An evaluation of modern Android Libraries and Frameworks

TFM-Desenvolupament d'Aplicacions en Dispositius Mòbils

Josep Rodríguez López - joseprl89@uoc.edu

Agenda

1. Introduction
2. App demo
3. Development methodology
4. Studied frameworks

Introduction

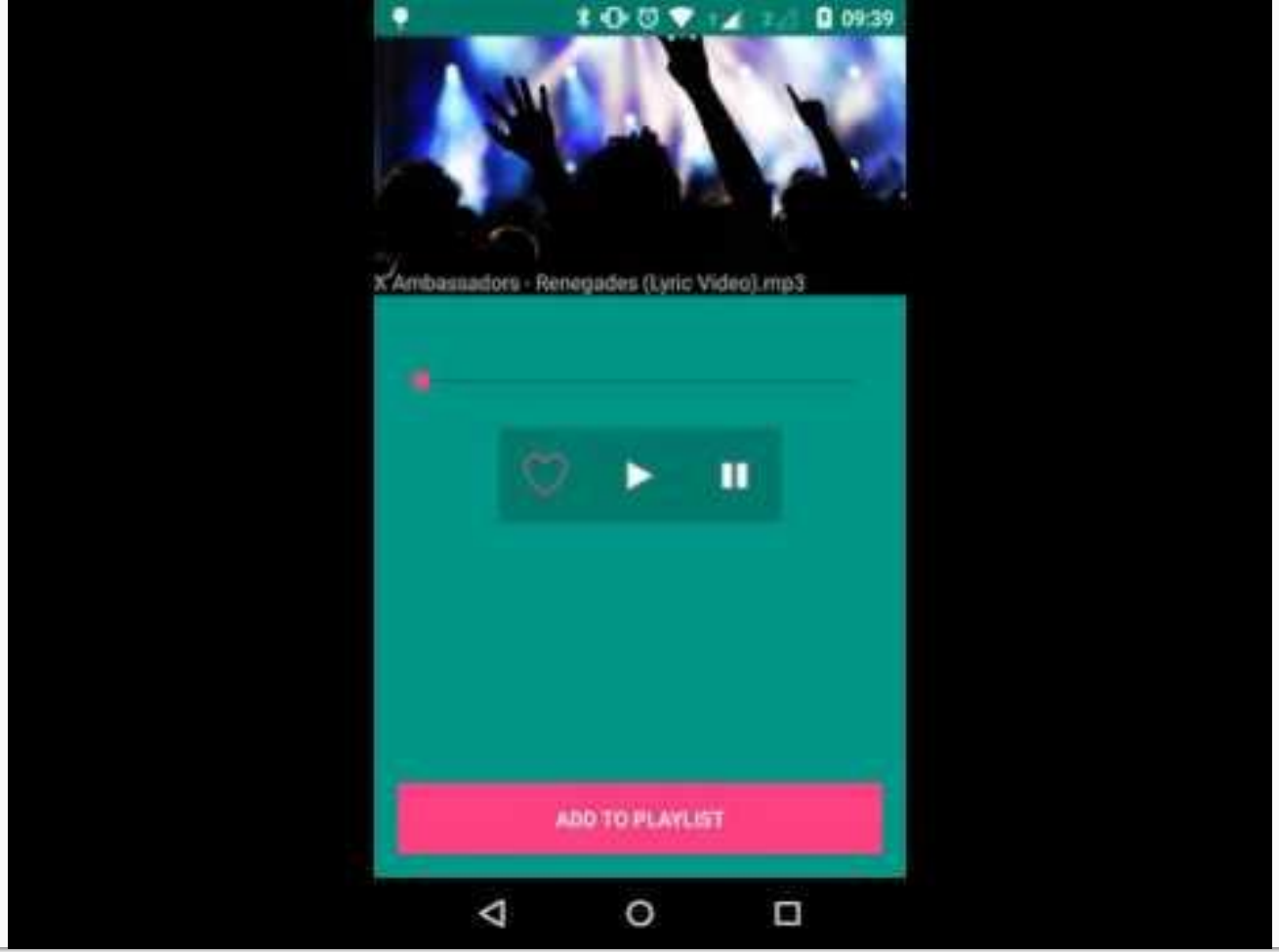
Introduction

- Motivation
- Goals
- Approach

Frameworks studied

- Dagger2
- Mosby
- RxAndroid
- Realm
- Android support design libraries
- Butterknife
- Retrofit
- Glide

App demo


09:39
X Ambassadors - Renegades (Lyric Video).mp3
ADD TO PLAYLIST

App demo

Development methodology

Development methodology

- SDLC
- Testing
- Release process

Software development lifecycle

- Kanban
- User stories
- Git flow

Development methodology

- SDLC
- Testing
- Release process

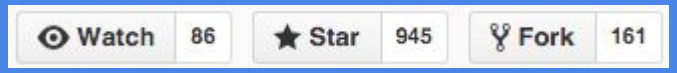
- Test script written in Gherkin
- Feature testing
- End to End testing

Development methodology

- SDLC
- Testing
- Release process

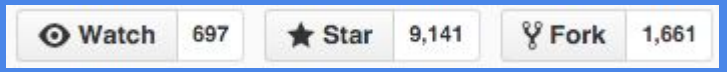
Studied frameworks

Mosby



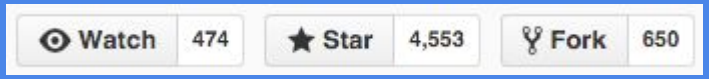
- Provides a framework to enforce an MVP architecture on your app.
- Heavylifts some of the difficulties of handling rotation in activities.
- Splits an activity/fragment into:
  - Interface for the view
  - Interface for the presenter
  - Implementation for the presenter
  - Activity implementing the view interface

Retrofit



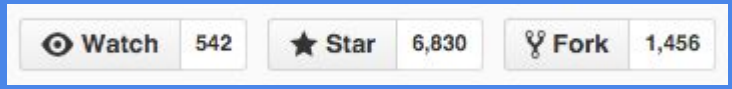
- Converts an annotated interface into a fully functional REST client.
- Removes a LOT of boilerplate.
- Highly customisable
  - Sometimes it’s difficult to tell where to customise for a specific purpose.
- Supports RxAndroid

RxAndroid



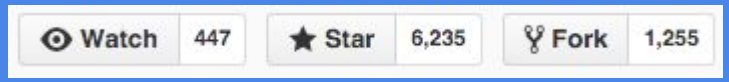
- Provides a stream of events to perform modifications functionally on
- Plethora of operations
- Simplifies complex tasks
- Plenty of support
  - documentation
  - several ports to other platforms/languages
  - RxMarbles

Butterknife



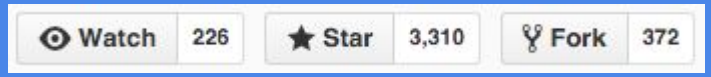
- Removes a lot of cumbersome boilerplate code
  - @Bind
  - @OnClick, @OnLongClick,...
  - Butterknife.bind
- Does not support all listeners Android provides
  - OnSeekBarChangeListener

Glide



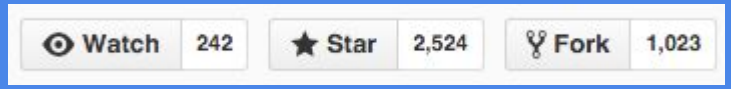
- Eases the process to load URL’s into image views
- Has a really nice declarative API
  - Glide.withContext(...)
  - .loadImage(...)
  - .withPlaceholder(...)
  - .intoImageView(...)
- Also caches the images downloaded

Realm



- DBMS for Android & iOS
- Nice API
- Small limitations
  - Models must inherit Realm class
  - Your class is overridden on runtime

Dagger2



- Works by defining a component and module to provide the dependencies.
- Not as “magic” as expected.
- Injecting dependencies via initialisers is simpler unless the class has several dependencies.
- Components can grow big and it can be non-trivial how to split them.

Android support design libraries

- Helps a lot on bringing the Material design look and feel to the app.
- This functionality should be on the Android SDK.

Conclusions

Conclusions

- Objectives met
  - Inadequate evaluation of Dagger2
- Future work

Thanks.

Josep Rodríguez López