GitHub Username: coastalgit

Application Title

Portu-go

Description

Portu-go is an English to Portuguese language learning aid, focusing specifically on verb usage.

It can be used as a visual and audio reference to the complexity of Portuguese verbs and pronunciation, in conjunction with the use of examples and quiz challenges.

Intended User

For anyone learning Portuguese, as a hobby or travelling.

Features

Main features of app:

- Support for real-time language updates from (Firebase) cloud storage.
- Local storage (Room database) of language sets.
- Saves previous quiz scores.
- Text-to-speech (TTS) support.

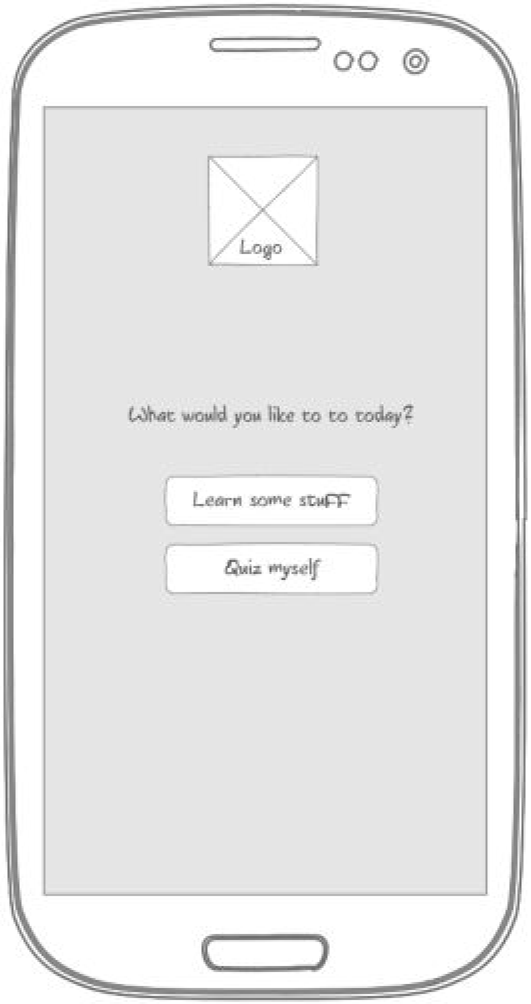
App also conforms to base Udacity rubrics:

- Written solely in Java.
- Integrates a third party library.
- Only use of stable releases of all libraries, Gradle and Android Studio.
- Integrates two Google services.
- Applies Material Design with standard transitions.
- Equipped with signing config and keystore details.
- Provision of app widget for home screen use.
- Inherent design of English to Portuguese negates need for RTL layouts.
- All resource strings to be stored in *strings.xml* resources.
- App has consistent theme defined in *styles.xml* and does not impact usability.
- All high quality icons and images will be stored as local resource with no immediate requirement for any image loading libraries such as Glide or Picasso.
- Applied typeface fonts will be stored within app assets.
- App validates all input (Quiz feature).

User Interface Mocks

Created with NinjaMock.

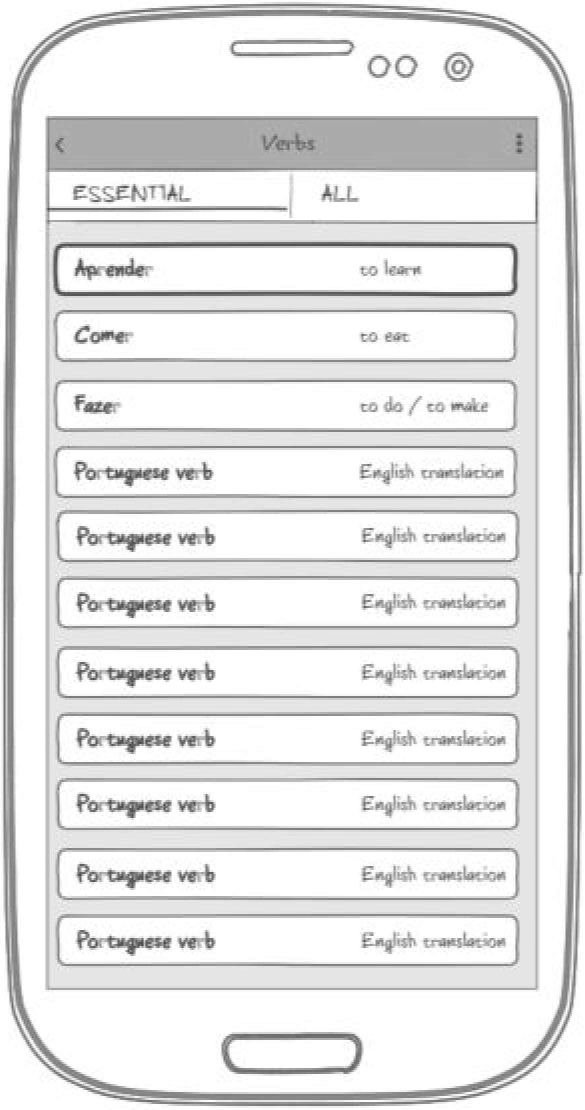
Screen 1 - Home



What would you like to to today?

Home screen allowing selection to LEARN or to QUIZ. Code behind will also be responsible for checking network connectivity, and if local device database is populated.

Screen 2 - (Learning) Verb Listing

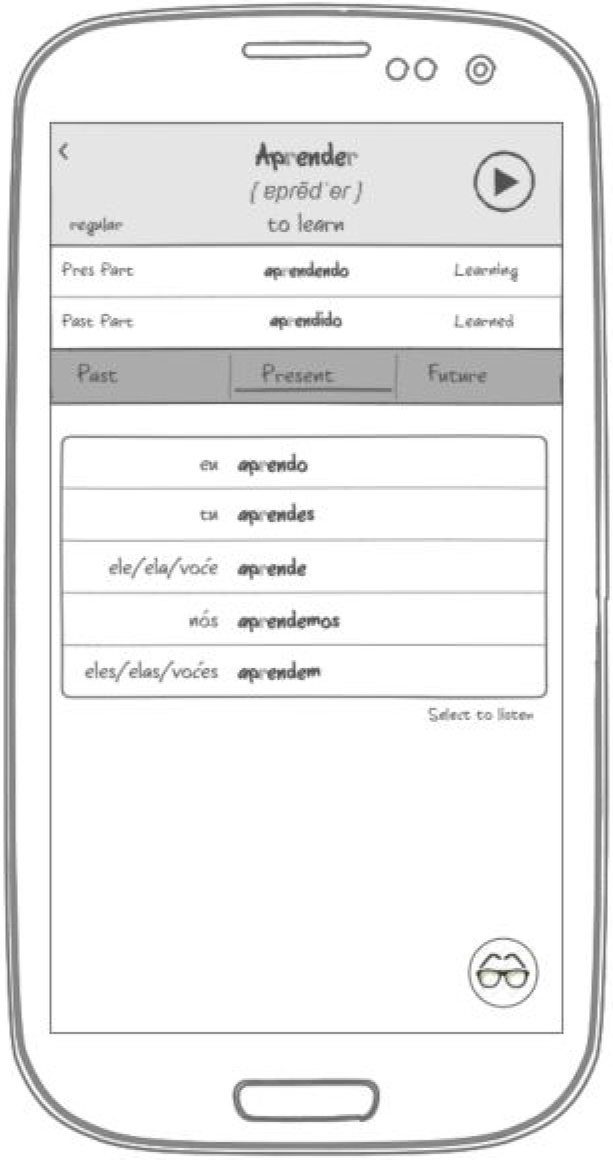


Aprender	to learn
Comer	to eat
Fazer	to do / to make
Portuguese verb	English translation
Portuguese verb	English translation
Portuguese verb	English translation
Portuguese verb	English translation
Portuguese verb	English translation
Portuguese verb	English translation
Portuguese verb	English translation
Portuguese verb	English translation

Learning screen, allowing view (via tab bar) of essential (most common) verbs or complete listing.

Selection of any verb (view holder) will load the VERB DETAIL screen.

Screen 3 - (Learning) Verb Detail



Pres Part	aprendendo	Learning
Past Part	aprendido	Learned

eu	aprendo
tu	aprendes
ele/ela/você	aprende
nós	aprendemos
eles/elas/vocês	aprendem

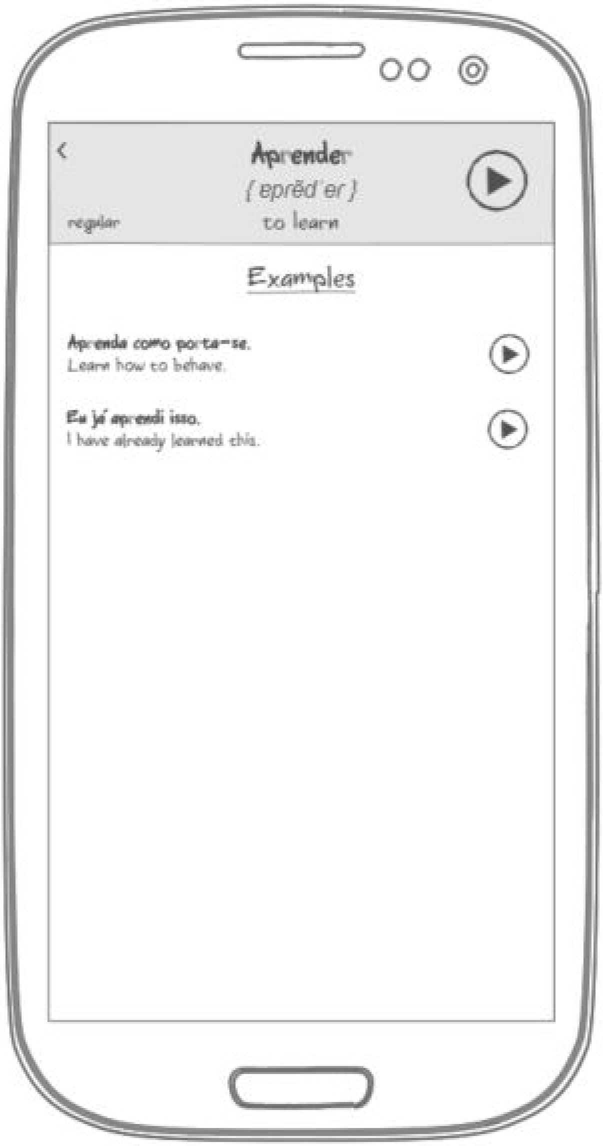
Learning screen, allowing view of conjugations of selected verb.

Allows text-to-speech (TTS) playback on individual item clicks and icon.

Tab bar also allows selection of most common tenses.

Floating action button loads example usage screen.

Screen 4 - (Learning) Verb Examples



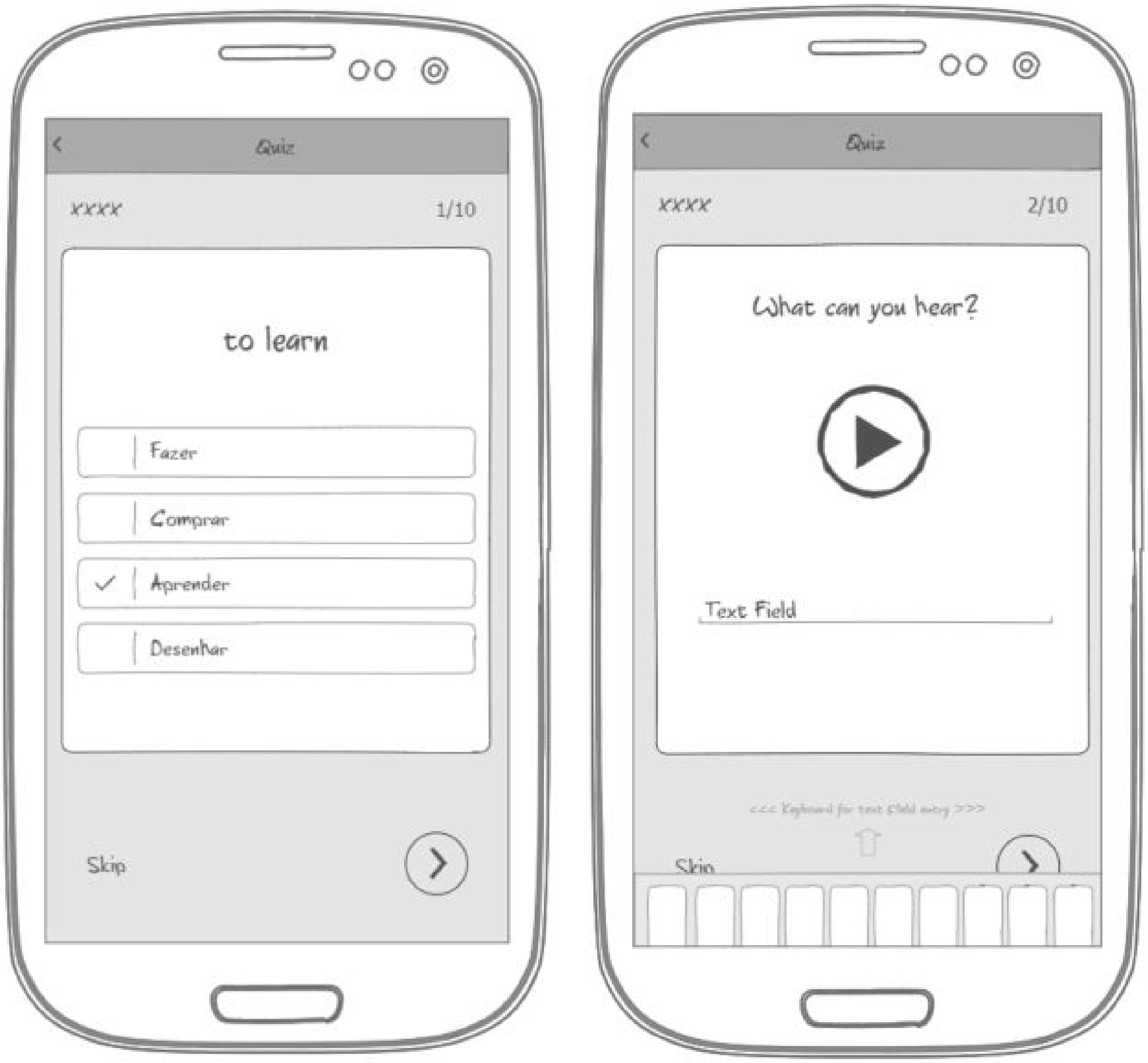
Aprenda como porta-se.
Learn how to behave.

Eu já aprendi isso.
I have already learned this.

Learning screen, allowing view of some examples of verb in use.

Allows text-to-speech (TTS) playback on individual item clicks and icon.

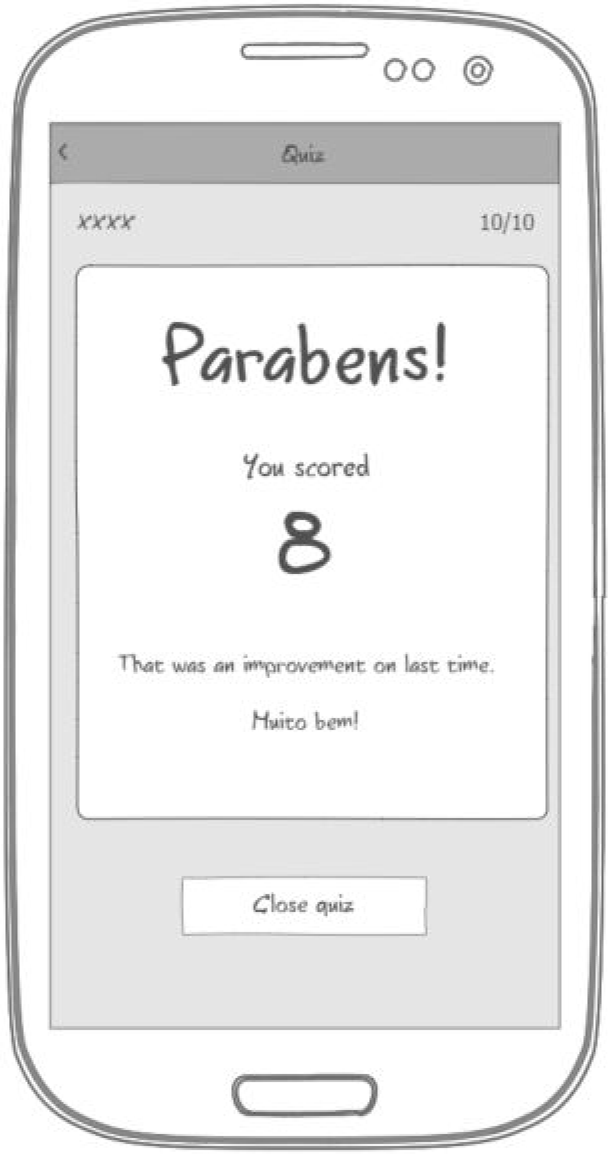
Screen 5 - (Quiz) Sequential pages



Quiz screens, with assorted types (visual / audio) of quiz, taking score/tally as user progresses.

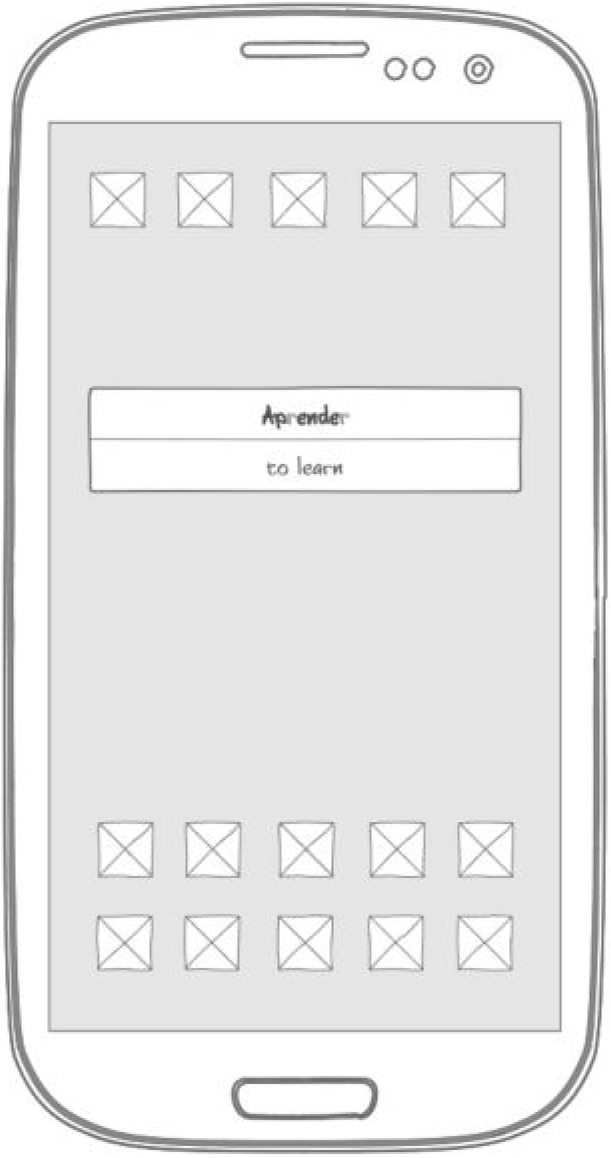
Applies text-to-speech (TTS) playback on audio question.

Screen 5 - (Quiz) Completion



Quiz finished screen, with score and comparison against previous saved scores.

Screen 6 - Widget



Home screen widget with changing displayed verb, also launches app.

Key Considerations

How will your app handle data persistence?

App stock data (all verb listings) will be stored in Firebase real-time database and synchronised to a local device Room database (on app start up and via implicit Firebase observer).

Describe any edge or corner cases in the UX.

Offline Android Portuguese TTS engine is required. App must detect if installed and launch installation screen, and also make an allowance for no TTS availability.

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

Android Architecture Components for ViewModels, LiveData, LifeCycle and Room.
Admob for displaying of adverts during app usage.
Butterknife for easier view binding.

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

The app will use Google Ads and Firebase

Next Steps: Required Tasks

This is the section where you can take the main features of your app (declared above) and break them down into tangible technical tasks that you can complete one at a time until you have a finished app.

Task 1: Project Setup

Write out the steps you will take to setup and/or configure this project. See previous implementation guides for an example.

You may want to list the subtasks. For example:

- Familiarise with all intended libraries.
- Configure normalised Gradle setup, such as using references for version numbers, specifying build variants etc.

- Set up Firebase project via console.

If it helps, imagine you are describing these tasks to a friend who wants to follow along and build this app with you.

Task 2: Creation of models and storage mechanisms.

Creation of models for use across Firebase, Room database records and stored preferences / run time data.

Also implement AsyncTask within DAO for all Room transactions.

Task 3: Implement UI for Each Activity and Fragment

- Build UI and ViewModel (VM) for Home Activity
- Build UI (VM) for Learning List Activity
- Build UI (VM) for Learning Detail Main Activity
- Build UI for Verb Detail and Verb Examples Fragments.
- Build UI for Quiz Main Activity, Questions Fragments & Completion screen.

Task 4: Synch verb data from cloud to device

Via implementation of MVVM and LiveData observables, the app must download all verb categories (and also observe for real-time server data updates) from Firebase, and store to persistent Room database on the device.

Task 5: Detect TTS support

In the absence of the ability to use beta libraries for this project, such as Google Cloud TTS API, I must handle the detection, and subsequent options to install the native Android Portuguese TTS engine for offline audio TTS playback.

Task 6: Implementation of Ads

Ads to be implemented on Verb Detail screen (AdView banner) and during quiz sequence (Interstitial full screen)

Task 7: Creation of home screen widget.

Creation of a home screen widget, displaying a chosen verb and some associated detail.

Task 8: Create relevant test harness.

Creation of mock data, unit tests and Espresso tests for UI.

Provisional Libraries and Environment

Environment

Android Studio (Windows)	v3.13
Gradle version	v4.4
Android plugin version	v3.1.3
Java target compatibility	v1.8

Libraries / Dependencies

Android Support	v27.1.1
Android Architecture Components - Room	v1.1.1
Firebase Database	v11.8.0
Firebase Ads	v11.8.0
Butterknife	v8.8.1
Junit	v4.12
Espresso	v3.02

[end]