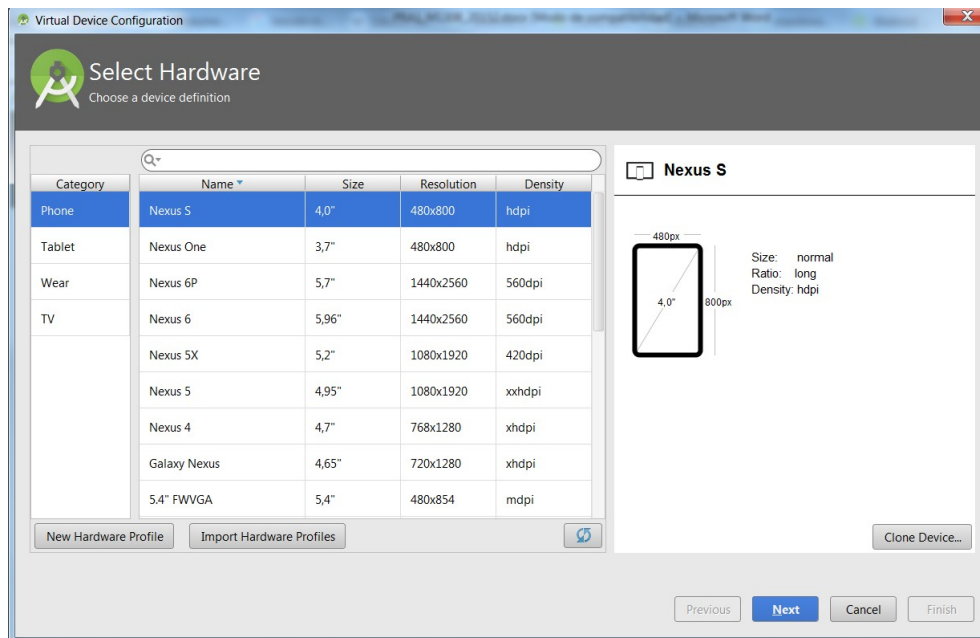
## Pas 3: Instal·lació d'Android Studio

Para poder ejecutar una aplicación, deberemos definir en el simulador cuáles son las características del dispositivo a emular (*Android Virtual Device* o AVD). Vamos a realizar este proceso utilizando la herramienta AVD Manager.

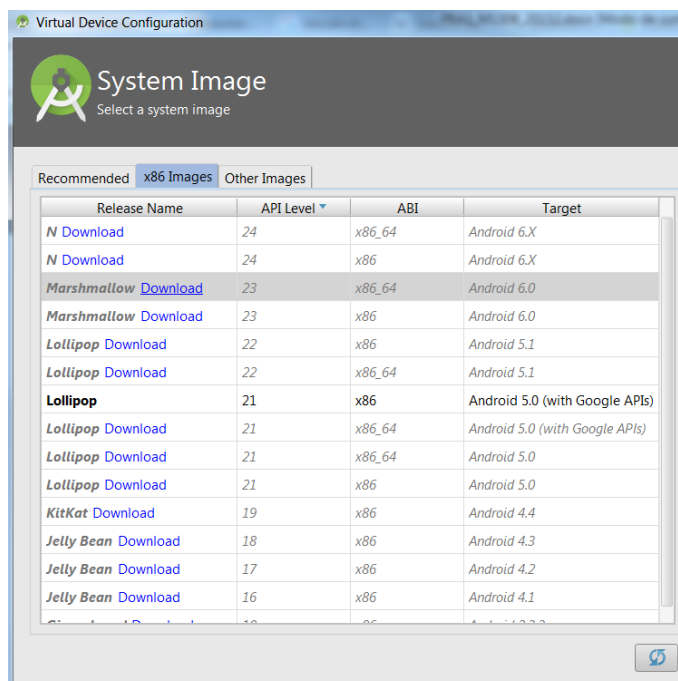
Cread un AVD de la versión Marshmallow 6.0 Api Level 23 (sin Google API). Para crear este AVD seguid estos Pass.



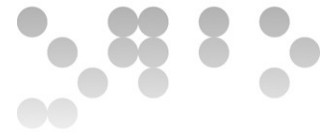
- A l'AVD Manager, premeu a “Create Virtual Device”.
- Seleccioneu “Nexus S”. Aquest serà el hardware emulat al nostre AVD.



- Ara seleccioneu la imatge del sistema operatiu. Si no la teniu disponible, localitzeu la versió de Marshmallow API level 23 x86 i premeu a l'opció de “Download”.



Com alternativa, un altre emulador d'Android que és molt popular és el [Genymotion](#). Si disposeu d'un dispositiu mòbil amb sistema operatiu Android configura els drivers perquè Android Studio reconegui el dispositiu quan es connecti a l'ordinador.

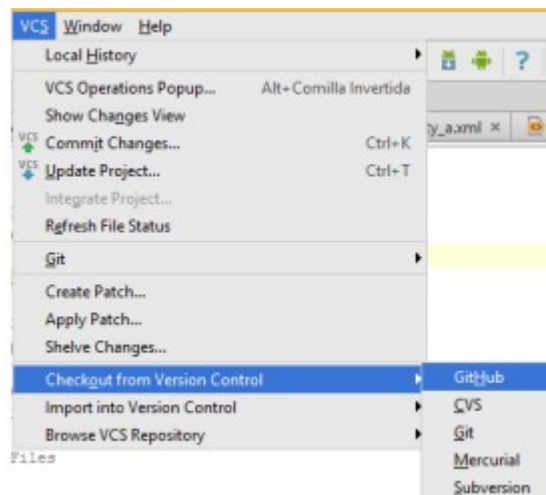


Un cop configurat, arranca l'emulador i explica si hi ha hagut alguna incidència durant el procés. En cas contrari, adjunta una captura de pantalla de l'emulador.

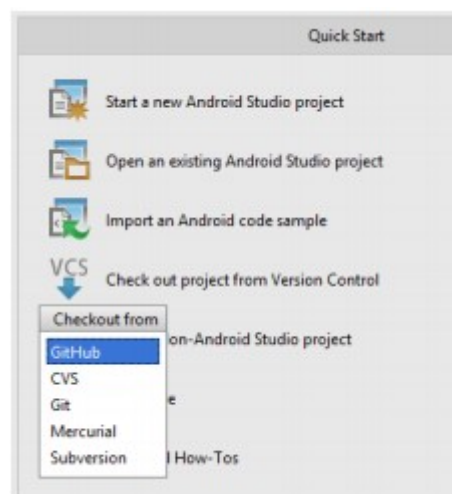
#### Pas 4: Descarregar l'aplicació

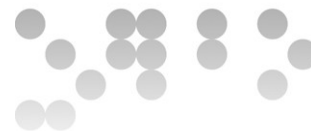
El codi de l'exercici està disponible al repositori GitHub (plataforma de desenvolupament col·laboratiu de programari per emmagatzemar projectes utilitzant el sistema de control de versions Git) de la UOC que us podreu descarregar seguint els següents passos:

- 1) Cal verificar que teniu disponible Git a l'ordinador. El podeu descarregar entrant al següent enllaç: <https://git-scm.com/downloads>.
- 2) Una vegada instal·lat, obrir Android Studio i anar a la pestanya VCS → Checkout from version control → GitHub.



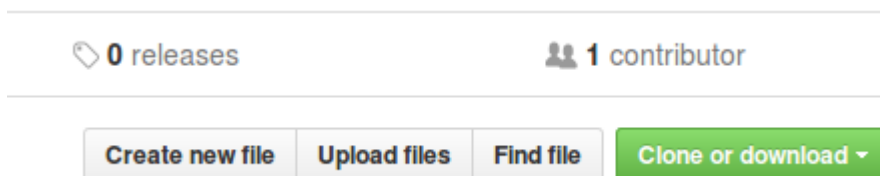
- 3) Si no teniu cap projecte obert, directament des de Quick Start i trieu l'opció Checkout from version control → GitHub.





- 4) Un cop seleccionat, a l'opció Vcs Repository URL escriviu la següent URL [https://github.com/uocappseimt/TDDM\\_Practica.git](https://github.com/uocappseimt/TDDM_Practica.git) i trieu la ubicació de destí.

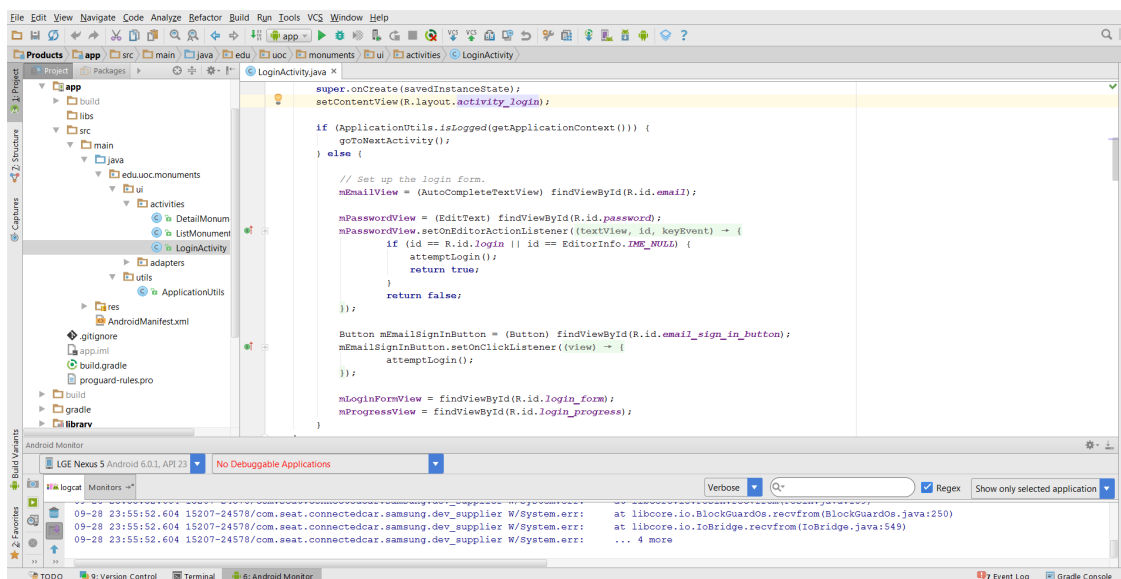
Com alternativa, podeu descarregar el projecte directament des de la pàgina del repositori usant el botó de “Clone or download”:



## Pas 5: Executar una app

La resta de passos de l'exercici 1 expliquen com executar i depurar una aplicació en Android Studio. No tenen preguntes avaluable però us serviran com a preparació per a realitzar la resta d'exercicis de la pràctica

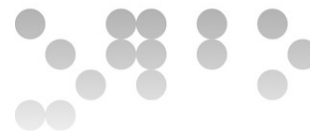
- a) En primer lloc, descarregar el codi del projecte proporcionat com a part de l'enunciat i obriu en Android Studio.



- b) Executeu-lo fent servir l'AVD que heu creat en el pas anterior.  
c) Per fer login a l'app, feu servir com a email [foo@uoc.edu](mailto:foo@uoc.edu) i com a password “1234”.

## Pas 6: Debug

Android Studio ofereix diverses funcionalitats per depurar una aplicació. Per llançar una aplicació en mode debug, executeu amb 




```
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_login);

if (ApplicationUtils.isLogged(getApplicationContext())) {
    goToNextActivity();
} else {

    // Set up the login form.
    mEmailView = (AutoCompleteTextView) findViewById(R.id.email);

    mPasswordView = (EditText) findViewById(R.id.password);
```

Podeu afegir punts d'interrupció (breakpoints) prement a la zona grisa al costat de la línia de codi on voleu col·locar el punt d'interrupció . L'aplicació s'executa normalment fins que arribi aquest punt concret del codi.

Per accedir a la consola de sortida de missatges (logcat) cliqueu a  **6: Android Monitor**

En logcat apareixen missatges escrits per l'App usant els comandaments de la classe Log. També apareixen missatges escrits pel sistema i per altres llibreries. Podem filtrar-los col·locant **UOC-PRA** a la zona de filtrat.

## Exercici 2. Programació concurrent (valoració d'un 20%)

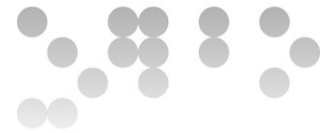
Generalment, una aplicació mòbil necessita realitzar diverses tasques de forma simultània. Per exemple, un gestor de correu electrònic pot estar comprovant si hi ha correus pendents al servidor mentre l'usuari redacta un nou missatge. Una característica molt desitjable és que la interfície d'usuari no quedi bloquejada mentre l'aplicació està realitzant altres tasques. Per aconseguir-ho, una pantalla o activity pot utilitzar diversos fils (threads) que s'executen de forma concurrent

Un ús molt habitual dels threads és l'accés a un servidor back-end sense bloquejar la interfície, per exemple, animant una barra de progrés (Progress). Moltes llibreries d'accés a back-end proporcionen funcionalitats que ens aïllen d'aquest tipus de programació, automatitzant part del procés. En aquest exercici anem a familiaritzar-nos amb la llibreria, que es proporciona en el codi, que simula l'accés de dades a un back-end de diverses formes :

- A la **MainActivity**, localitzeu el codi `// UOC-BEGIN-CODE1`.  
El fragment de codi inclòs entre els comentaris `// UOC - BEGIN - CODE1` y `// UOC - END - CODE1` presenta una programació similar a la que s'ha de realitzar amb els serveis de back-end com Firebase.
- Executeu l'aplicació en mode "Run". Observeu el Progress i el nom de les activitats.
- Executeu després en mode "Debug" col·locant abans un punt d'interrupció a la línia `Log.d(LibraryConstants.TAG, "Set breakpoint 1");`
- Observeu el contingut de logcat. Podeu filtrar només els missatges **UOC-TAG**.

**Pregunta 2.1.** Quins missatges apareixen en el logcat ?





**Pregunta 2.2.** Tots els Threads tenen els mateixos Ids? (Mirar logcat)

**Pregunta 2.3.** Es mou el Progress? Per qué?

- e) Observeu el contingut de logcat. Podeu filtrar només els missatges **UOC-PRA**.  
Comenta el codi que es troba entre `// UOC - BEGIN - CODE1` i `// UOC - END - CODE1` i descomenta el codi que es troba entre `// UOC - BEGIN - CODE2` i `// UOC - END - CODE2`.
- f) Col·loca un *breakpoint* a la línia **Log.d(LibraryConstants.TAG, "Set breakpoint 2")**;

**Pregunta 2.4.** Quins missatges apareixen en el logcat ?

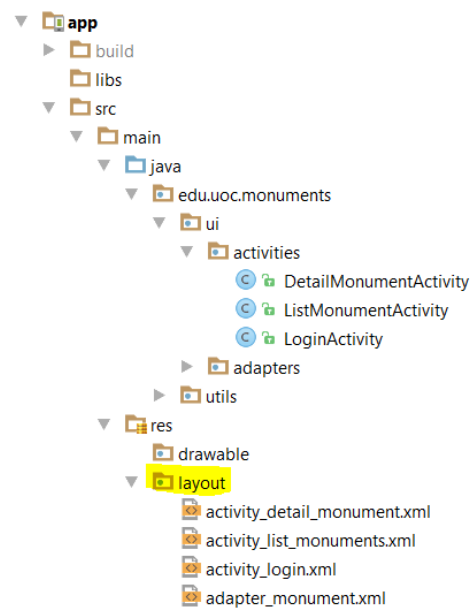
**Pregunta 2.5.** Tots els Threads tenen els mateixos Ids? (Mirar logcat)

**Pregunta 2.6.** Es mou el Progress? Per qué?

**Pregunta 2.7.** Conclusió: què està creant la llibreria UOC d'accés a dade en el cas del codi `UOC - BEGIN - CODE1`?

### Exercici 3. Modificació de layout (valoració d'un 30%)

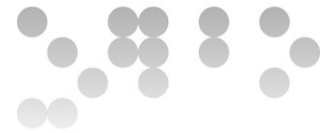
En aquest exercici anem a modificar els fitxers de layout amb els quals es controla l'aparença de les diverses pantalles d'una aplicació Android



L'activitat `ListMonumentActivity` dibuixa el llistat d'elements. Per fer-ho fa servir un tipus `MonumentAdapter` que implementa el mètode `getView ()`. Aquest mètode és cridat per la `ListView` per obtenir el dibuixat d'una fila. El primer que fa és utilitzar un `LayoutInflater` i usant el layout `adapter_monument` genera la vista..

Després en el codi es mostra com accedir a les dades i omplir la vista:





```
// Get monument object
Monument monument = monumentList.get(position);
// Set text with the item name
viewHolder.mName.setText(monument.getName());
// Set image
Bitmap image =
    LibraryManager.getInstance(context).getImage(monument.getImagePath());
viewHolder.mThumbnail.setImageBitmap(image);
```

**Pregunta 3.1.** Afegiu el país del monument tal com mostra la imatge de l'inici de la pràctica. Recordeu que heu de afegir-lo al layout adapter\_monument i carregar el seu contingut en el mètode getView(). Haureu definir la nova vista en la classe ViewHolder (dins del fitxer MonumentAdapter).

Pel que fa al layout volem que el país aparegui sota el nom i amb un menor mida de font. Quant a l'accés de dades, l'objecte Monument té un mètode que retorna el país de cada monument.

**Per a aquesta pregunta, adjuntar una captura de pantalla de la vista en el document a lliurar i el codi font.**

**Pregunta 3.2.** En l'activitat **ListMonument Activity** omplir el mètode onItemClick() per iniciar la activity DetailMonumentActivity. Se li ha de passar com a paràmetre l'id del monument seleccionat.

**Per aquesta pregunta cal lliurar el codi font utilitzat.**

#### **Exercici 4. Accés a dades i layout detallat (valoració d'un 30%)**

L'objectiu d'aquest apartat és implementar la vista detallada de producte de la nostra aplicació. Per això heu rebre un paràmetre en una activity, fer servir la llibreria de dades de la UOC, modificar el layout i carregar els elements d'aquest layout amb les dades de l'objecte obtinguts des de la llibreria de dades.

Al mètode onCreate() de l'activitat DetailMonumentActivity heu de realitzar les següents tasques:

**Pregunta 4.1.** Recupereu l'id rebut com a paràmetre a l'activitat.

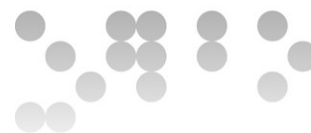
**Pregunta 4.2.** Modifiqueu el layout activity\_detail\_monument per afegir els camps descripció i imatge. Observeu el layout **adapter\_monument.xml** per veure com definir la vista ImageView.

**Pregunta 4.3.** Fent servir l'id que heu rebut, dins del mètode onSuccess(), es retornarà l'objecte Monument per poder accedir a les propietats de l'objecte.

```
LibraryManager.getInstance(getApplicationContext()).getMonumentById(id,
new GetCallback<Monument>() {
    @Override
    public void onSuccess(Monument monument) {

    }
    @Override
    public void onFailure(Throwable e) {

    }
});
```



**Pregunta 4.4.** Omplir el layout amb les propietats a les que has accedit al punt 4.3

*IMPORTANT:*

Per realitzar tot aquest exercici podeu repassar com actua el mètode `getView ()` de `result_activity`. Recordeu que podeu fer servir directament `findViewById ()` perquè s'accedirà al layout **layout\_activity\_detail\_monument** associat a aquesta activitat en el mètode **onCreate** (`setContentView(R.layout.activity_detail_monument);`) .

Per les preguntes 4.1 a 4.4, és precís adjuntar el codi font. A més a més, s'ha d'ajuntar una captura de pantalla de la vista al document a lliurar.

## Recursos

### Recursos Bàsics

- Mòdul didàctic 5. Desenvolupament d'aplicacions mòbils sobre Android.

### Recursos Complementaris

- [Desarrollo de aplicaciones para dispositivos Android](#)
- [Documentació oficial d'Android](#)

## Criteris de valoració

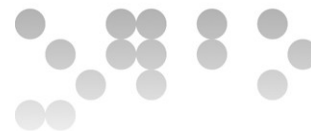
- La pràctica s'ha de resoldre de forma individual.
- Cada exercici té el següent pes:
  1. 20%
  2. 20%
  3. 30%
  4. 30%
- És necessari justificar la resposta de cada apartat. Es valorarà tant le resultat final com la justificació donada.

## Format i data de lliurament

S'ha d'entregar **un document PDF** amb les diferents respostes dels exercicis. El nom del fitxer ha de ser: **PRA1\_Cognom1Cognom2Nom.pdf**. Pels exercicis 3 i 4 és necessari enviar:

- .apk de l'aplicació.
- Tot el projecte empaquetat en un .zip (incloent el codi font).
- El codi font ha d'estar comentat en anglès.

Aquests fitxers s'han de lliurar a l'espai d'**Entrega i Registre d'AC** de l'aula **abans de les 23:59 del dia 15/11/2016**. **No s'acceptaran lliuraments fora de plaç**.



#### **Nota: Propietat intel·lectual**

Sovint és inevitable, en produir una aplicació mòbil, fer ús de recursos creats per terceres persones. És per tant comprensible fer-ho en el marc d'aquesta PAC, sempre i quan això es documenti clarament i no suposi un plagi.

Per tant, en presentar un lliurament que faci ús de recursos aliens (textos, imatges, icones, llibreries, fragments de codi,...), s'ha d'indicar clara i explícitament 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 utilitzada no impedeix el seu ús en aquesta PAC. En cas de no trobar la informació corresponent haurà d'assumir que l'obra està protegida pel copyright.