

# Zaida Mascorro Valdés

(+52) 474 110 1900 | Guadalajara, Jalisco, México



<https://github.com/ZaidaZadkiel>



<https://www.linkedin.com/in/z-mascorro-0b16b81b0/>



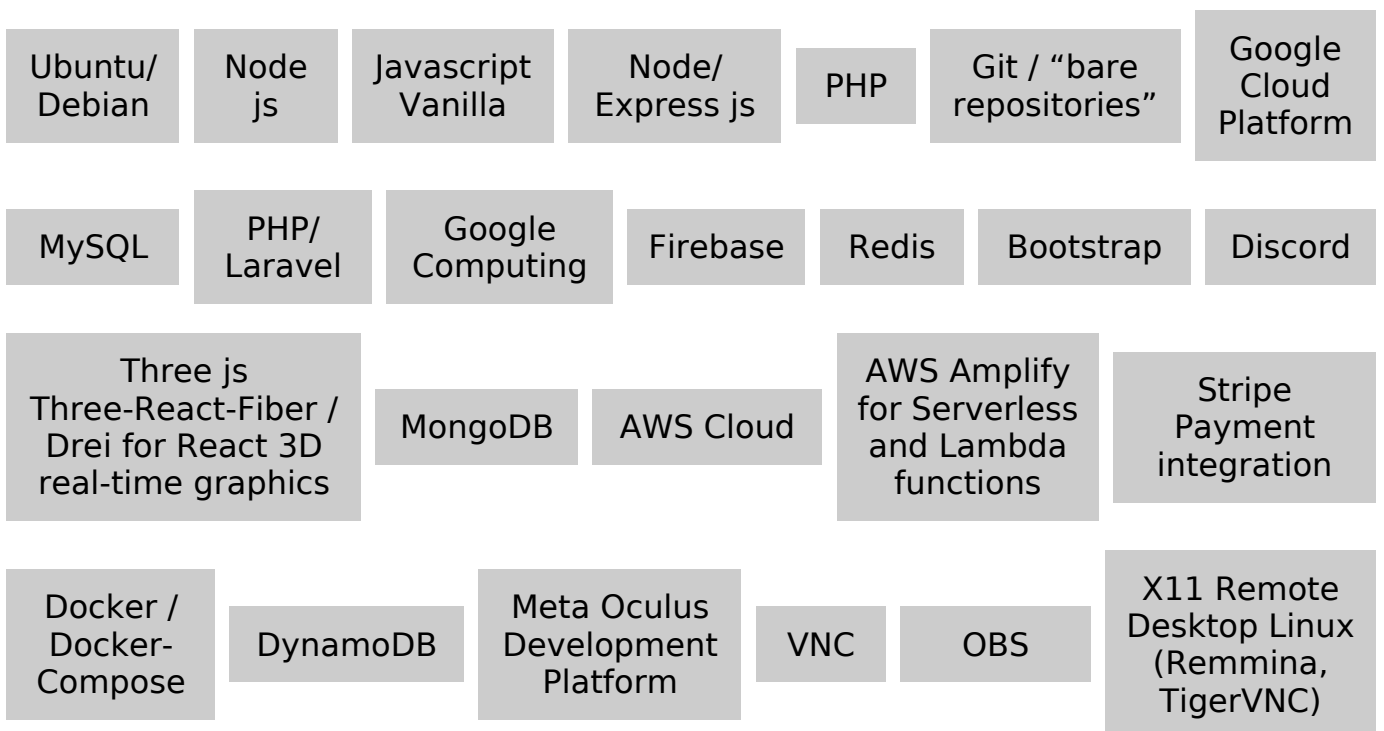
[angel.zadkiel@gmail.com](mailto:angel.zadkiel@gmail.com)

## Summary

Expertise in Algorithm design, Code-First Developer, Software Architecture, System Integration, Research and Development, Internet Standards, REST API Design Development and Implementation, Technical writing and Documentation

**Availability** Immediate

## Keywords



# Previous Experiences

**Client: Foo Studio (Mexico)**

Feb 2023 – May 2023

---

## Task

Implement Figma designs as fully functional webpages, update legacy websites to newer designs from Figma, troubleshooting old codebases

## Platform

HTML/JS vanilla, PHP, Laravel, React, Vue, Linux, Figma, Slack

## Key Highlights

“Pixel Perfect” webpage implementation based on vector graphics, developing methods to draw files based on parameters (section colors), REST-like PHP api implementation based on my own design.

**Client: Covrel Spain**

February 2022 – August 2022

---

## Task

Assigned a legacy code, both back-end and front-end, to fix completely non-functional multi-user real-time voice chat with 3d avatars in a virtual space

## Platform

Front end mixed React and Javascript vanilla

Back end with a Docker / Docker-Compose with Redis, MongoDB, Express.

## Key Highlights

The task involved massive refactoring of a code base written in an ad-hoc manner, implementing several sub-systems abstractions and general clean up of the code base.

Fixing the back-end installation from manually edited files to automatic configuration generation and installation in server, with the purpose of having multi-tenant capacity with a single code base.

Researching the real-time voice options, selected Jitsi and Jitsi-meet to be self-hosted in an Ubuntu VPS in Oracle Cloud service to implement the voice chat feature from zero, integrating it with the front end and the back end using socket.io and express

The integration between React and Threejs in vanilla Javascript with Jitsi on the frontend making all three independent and able to coexist

Researching the tech for Oculus VR, implementing from zero the front-end capabilities to use the VR Immersive platform properly along with detecting the limitations of the platform

## **Task**

Improved PHP legacy code written in an ad-hoc manner.

Add methods to serve the database legacy system, keeping current behaviour, and new behaviour for new front-end pages for the Realtor personnel to allow each one of them to have independent features and designs for their work

## **Platform**

Back end: Docker / Docker-Compose with Redis, MongoDB, Express.

Written in PHP / MySQL in Ubuntu/Debian using client's VPN, Git using "bare repositories" hosted in client's premises.

## **Key Highlights**

The PHP was restructured to create a REST-like API interface built to also support legacy direct HTML parameterised URLs with sane defaults, the code was refactored in a way to centralise queries in a single location for each unique query, splitting massive redundant queries into smaller composable parts, and writing proper error checking with suitable handling for all cases

The front end was build from scratch, based on a Photoshop design template, using React.

The first stage was hand-rolled with love, implementing a Real-estate search and Listing system loosely based on the existing legacy design, and designing a contact form with integration with the legacy system.

This project then had to be modified at a later stage to allow for every user's unique request to be processed.

The second stage involved a management system to edit the user's personal page to allow changing different aspects including background images, color scheme, page layout and text blurbs.

This management system was developed as a new layer on top of the old PHP query system, adding to use the same data from different systems that I had no control of.

Several types of tasks

## **Task**

Designed and developed a Software-as-service for Cloud Based Machine Learning service with integrated user accounts, payments, user micro-transactions credit usage.

## **Platform**

Hosted on Google Cloud Platform using Firebase, Google Computing, and Stripe Payment integration, Front end using React and Bootstrap.

## **Key Highlight**

Got a single HTML graphic design file with no functionality, designed the interactivity and behaviour of the front-end based on the design and client's feedback, designed and implemented the REST API with Firebase Serverless features, developed a serverless management system to create and destroy dynamic instances for processing the data on request, added front and back end real-time data to notify users of the processing and account details

Modified Python source code to adapt the ML model into a REST-like API style communication, added access database & event management in a layer above the Python program using NodeJS/Express.

## **Task**

YouTube Live Radio Streaming station back end

## **Platform**

GCP, X11 RemoteDesktop Linux (Remmina, TigerVNC), Ubuntu Virtual Machine, OBS, Express js

## **Key Highlights**

Research and development of a solution for streaming using YouTube API, using an Ubuntu cloud instance and OBS with YouTube API hooks for basic stream management.

Designed scripts to play and change the stream feed.

## **Task**

AWS and Online Store Connection back end

## **Platform**

AWS Amplify for Serverless and Lambda functions, Node/Express, DynamoDB

## **Key Highlight**

Got the design of a "Store delivery like Uber" mobile app. Designed the back end functionality to use the data of several 3rd party stores without API access to give a list of products in a REST standard API

## **Task**

Landing Pages

Designed several landing pages for sales of different products

**Task**

Straightforward implementation of Java / Android app

**Highlights**

Got a graphic design with a bare java screen. Implemented the code based on client's design, connected with Google Maps and Client's Rest-like API

**Tasks**

Implement from scratch the designs and functionality required by the client, implement Google Maps API and client's backend API, coordinate with the Client's BackEnd developers to deliver the required data

# Personal projects

Currently Active

## Task

Researching the possibilities of the Oculus VR platform and hardware with web-based interactive 3D experiences with a multi-platform design for Android Mobile, PC Chrome Browser, Meta Oculus Browser, in a single code base

## Platform

Oculus web browser, Javascript Vanilla, React js, Node js, Express js, Three js, Three-React-Fiber / Drei for React 3D real-time graphic

## Key Highlights

Monitoring data transfers between device and PC to identify underlying technologies, software and mechanics of the device's internal functionality. Researching the device's history and trends. Studying hardware characteristics. Testing and trying different techniques to make the hardware more optimization.

## Task

Mentoring and Teaching for beginner developers on Discord, Forums and other social medias, over voice and screen share, helping with debugging tasks, teaching of software development practices and technical details, and teaching useful tools and methods to debug and solve their tasks

## Platform

Discord, Zoom, TeamViewer, VNC, Remote Desktop, Others

## Key Highlights

Helping people improve their learning and scores with one-on-one sessions of multiple hours going through the material and explaining the different aspects that are not usually covered by their course, Sharing experiences and "Know-how"

## Task

Create a platform for managing and organizing project design documents for Independent Creators by designing tools which permit members to decide on each detail of the project by voting on proposals offered by any member of the team

## Platform

PHP, MySQL, React, REST-like API

## Key Highlights

Multi-user CRUD synchronized on realtime across members of a team, where

each can see immediately what is the current state of the documents

### **Older personal projects**

- Lisp like scripting language in C
- Emulator of a Primitive CPU in C
- Non-released Video games in C, Java, Javascript, Net, Unity, Godot, others
- Hands-on security research on technologies
- Disallowed use of software for experimentation and learning
- Manual art and crafts incl. painting, music writing, story writing, web comic creation