



Introduction to PIS: Integrated Software Project

Karim Lekadir, PhD

*Universitat de Barcelona
Dpt. Matemàtiques & Informàtica*



Objectives



1. Learn how to build an app using Android
2. Apply knowledge from past classes
3. Learn (a little bit) about the software development life-cycle
4. Learn how to work in a team
5. Learn how to learn (e.g. how to find information)
6. Gain experience on soft skills for software development



Objectives



1. Learn how to build an app using Android
2. Apply knowledge from past classes
3. Learn (a little bit) about the software development life-cycle
4. Learn how to work in a team
5. Learn how to learn (e.g. how to find information)
6. Gain experience on soft skills for software development

Android



- Linux-based (highly modular and secure)
- Open-source and free
- Easy to use (and to re-use code)
- Portable: Can be used in mobiles, tablets, watches, etc
- 85% of app share (2016)



Android Studio



MyApplication1 > app > src > main > AndroidManifest.xml

AndroidManifest.xml x MainActivity.java x activity_main.xml x

App (Manifest)

- manifests
 - AndroidManifest.xml
- java
 - com.example.myapplication
 - MainActivity
 - com.example.myapplication (androidTest)
 - com.example.myapplication (test)
- java (generated)
 - com.example.myapplication
- res
 - drawable
 - layout
 - activity_main.xml
 - mipmap
 - values
 - res (generated)
- Gradle Scripts

Code (Java)

Res (Layout, etc)

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3       package="com.example.myapplication">
4
5     <application
6         android:allowBackup="true"
7         android:icon="@mipmap/ic_launcher"
8         android:label="My Application"
9         android:roundIcon="@mipmap/ic_launcher_round"
10        android:supportRtl="true"
11        android:theme="@style/Theme.MyApplication">
12     <activity android:name=".MainActivity">
13         <intent-filter>
14             <action android:name="android.intent.action.MAIN" />
15             <category android:name="android.intent.category.LAUNCHER" />
16         </intent-filter>
17     </activity>
18 </application>
19
20 </manifest>
```

manifest

Text Merged Manifest

Android Studio

Gradle sync finished in 46 s 91 ms (12 minutes ago)

Event Log Layout Inspector

19:1 CRLF UTF-8 4 spaces

9:44 Hello World

Karim Lekadir



1. Manifest



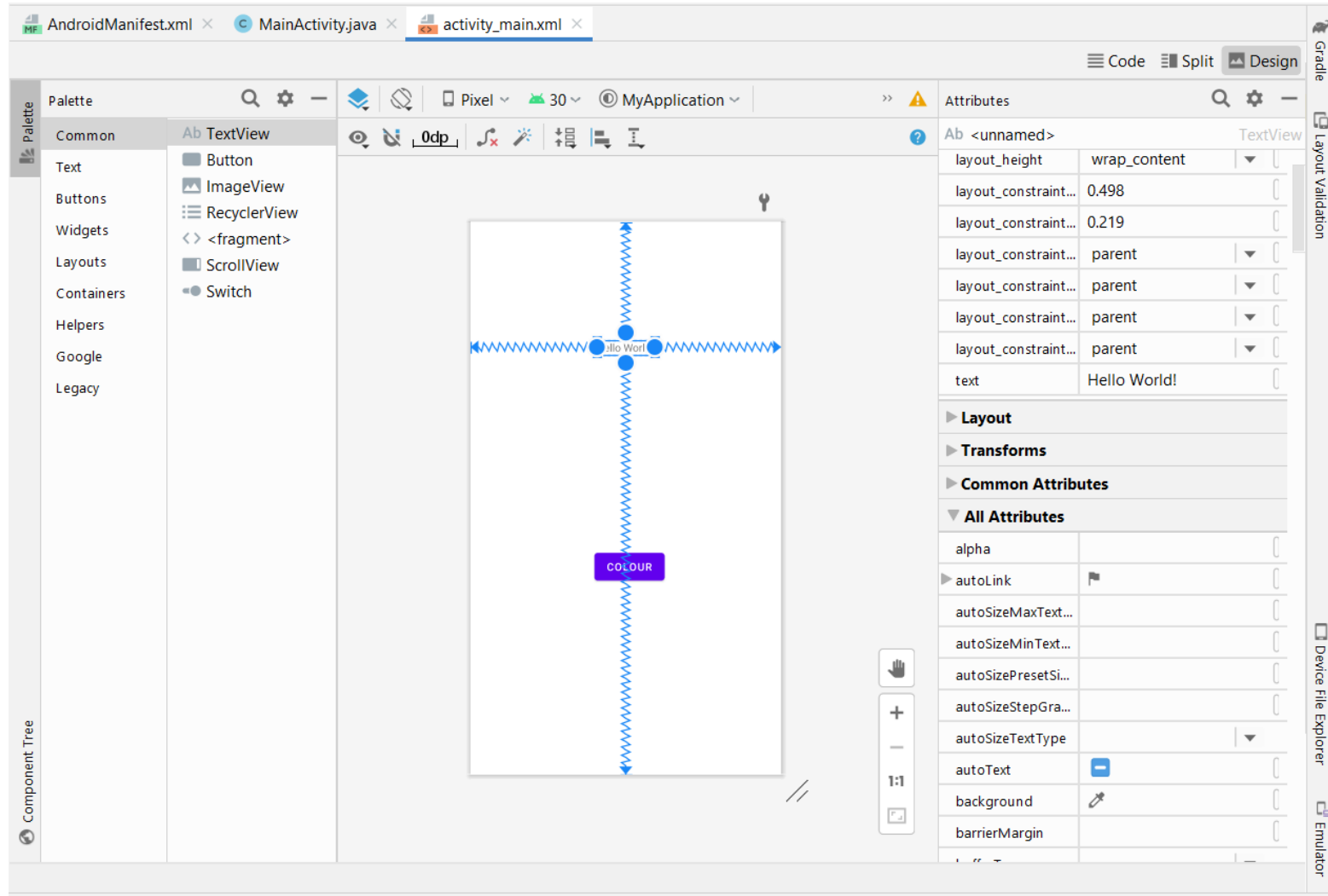
XML file containing all key information about the app:

- Name
- Main components
- Activities
- Permissions
- Software requirements
- Hardware requirements

```
AndroidManifest.xml x MainActivity.java x activity_main.xml x
1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3     package="com.example.myapplication">
4
5     <application
6         android:allowBackup="true"
7         android:icon="@mipmap/ic_launcher"
8         android:label="My Application"
9         android:roundIcon="@mipmap/ic_launcher_round"
10        android:supportsRtl="true"
11        android:theme="@style/Theme.MyApplication">
12        <activity android:name=".MainActivity">
13            <intent-filter>
14                <action android:name="android.intent.action.MAIN" />
15                <category android:name="android.intent.category.LAUNCHER" />
16            </intent-filter>
17        </activity>
18    </application>
19
20 </manifest>
```



2. Layout





3. Code



```
AndroidManifest.xml x MainActivity.java x activity_main.xml x
1 package com.example.myapplication;
2
3 import ...
11
12 public class MainActivity extends AppCompatActivity {
13
14     @Override
15     protected void onCreate(Bundle savedInstanceState) {
16         super.onCreate(savedInstanceState);
17         setContentView(R.layout.activity_main);
18         ConstraintLayout bgElement = (ConstraintLayout) findViewById(R.id.activity_main);
19         bgElement.setBackgroundColor(Color.RED);
20         myButtonListenerMethod();
21     }
22
23     public void myButtonListenerMethod() {
24         Button button = (Button) findViewById(R.id.button);
25         button.setOnClickListener(new View.OnClickListener() {
26             @Override
27             public void onClick(View v) {
28                 ConstraintLayout bgElement = (ConstraintLayout) findViewById(R.id.activity_main);
29                 int color = ((ColorDrawable) bgElement.getBackground()).getColor();
30                 if (color == Color.RED) {
31                     bgElement.setBackgroundColor(Color.BLUE);
32                     button.setText("BLUE");
33                 }
34                 else {
35                     bgElement.setBackgroundColor(Color.RED);
36                     button.setText("RED");
37                 }
38             }
39         });
40     }
41 }
```

R: Object containing all resource IDs

```
com.example.myapplication.R.id
public static final int activity_main = 1000134 ;
```

Functionalities (e.g. getColor())



Objectives



1. Learn how to build an app using Android
2. Apply knowledge from past classes
3. Learn (a little bit) about the software development life-cycle
4. Learn how to work in a team
5. Learn how to learn (e.g. how to find information)
6. Gain experience on soft skills for software development



Knowledge Application



Past classes	Future classes
Programming I	Databases
Programming II	Distributed computing
Algorithms	Artificial intelligence
Advanced algorithms	Computer vision
Data structures	Graphics and visualization
Software design (patterns)	Software engineering





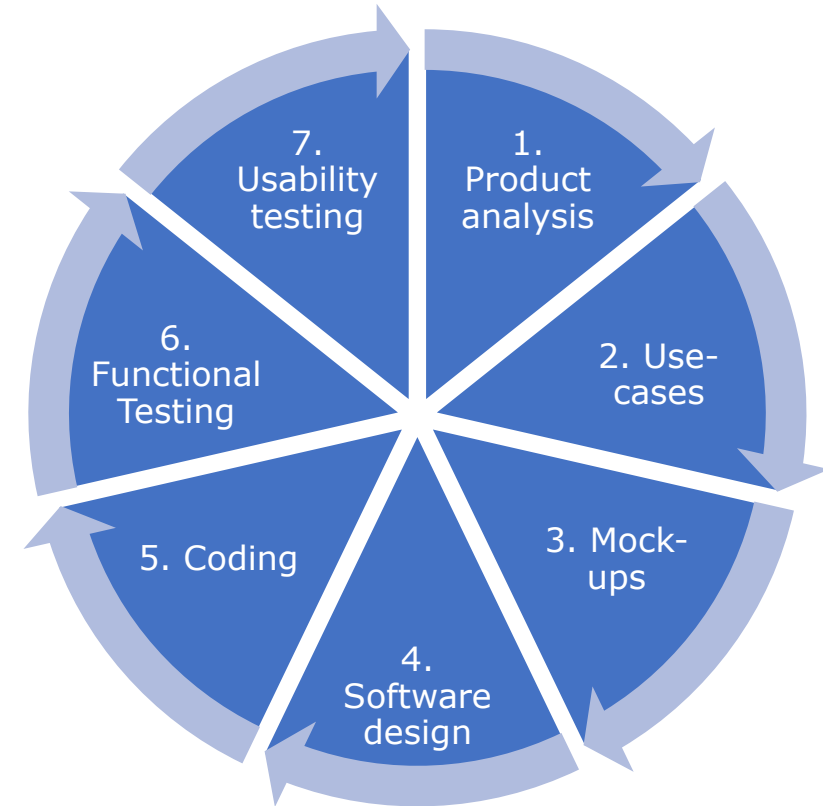
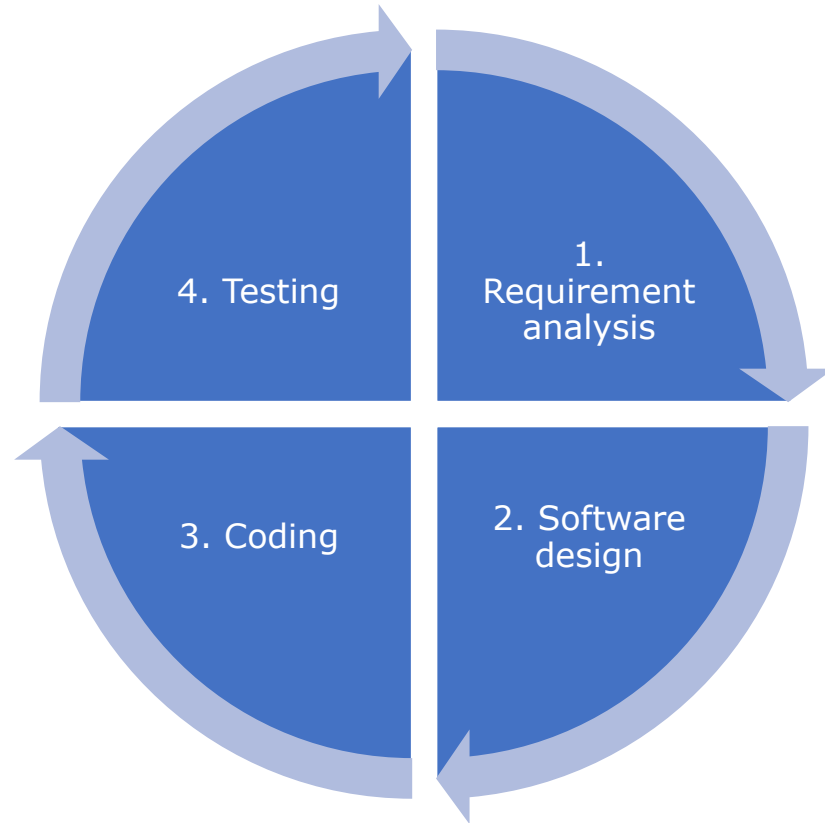
Objectives



1. Learn how to build an app using Android
2. Apply knowledge from past classes
3. Learn (a little bit) about the software development life-cycle
4. Learn how to work in a team
5. Learn how to learn (e.g. how to find information)
6. Gain experience on soft skills for software development



Development Life-Cycle





Objectives



1. Learn how to build an app using Android
2. Apply knowledge from past classes
3. Learn (a little bit) about the software development life-cycle
4. Learn how to work in a team
5. Learn how to learn (e.g. how to find information)
6. Gain experience on soft skills for software development

Team-Work



- Contribute to your team and help each others
- Define roles & responsibilities (incl. leader)
- Communicate and meet regularly (once a week?)



TE Scores & Team-Work



- Give a score (TE: Team Evaluation) to each team member on their contribution to the project (/10):
 - ✓ Member 1: 10/10
 - ✓ Member 2: 10/10
 - ✓ Member 3: 2/10
 - ✓ Member 4: 8/10
 - ✓ Lab teacher: 10/10
 - ✓ **TE (Median): 10/10**



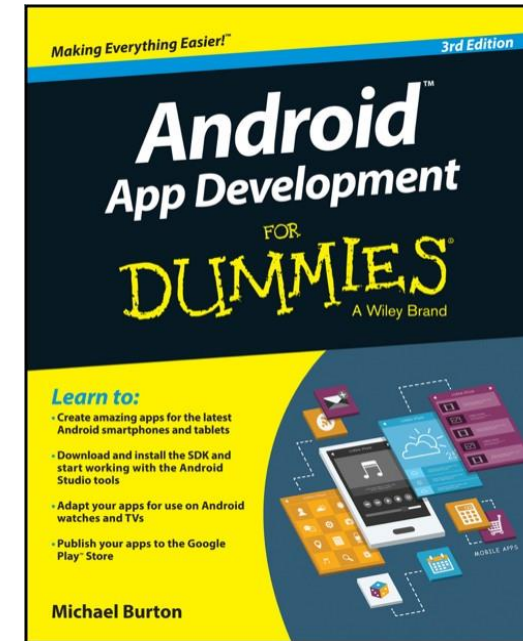
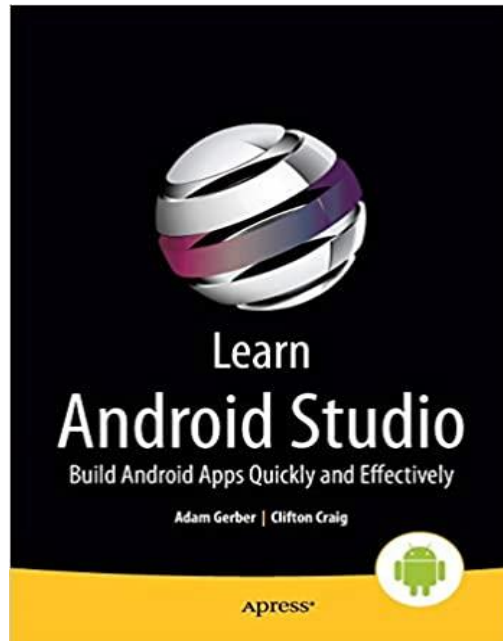
Objectives



1. Apply knowledge from past classes
2. Learn how to build an app using Android
3. Learn (a little bit) about the software development life-cycle
4. Learn how to work in a team
5. Learn how to learn (e.g. how to find information)
6. Gain experience on soft skills for software development

Finding Information

- Many topics in Android: UIs, text, sound, images, videos, calendars, GPS, maps, social media, external data, databases, wearables, sensors, etc, etc...





Finding Information



developers 

Platform

Android Studio

Google Play

Jetpack

Kotlin

Docs

News

 Search

ENGLISH ▾

[SIGN IN](#)

OVERVIEW

GUIDES

REFERENCE

SAMPLES

DESIGN & QUALITY

Google is committed to advancing racial equity for Black communities. [See how.](#)

TECHNOLOGY ▾

LANGUAGE ▾

RESET

15 30 60 per page

< 1 2 3 4 5 6 7 8 9 10 11 >

163 results

JAVA

FEATURED

Fragment Transitions: RecyclerView to ViewPager

This Android project accompanies the Continuous Shared Element Transitions: RecyclerView to ViewPager article. The code

JAVA

FEATURED

HdrViewfinder

This demo implements a real-time high-dynamic-range camera viewfinder, by alternating the sensor's exposure time

JAVA

FEATURED

BeamLargeFiles

This sample demonstrates how to transfer large files via Android Beam. After the initial handshake over NFC, file transfer will take

JAVA

FEATURED

BluetoothAdvertisements

Sample demonstrating how to advertise small amounts of data using the Bluetooth Low Energy API. Also demonstrates how to

JAVA

FEATURED

BluetoothLeGatt

This sample demonstrates how to use the Bluetooth LE Generic Attribute Profile (GATT) to transmit arbitrary data between devices.

JAVA

FEATURED

AutofillFramework

This sample demonstrates the use of the Autofill Framework. It includes implementations of client Activities with

JAVA

FEATURED

CommitContentSampleApp

This sample demonstrates how to write an application which accepts rich content (such as images) sent from a keyboard using the

JAVA

FEATURED

CommitContentSampleIME

This sample demonstrates how to write an keyboard which sends rich content (such as images) to text fields using the Commit



Objectives



1. Apply knowledge from past classes
2. Learn how to build an app using Android
3. Learn (a little bit) about the software development life-cycle
4. Learn how to work in a team
5. Learn how to learn (e.g. how to find information)
6. Gain experience on soft skills for software development

Soft Skills



Communication, presentations (e.g. demos), documentation, problem-solving, critical thinking, project management, etc



<https://www.youtube.com/watch?v=MnrJzXM7a6o>



Deliverables



Only three deliverables for the whole course:

1. Project team proposal/requirements + User Interface (UI)
2. Software design + demo
3. Final project + presentation



Continuous Evaluation



- A. Project team proposal + User Interface (UI)
- B. Software design + demo
- C. Final project + presentation
- D. 30-minute exam (multiple choice questions)

$$\text{Final mark} = (A*0.25 + B*0.25 + C*0.25)*TE + D*0.25$$

TE = Team Evaluation, will be normalised between 0.5 and 1



Unique Evaluation



- A. Project team proposal + User Interface (UI)
- B. Software design + demo
- C. Final project + presentation
- D. 30-minute exam (multiple choice questions) + 2.5 hours coding

$$\text{Final mark} = (A*0.20 + B*0.20 + C*0.20)*TE + D*0.40$$

TE = Team Evaluation, will be normalised between 0.5 and 1



Teachers



Theory
(One hour
per week)



Karim Lekadir

Labs
(2.5 hours
per week)



Carlos Martin



Victor Campello



Calendar



Teoricopràctica [Presencial]

M1	dl.	dt.	dc.	dj.	dv.	2n sem.	11.00-12.00	Lekadir , Karim	Aula B5	Castellà
-----------	-----	-----	-----	-----	-----	---------	-------------	---------------------------------	-------------------------	----------

Pràctiques de laboratori d'ordinadors [Presencial]

a00	dl.	dt.	dc.	dj.	dv.	2n sem.	08.30-11.00	Martín Isla, Carlos	Aula ID	Català
b00	dl.	dt.	dc.	dj.	dv.	2n sem.	08.30-11.00	Martín Isla, Carlos	Aula IG	
c00	dl.	dt.	dc.	dj.	dv.	2n sem.	08.30-11.00	Campello Román, Víctor Manuel	Aula IG	
f00	dl.	dt.	dc.	dj.	dv.	2n sem.	08.30-11.00	Campello Román, Víctor Manuel	Aula ID	

Exàmens : 1r parcial [Presencial]

G1	5 d'abril de 2022.	18.00-21.00	Lekadir , Karim Campello Román, Víctor Manuel Martín Isla, Carlos		-
-----------	--------------------	-------------	---	--	---

Exàmens : Final [Presencial]

G1	2 de juny de 2022.	15.00-18.00	Lekadir , Karim Campello Román, Víctor Manuel Martín Isla, Carlos		-
-----------	--------------------	-------------	---	--	---

Exàmens : Reavaluació [Presencial]

G1	22 de juny de 2022.	18.30-21.30	Lekadir , Karim Campello Román, Víctor Manuel Martín Isla, Carlos		-
-----------	---------------------	-------------	---	--	---



Calendar



Week		Theory	Labs	Deliverables
1	16 February	Introduction to PIS	Introduction to Android	
2	23 February	Design pattern 1	Session 1	
3	2 March	Requirements	Session 2	
4	9 March	Examples requirements	Session	
5	16 March	Feedback deliverable 1	Feedback & support	
6	23 March	Design pattern 2	Feedback & support	1. Requirements + UI
7	30 March	Software testing	Feedback & support	
8	6 April	Week partials (not for PIS)	Feedback & support	
9	13 April	<i>Semana Santa</i>	Feedback & support	
10	20 April	Feedback deliverable 2	Feedback & support	
11	27 April	<i>Matefest-Infofest (no lectiu)</i>	Feedback & support	2. Design + demo
12	4 May	<i>Fira d'Empreses</i>	Feedback & support	
13	11 May	Feedback deliverable 3	Feedback & support	
14	18 May	Trial exam	Feedback & support	
15	25 May	Trial exam	Presentations	3. Final project

Project 2022



How to choose a project:

- Something you like
- Something you can do
- Something that will allow you to learn
- BUT: keep it as simple as possible





Project 2022



Note-taking App integrating multiple options:

- Text
- Photo
- Sound
- Link (e.g. Facebook event)
- Drawing
- Calendar
- File
- Etc



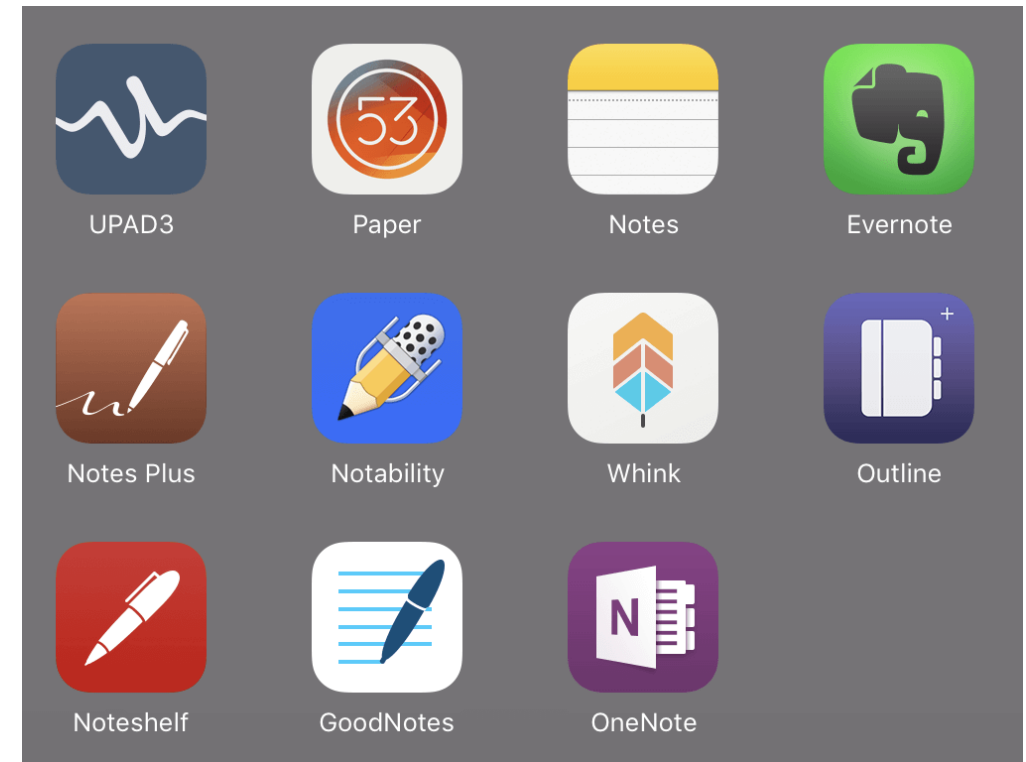


Project 2022



Note-taking App integrating multiple functionalities:

- Add
- Edit
- Remove
- Share
- See statistics
- Add to calendar
- Mark as important
- Etc





Project 2022



Can be domain-specific:

- Personal
- Group (e.g. family, friends)
- Work
- Sports
- Cultural events
- Political
- News
- Etc...





Next...



1. Implement App examples in the labs
2. Set up groups: About 4 members
3. Start working on your project proposal
4. Next week we will start with Deliverable 1 of the project:
Requirements & Mock-ups



Question 1



Manifest.XML in Android are used to specify:

- A. The main objects and classes.
- B. The main files in the project.
- C. The main properties and components of the App.
- D. The main layout of the App.



Question 2



Which of these lifecycles of software development is incorrect?

- A. Product analysis, requirements, design, coding, testing.
- B. Mock-ups, use-cases, design, coding, testing.
- C. Requirements, design, coding, functional testing, usability testing.
- D. Requirements, design, coding, functional testing.



Question 3



A software team member :

- A. Must have only one role
- B. Can have maximum two roles
- C. Must be involved in all roles
- D. Should be involved in all software development phases