

# Effective TDD on Android with Kotlin



Olmo Gallegos Hernández



@voghDev



# Overview

- **Very simple Android App using TDD**
- **Develop in cycles**
- **Start each cycle with a (Failing) Test**



# Source of data

<http://chucknorris.io> API

# Source of data

<http://chucknorris.io> API





# The recipe

- Very simple architecture based on MVP
- `kotlinx.coroutines`
- `arrow-kt`

# Example project

<https://github.com/voghDev/ChuckNorrisJokes>





# Use cases

- Get Random Joke
- Search Jokes by keyword
- Get Jokes by category
- Get categories

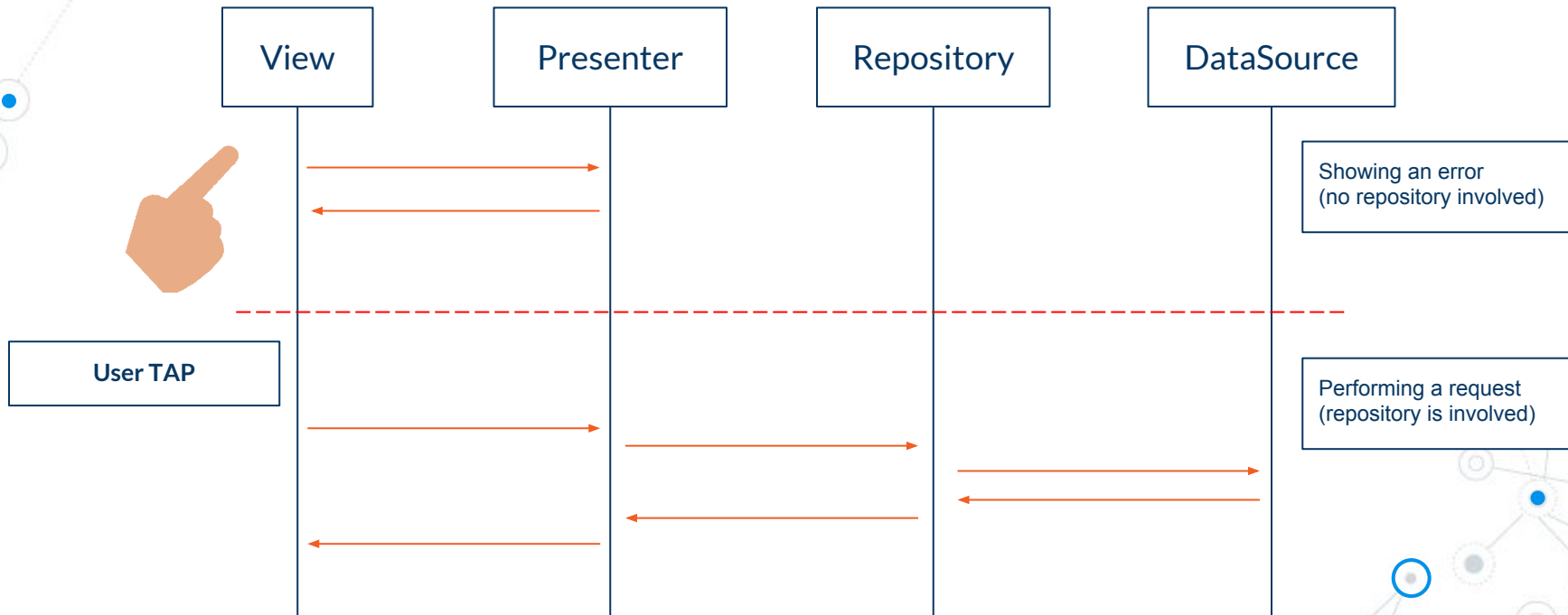
# #1: Get Random Joke

- 1.- Request a Joke from the Api
- 2.- Render its text
- 3.- Render its image





# The Architecture - Android MVP



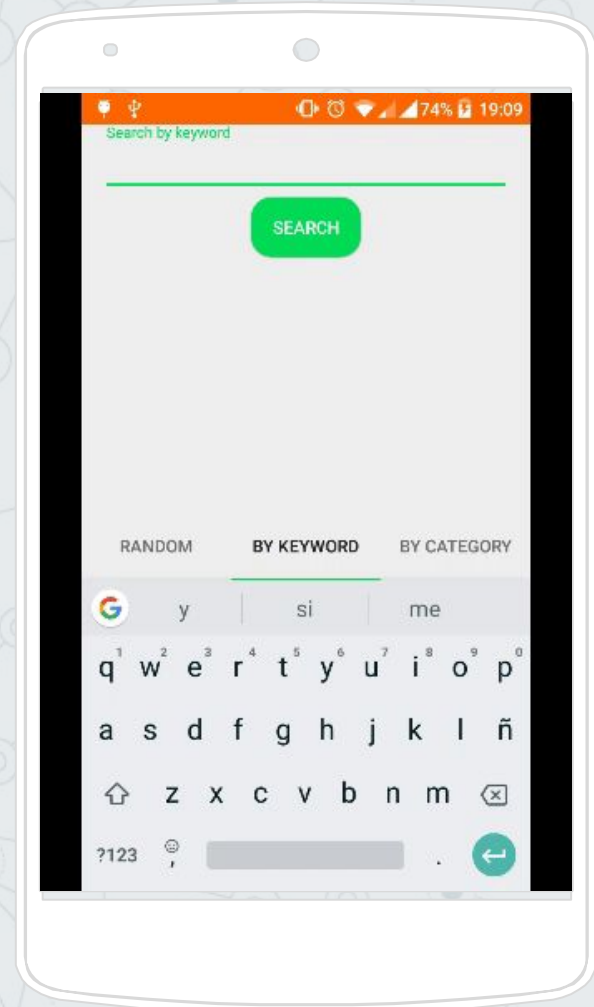


# Interesting stuff while it compiles

- Kent Beck's "two modes of implementation" to get the green
- Martin Fowler's "Passive View"
- We're doing many assumptions in this approach
- Production apps following this guidelines
- Alternate executions (all Tests -vs- local Tests)

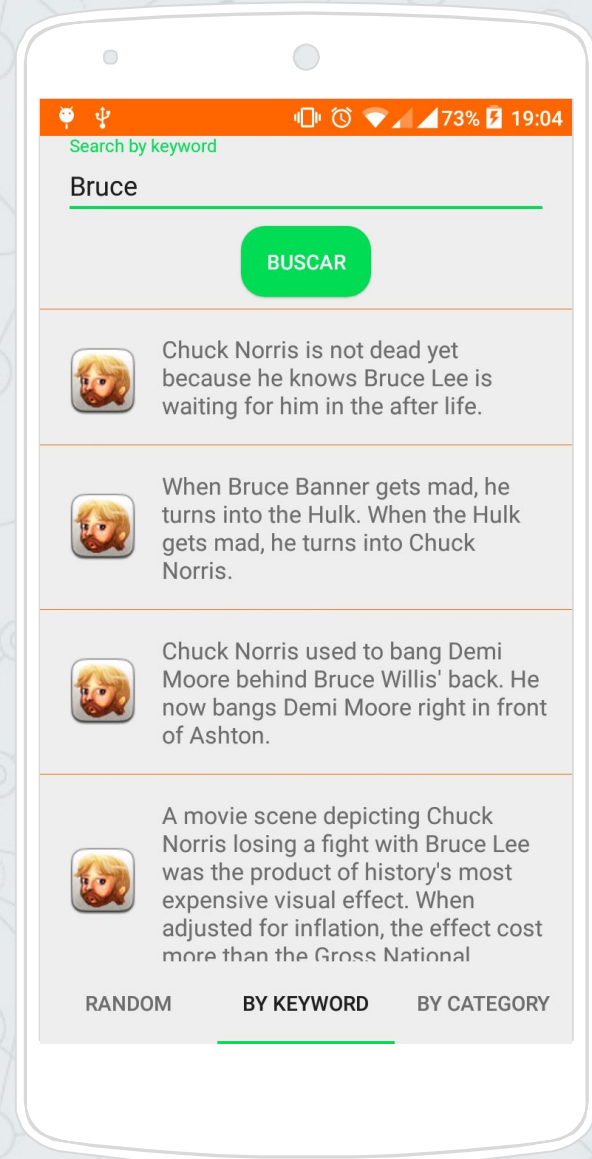
## #2: Search by keyword

- 1.- Performing the search
- 2.- Dealing with wrong values
- 3.- Showing the results
- 4.- Dealing with the empty case
- 5.- Refactoring



## #2: Search by keyword

- 1.- Performing the search
- 2.- Dealing with wrong values
- 3.- Showing the results
- 4.- Dealing with the empty case
- 5.- Refactoring





# Credits

- **TDD by Example (Kent Beck)** - [https://books.google.es/books/about.html?id=gFgnde\\_vwMAC](https://books.google.es/books/about.html?id=gFgnde_vwMAC)
- **arrow-kt** - <http://arrow-kt.io>
- **kotlinx.coroutines** - <https://kotlinlang.org/docs/reference/coroutines.html>
- **Bandook-Kotlin** - <https://github.com/antoniolg/Bandhook-Kotlin>
- **OpenLibraryApp** - <https://github.com/jrgonzalezg/OpenLibraryApp>

# Questions & Answers

<https://github.com/voghDev/ChuckNorrisJokes>



voghDev



@voghDev



Olmo Gallegos Hernández



# Thanks for coming!



voghDev



@voghDev



**Olmo Gallegos Hernández**