

Contact

✉ cfabrica46@gmail.com
in [linkedin.com/in/cfabrica46](https://www.linkedin.com/in/cfabrica46)
github.com/cfabrica46
instagram.com/cfabrica46
twitter.com/cfabrica46

Main Skills

Software Development
Web Development
Golang
Linux

Languages

Spanish (Native or Bilingual)
English (Limited Working)
Italian (Elementary Proficiency)

Certifications

*React – Definitive guide: hooks
router redux next + Projects*
*Blockchain Complete Course from
zero to expert*
*Basic English: All the essentials for
spanish speakers*

César Caycho

Software Engineer (Self-employed)
Lima metropolitan area

Extract

Software Engineer with 1.6 years of experience with knowledge in the following technologies: [Golang](#) + [Cobra](#) + [Gin](#) and [Echo](#), [PHP](#), [NodeJS](#), [JavaScript](#) + [React](#), [HTML](#), [CSS](#), [Shell Script](#), [MySQL](#), [PostgreSQL](#), [SQLite](#), [LevelDB](#), [Redis](#) and [MongoDB](#) databases, [Linux](#) server administration. [Blockchain](#) and [Solidity](#). [Docker](#).

Experience

Self-employed
Software Engineer
June 2020 - Present (1 year 6 months)

Education

UTP Universidad Tecnológica del Perú
Software Engineering, Software Engineering · (August 2021 - June 2026)

Proyectos

chat-gin-web-socket (Oct 2021) [Source Link](#) [Page Link](#)

Chat rooms in real time developed with [Golang](#) and [React](#), in the server was implemented [Gin Framework](#) and [Web Socket](#) for chat communication in real time, in the Frontend was used the [Webpack](#) tool for packaging and task automation, site styles were made with [SASS](#), a [Docker](#) and [Docker-compose](#) configuration was implemented, the deployment was done on a [Heroku server](#).

crud-with-redis-cache (Sep 2021) [Source Link](#)

CRUD for user management developed with [Golang](#) and [TypeScript](#), on the server we implemented [Gin Framework](#), we used [JWT](#) for the authentication system with the backend API, we stored the [JWT](#) in the NoSQL [Redis](#) database as cache and [PostgreSQL](#) for the user data; on the frontend we used [Webpack](#) tool for packaging and task automation, integrated [Babel](#) for [JavaScript](#) transpilation for browser compatibility, styling was done with [SASS](#), implemented [Docker](#) and [Docker-compose](#) configuration.

signal-transmission (May 2021) [Source Link](#)

Practice program developed in [Golang](#) that demonstrates the ability to send and receive messages between processes using system signals.

golang-keylogger (Jul 2021) [Source Link](#)

Keylogger made in [Golang](#), the captured data is stored in a text file for later management.

xcode-bits (Nov 2021) [Source Link](#)

Practice of encoding and decoding binary data developed in [Golang](#) with the CLI [Cobra library](#), the practice takes or generates an integer or a set of bits with the information to encode or decode.

social-network-mongodb (Jun 2021) [Source Link](#)

Practice social network with the ability to manage friends and posts developed with [Golang](#) and [JavaScript](#), the [Gin framework](#) was implemented on the server, the NoSQL [MongoDB](#) database was used, the site styles were made with [SASS](#).

js-snake (Jun 2021) [Source Link](#)

Practice of the well-known snake game, using the [JavaScript](#) API [Canvas](#).

tcp-reverse-terminal (Jan 2021) [Source Link](#)

Practice reverse terminal made in [Golang](#) using the [TCP](#) protocol.