Rodrigo Javier Garrido Dagle

RodrigoJavierGD@gmail.com
+56 9 9609 5384

https://linkedin.com/in/rodrigo-javier-gd/
https://rodrigojavier101.github.io/portfolio/

SUMMARY

I feel comfortable in a teamwork where I can contribute with ideas, so I can be autonomous but connected with the team.

I have a few years working and improving on my specialization in **Android and Kotlin Multiplatform** mobile development**.**

EDUCATION AND CERTIFICATIONS

- ***Android bootcamp badge***, Talento Digital, *https://www.credly.com/badges/cc6bc067-4834-468d-9fc2-27c4ea2955ed?source=linked_in_profile*
- ***English upper intermediate level*** *(B2)*: *https://cert.efset.org/pNEGCW*
- **Facilitator certificate** for E-learning technological bootcamp.

PROFESSIONAL SKILLS

- ***JDK*** tool.
- ***Programming languages and Scripting***: kotlin, java, javascript, bash, sql.
- ***Design Patterns:*** builder, singleton, adapter, observer, interceptor, factory.
- Familiarity with Agile/Scrum management with ***JIRA***.
- Applications published into the Play Store. Familiarity with the process.
- ***SOLID*** principles and ***Clean Code*** rules. Abstraction and useful algorithms with use cases.

- Software architecture for android development I use ***MVVM***, but I also know about ***MVC***, ***MVP*** and ***MVI. Hexagonal architecture*** as well.
- ***Unit and Integration Testing*** with Junit, Espresso, Mockito, UI testing**.**
- Automated testing / Continuous Integration and delivery (ci-cd) with ***GitHub actions*** and ***Jenkins***.
- ***Asynchronous work:*** concurrency blockers, coroutines, ***LiveData***, ***Flows***, ***RxJava.***
- ***Background tasks:*** BroadcastReceiver, WorkManager, implicit broadcast exceptions, Battery optimization, Foreground Services, Tasks Scheduling, Sequential background tasks, Network timeout, Connection cancel request, Job Scheduler, etc.
- ***Network with Retrofit and okHttp,*** use of custom interceptors for headers appwide.
- ***Services.***
- ***Debugging in Android Studio,*** Identifying and correcting bottlenecks to fix bugs**.**
- ***Intents.***
- ***Push Notifications.***
- ***Permissions handling.***
- ***Authentication,*** caching strategy, reminders.
- ***Gradle*** with groovy and kotlin, using ***Version Catalog*** as well.
- ***Version control system (VCS)*** with Git; repositories in **GitHub** and **Bitbucket**.
- ***Integrated development environment***: Android Studio, Vim, Visual Studio Code, IntelliJ IDEA.
- ***Functional*** an ***Object-oriented*** programming.
- ***Dependency injection*** with dagger hilt.
- ***Persistence:*** ROOM and RealmDB.
- Proficient UI programming skills: ***Jetpack Compose***: Pagination, dark/light screens themes, dialogs, collapsing toolbar, swiping images, clickable/draggable spinner, keyframe animation, animated vector drawable, shape shifter, support for multiple screens, etc.
- ***XML*** for ***UI*** experience, navigation graph.
- Experience with commonly used Android components and libraries.

- ***Activities and Fragments, lifecycle*** management.
- **Crashlytics**.
- ***Firebase, Supabase.***
- ***FRONTEND-BACKEND*** skills: ***Docker***, ***Databases*** like PostgreSQL, MySQL, MongoDB (in this order), **Node/Express**, ***ReactJS***, cloud admin with ***AWS and Digital Ocean***.
- Creating and maintaining comprehensive **documentation**.

PROJECTS

"Electric vehicle charger"

Insights:

Is a clone application from my dependent work in ***Dhemax*** which is a IT company located in ***Chile*** (***https://www.dhemax.com/en***) where I was a junior mobile developer between **April to August in 2023**. Using ***Jira*** as a scrum job admin.
The mobile team size were three mobile developers (2 android and 1 IOS) plus one Quality assurance tester, one UI/UX designer a backend team and one Product Manager. So, I'm experienced collaborating with designers and UX experts.

***Project*:**

This app performs a useful use case when an electric car owner needs a public charger. It indicates the progression in the charge, connectors available (also its positions in real time), the record of previous charges, and all the information related with the customer charges.

***Tasks performed*:**

It works with a mocked backend from Firestore which response all the data.
Is essential the data flow immediacy because, the use case is that the customer charging his car and we need to give him the status of it.

***Environment*:**

Important use of StateFlows and Coroutines for async work. State management. Also, I use google maps API to locate the charging points.

Source code: *https://github.com/jardindunoix/chargerapp*

Play Store Name*: "ELECTRIC CAR CHARGER CLONE"*

"Mercado Libre global seller administrator"

Insights:

I worked for BL Trading since **April 2020** to **April 2023**. There I developed the system that integrates Mercado Libre information API.

***Project*:**

Is an integration with the Mercado Libre API, it uses a mocked backend from ***https://sellers-bltrading.cl*** system (developed by me as well) is the in-production system which shows the processes and results of stock movements from various global sellers.

***Tasks performed*:**

Build de backend for the database transactions just to make possible the information visible in the application.

***Environment*:**

In the original project uses a cloud from ***Digital Ocean/AWS*** cloud server where is the PostgreSQL database and dockerized all the backend functionalities built for the application deployment data, but in the mobile mock version uses Firestore to mock the backend.

Source code: *https://github.com/jardindunoix/global-seller-admin*

Play Store Name*: "GLOBAL SELLER ADMINISTRATOR"*

“Monitor Político”

Project:

It is a backend-agnostic project that uses web scraping to get the data and retrieves the congressmen information.

***Tasks performed*:**

I maintain the endpoints updated and the web scraping structure in the software.
I use persistence with ***Room*** which is used immediate after every network call to keep the immediacy of the UI response.

***Environment*:**

Kotlin for mobile development and ***JSoup*** for web scraping.
Source Code: *https://github.com/jardindunoix/monitor_politico*
Play Store Name: *“MONITOR DE LA POLITICA CHILENA”*

“Scheduler for Common Debts”

Project:

Organizes the accounts in a group of people with related debts. The idea is to keep the common debts synchronized in every participant app and so everybody knows how updated everyone is.

***Tasks performed*:**

Communication among independent apps, synchronizing the data too keep all devices in the same level of information about their individual payments. ***Supabase*** as a backend support.

***Environment*:**

XML UI performance, navigation graph.
Source Code: *https://github.com/jardindunoix/job_schedule*
Play Store Name: *“SCHEDULER FOR COMMON DEBTS”*

REFERENCES

Require them in the contact resources links.