

# JUAN CEDENO

---

## SKILLS

Autodidact, fast-learner, math-wizard, and fast-writer (100 WPM). I'm a self-taught backend and devops engineer, as well as a cloud architect, with an obsession for knowledge and efficient software. I also speak English, Spanish, and Italian.

## TECHNOLOGIES

- Linux - 4 years.
- Git - 3 years.
- SQL(Postgres) & NoSQL(MongoDB) - 3 years.
- CI/CD - 2 years.
- Cloud-Native Designs - 2 years.
- Docker & Kubernetes - 2 years
- OOP & Functional Programming - 5 years.
- AWS EC2, Microsoft Azure, Google Cloud - 3 years.
- Infrastructure as a Service (Terraform & Ansible) - 2 years.
- Software as a Service - 2 years.
- Rest APIs - 3 years.
- Caching (Redis) - 3 years.
- Load Balancing (HAProxy) and VPNs - 2 years.

- Computer Vision (OpenCV) - 1 year.
- InterPlanetary File System (IPFS.io) - 6 months.

## PROGRAMMING LANGUAGES

- Java - 5 years (proficient).
- JavaScript (Nodejs) - 4 years (proficient).
- Python - 3 years (proficient).
- Bash Scripting - 3 years (proficient)
- C - 3 years (adept).
- HTML - Technically not a programming language - 2 years (adept).
- C# - 2 years (adequate).
- Rustlang - 2 years.
- GoLang - 1 year.
- Solidity - 2 months (beginner, but it's virtually the same as c++).

## EXPERIENCE

### **Hynix LLC, Remote** – *President & Chief Technologies Officer*

September 2018 - PRESENT

My first and only experience in the Tech Industry so far. Most of our stack is written in Java. I founded this company because the job I wanted did not exist at the time, so I created it. Running a small software company is a great learning experience, here are a few of my responsibilities:

- Avoid bankruptcy at all costs.

## EDUCATION

### Valencia College, Orlando – Degree

January 2020 – December 2021

Associate in Arts and enrolled to transfer to UCF to pursue a Bachelors in Computer Science. The most important classes from this period of my life were: Trigonometry, Calculus I & II, Meteorology, International Politics, and Introduction to C.

## REPOSITORIES

Although my github profile is a work in progress, here are some public repos that I'm proud of having worked on because of the technologies used and/or developed for each of the projects.

- [OpenSquid](#) is a Java 16 Project, using gradle for dependency management, written using Spigot, an event-driven API. The class [LineVector.java](#) is a direct application of what I learn from MIT OCW's Linear Algebra [Course](#).
- [SkinToolPython](#), a Python Restful API that utilises opencv to copy pixels from a Player's Minecraft [Skin](#) and generates new variants by pasting those pixels on to premade template skins. [Demo](#).
- [SkinToolIPFS](#), a SpringBoot Rest API that acts as a "front-end" for skin-tool-python. Originally, it used the InterPlanetary File System Protocol to store the data, but after some complications I settled with redis for storage.
- [CSF](#), another minecraft plugin I recently helped a friend develop. Though the mechanics of this project are not special, the way in which I wrote it, the documentation, and github actions integration are a modern example of my current code quality (A+ code only).
- [Dedsafio Bingo](#), our first collaboration with [TwitchRivals ES](#). A Java 16 Spigot Plugin, for this project I had to develop a data concurrency system – using redis-, a teams system, a tablist system to display the scoreboard, a public [Rest API](#) for twitch to query data from, and the [infrastructure](#) deployment for the project – using terraform and bash scripts. Digital Ocean's S3 Object Storage was utilized in this project too.
- [Lair](#), a JavaScript Rest API for a SaaS service that we called "Community Host," a service that allowed paying users to deploy minecraft server instances on demand from within the game. MongoDB was used to store all the long-term data, and redis was used by the instances themselves to communicate game data to the so-called "[Lobby](#)" instances to make themselves accessible. This project used AWS EC2, Azure, and Vultr instances – dependent upon availability and price at the moment of request.
- [Kern](#), a swiss-army-knife repository that contained all the necessary logic and dependencies for all other projects in the Noobsters Minecraft Server to function. My favorite part of the project was the punishments system which used a single MongoDB Query to detect if an user was trying to bypass a punishment in (almost) constant time.
- [InventorySync](#), an old Kotlin minecraft server plugin that allows users to sync their inventory across servers using base64 encoding and redis for the data transmission.