
GitHub Username: **acasadoquijada**

Podcastfy

Description

Do you like listening to podcast? There are several podcast application out there. Don't you know which one you should download in order to avoid missing any interesting podcast?

If that is the case, this is your lucky day! With Podcastfy you will have access to podcast hosted in different pages, such as Spotify and Ivoox. And this is only the beginning!

Without further due, here are the thing that Podcastfy is capable of:

- Find podcasts about any kind of theme and genre in different podcast providers such as Spotify and Ivoox.
- Learn, explore and discover new topics, technologies, mean you are enjoying a walk, reading or simply lying in your sofa!
- Listen, download and share podcast with your friends
- Create a list with your favorite podcast to not miss any episode
- Keep you in touch with what is happening in the world

Intended User

The intended user for Podcastfy is anyone that wants to have a great time, learn new themes, have fun, or simply relax.

Features

- Search podcast in different podcasting services
    - Spotify: Using a specific library to interact with the web api
    - Ivoox: Parsing the web page using a specific library
    - See Key consideration for more info about these libraries
- Reproduce podcast
    - When the user leaves the app, the podcast will be controlled by a notification
- Download podcast episodes to listen them offline
- Create custom play list with favorite podcast
- Share play list, podcast and podcast episodes with friends

User Interface Mocks (Work in progress)

These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Google Drawings, www.ninjamock.com, Paper by 53, Photoshop or Balsamiq.

Screen 1: MainActivity

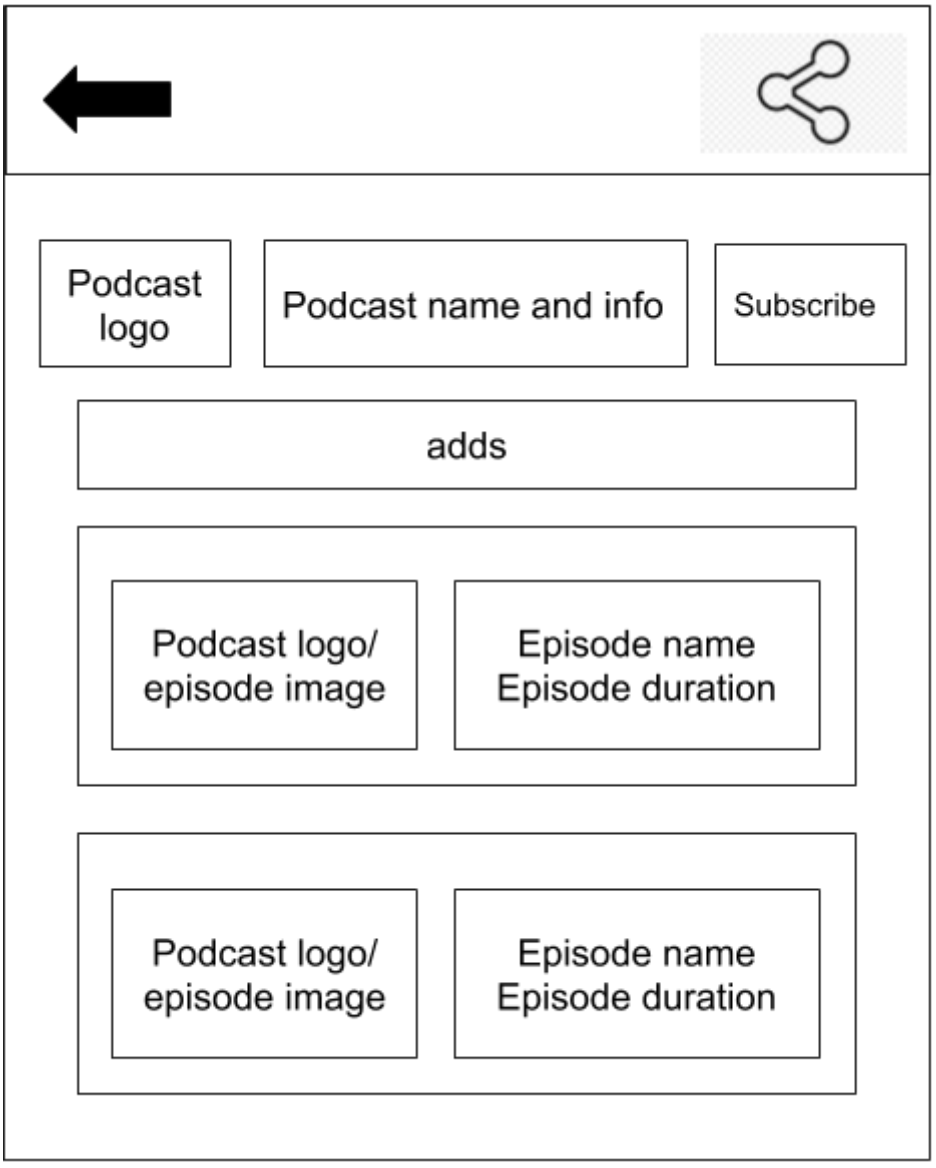
There will be only one activity in this application. The Activity is composed by two principal elements:

- A fragment which will be changed according to the user actions
- Bottom menu with 4 different items, each one represents a fragment. This fragments will be described in the following sections:
    - A -> PodcastListFragment
    - B -> SearchFragment
    - C -> PodcastSubscriptionFragment
    - D -> ConfigFragment

- The initial fragment is A, PodcastListFragment (is represented in the Mokcup) i presents to the user different podcast:
    - Subscribed to (if not subscribed, nothing is shown)
    - Podcasts of each differente webpage/provider. For now:
        - Ivoox
        - Spotify

Once a podcast is clicked. This fragment is changed by PodcastFragment

Screen 2: PodcastFragment

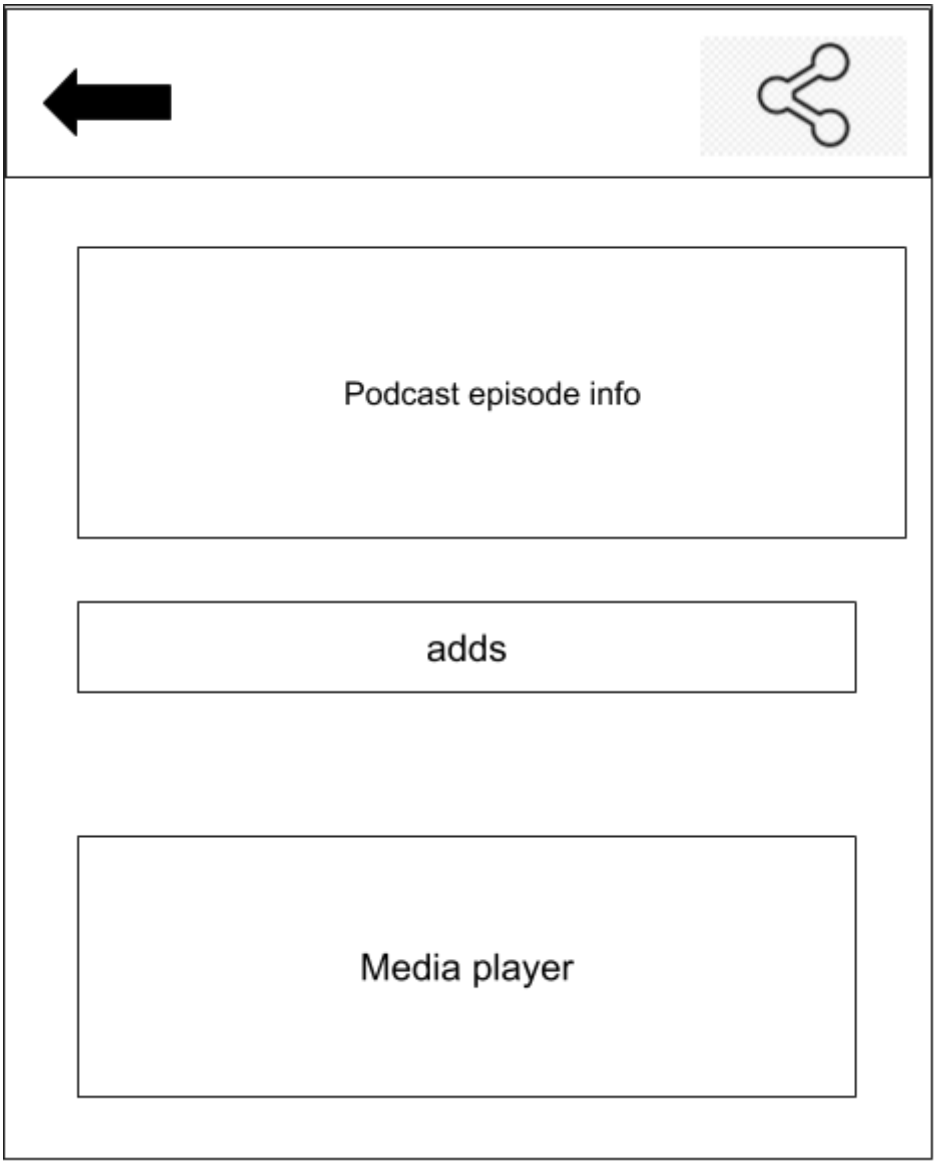


The AppBar allows to the user to go to the previous fragment (PodcasListFragment) and share the podcast with friends.

This fragment presents to the user information about the podcast and a list of available episodes.

When an episode is clicked, the PodcastEpisodeFragment is launched.

Screen 3: PocastEpisodeFragment



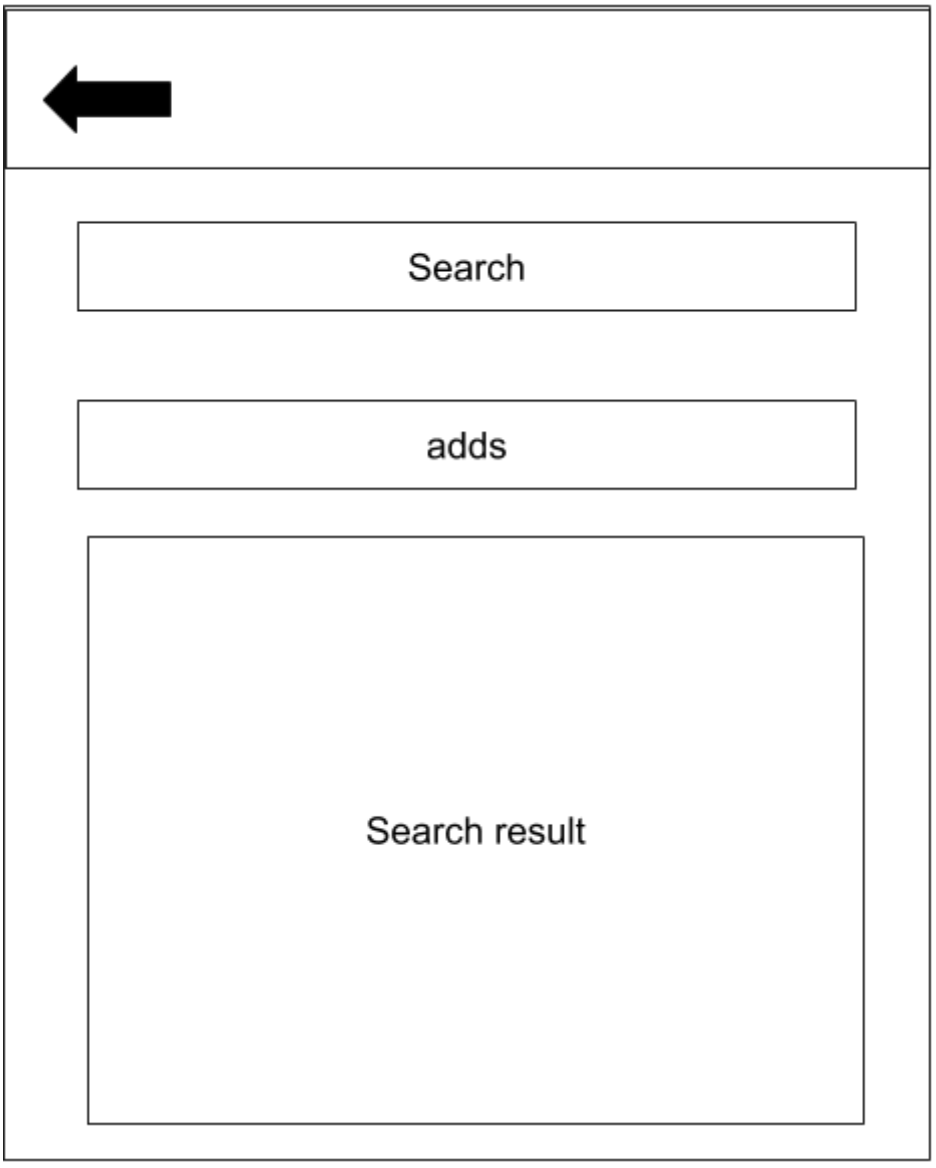
The AppBar allows the user to go to the previous fragment (PocastFragment) and share the episode with frriends.

This fragment presents information about the episode along with a media player. When the user leaves the fragment, a media player will keep reproducing the current episode above the bottom navigation menu of MainActivity.

Besides this, a notification will be created to interact with the media player.

In addition to this, the user will be able to download the episode to reproduce it offline

Screen 4: SearchFragment



Allows to the user to perform Podcast searching, then it presents the results below. Clicking on a podcast will open a PodcastFragment.

Screen 5: SubsribedPodcastFragment

Presents to the user the podcast they are subscribed to. Clicking on a podcast launches PodcastFragment.

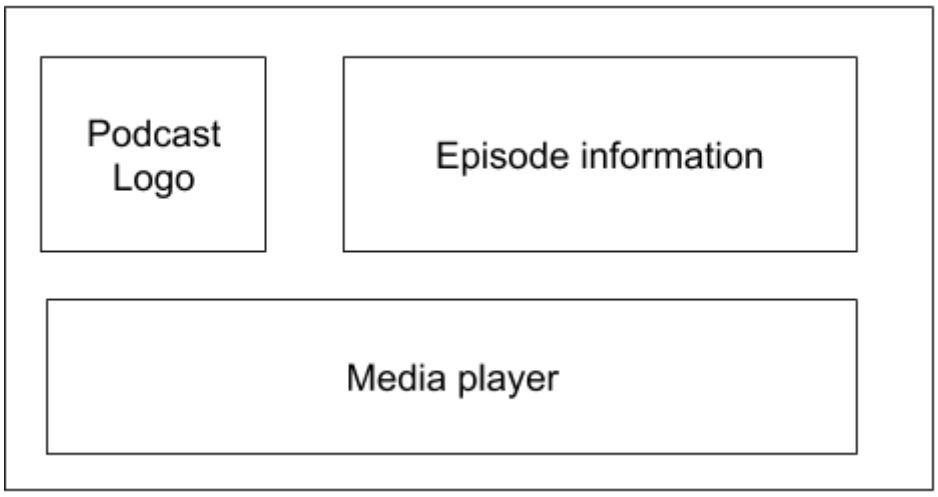
Screen 6: SettingsFragment

In this fragment the user will be able to:

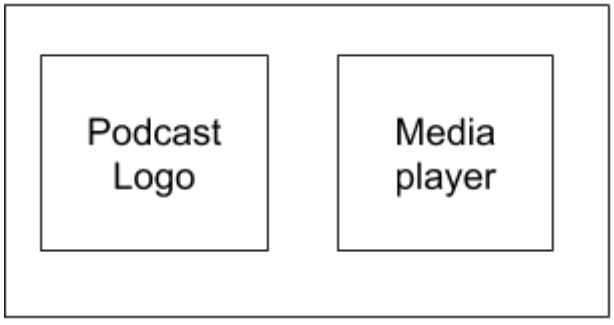
- Select podcast provider
    - Ivoox and/or Spotify. At least one
- Check playlist
    - For this PodcastListFragment will be used
- See episode history
    - For this PodcastListFragment will be used
- Manage saved episodes
    - Reproduce them
    - Delete them
    - For this PodcastListFragment will be used
- Clean all the data stored in the device

Screen 7: Widget

Regular Widget



Small widget



If the user has enough space in screen the widget will have the layout described in *Regular Widget.* Otherwise, *Small Widget* will be displayed.

The idea of the Widget (independently of its size) is to interact with the current episode that is reproducing.

Key Considerations

How will your app handle data persistence?

This is the data that the app will store and how it will do it:

- Podcast provider

    - Spotify or Ivoox
    - This will be a setting
- Podcast episodes
    - Audio files like mp3
    - This files will be stored as mp3 (or corresponding format) in the application folder within the device
- Podcast information
    - This is information such as podcast image, name, description, url..
    - For this one Room will be used
- Historical information
    - Episodes reproduced by the user along with timestamp
    - For this one Room will be used
- PlayList
    - Podcast episodes information (url, title, description)
    - For this one Room will be used

Describe any edge or corner cases in the UX.

If the user is in PodcastEpisodeFragment and click back at the Appbar of in the phone bottom, the audio will still be reproducing above of the bottom menu of the MainActivity. Besides this, a notification will appear, allowing the user to interact with the episode

Describe any libraries you'll be using and share your reasoning for including them.

- Picasso to handle the loading and caching of images
- Spotify Web API for Android for interacting with the Spotify web api
    - First I wanted to use Spotify Web API Java as is more recent, but in the README file there is a note for Android Developers to use Spotify web API for Android due to an issue
- Jsoup to perform web scrapping and interact with the Ivoox webpage
    - The web scrapping will be performed as par of the Task 4 of development
- Drager to perform dependency injection as we are going to follow MVVM as described here

Describe how you will implement Google Play Services or other external services.

- Admob to display some test adds in the different layouts. Interstitial will be shown too.
- Analytics to identify where the users spend more time in the application. Useful as feedback to improve and add features.

Next Steps: Required Tasks

Development pattern followed:

- **Test Drive Development**. By doing so, we ensure any functionality added is tested and working as expected. This will be helpful when migrating from fake data to actual data.

- **Model View ViewModel** (MVVM) as explained here

Task 0: Project Setup

- Set up the libraries
- Check that all dependencies are a stable release versions
- Check Android Studio version is the last one stable available

Task 1: Create model classes for Podcast, Podcast Episode, PlayList and Episode history

- This needs to be done in early development stage as is going to be used from the beginning using fake data and eventually moving to actual data.

Task 2: Add Room for data persistence

- Add the necessary Room structure and classes to store the models of Task 1

Task 3: Implement UI and logic for Each Activity and Fragment

- In this step of the development:
    - The data will be fake. This will change in future tasks
    - The adds will be imageViews. This will change in future tasks

- Build UI for MainActivity
    - Implement fragment change using the bottom menu
    - Create PodcastListFragment
- Build UI for PodcastFragment
    - Present podcast info to the user
    - Implement share podcast functionality
    - Create needed files to presents the podcast episodes to the user
- Build UI for PodcastEpsiodeFragment
    - Present the podcast episode info to the user
    - Implement share functionality
    - Use exoplayer to reproduce podcast episodes

- Build UI for SearchFragment
    - Implement search functionality
    - Present results to the user
- Build UI for SettingsFragment
    - Prepare this functionallitites
        - Select podcast provider
        - Check playlist
        - See episode history
        - Manage saved episodes
        - Clean all the data stored in the device
- Build widget
    - Implement both layouts, Regular and Small widget

Task 4: Make API for Ivoox

- Use jsoup to obtain podcasts and podcast episodes
- Build the elements needed for the storing podcast in the DB
- Modify needed activities to use data from Ivoox instead of fake one

Task 5: Use Spotify Web API for Android to get podcast from Spotify

- Use the spotify web api to get podcast and episodes
- Build the elements needed for storing podcast in the device
- Modify needed activities to use data from Ivoox instead of fake one

Task 6: Setup Google Play services

- Add admob to the different layouts
    - At this point all the UI are finished and working as expected
    - There will be only adds at the bottom of the screen, no
- Add analytics to each Activity and Fragment

Submission Instructions

- After you've completed all the sections, download this document as a PDF [File → Download as PDF]
    - Make sure the PDF is named "**Capstone_Stage1.pdf**"
- Submit the PDF as a zip or in a GitHub project repo using the project submission portal

If using GitHub:

- Create a new GitHub repo for the capstone. Name it "**Capstone Project**"
- Add this document to your repo. Make sure it's named "**Capstone_Stage1.pdf**"