

Roney de Oliveira

Software Engineer

Backend · Frontend · Full Stack · DevOps · Go (Golang) · Node.js · React · TypeScript · Kubernetes (K8s)

contact@roneyrogerio.dev | +55 43 99196-1524 | Apucarana, Parana, Brazil | roneyrogerio.dev |

www.linkedin.com/in/roneyrogerio | github.com/roneyrogerio

PROFESSIONAL SUMMARY

Software engineer with hands-on experience building web applications, APIs and production microservices end to end, across backend, frontend and DevOps.

Works mainly with Go, PostgreSQL, Node.js and React, plus cloud platforms (AWS, GCP and Oracle Cloud), containers and service-oriented architecture.

Focused on simple, scalable and maintainable code, with a continuous interest in performance, concurrency and distributed systems.

TECHNICAL SKILLS

Languages and frameworks: Go (Golang), TypeScript, JavaScript, Node.js, PHP, WordPress, C, C++, Python.

Frontend: React, Next.js (App Router), HTML and CSS, Astro, Vite, Three.js, Responsive interfaces.

Backend and architecture: Microservices, REST APIs, GraphQL, Webhooks, Server-Sent Events (SSE), Real-time video (Cloudflare RealtimeKit, Streaming), Idempotency, Gin, Service-oriented architecture (SOA), API versioning (Contract versioning), Automated testing, Secure development (Security), Code review, AI-assisted development, Asynchronous messaging (Message queues), Resilience (Fault tolerance), Scalability, Performance optimisation (Performance).

Cloud and operations: AWS Lambda (Serverless), Amazon S3, Amazon RDS, AWS CloudFormation, AWS API Gateway, AWS Rekognition, Google Cloud Platform (GCP), Cloud Run, Cloudflare R2 (Object storage), BigQuery, Pub/Sub, Vertex AI, Oracle Cloud Infrastructure (OCI, Oracle Cloud), Oracle Kubernetes Engine (OKE), Observability (Monitoring), OpenTelemetry (Distributed tracing, Tracing), Production troubleshooting, DevOps.

Data and platform: PostgreSQL (Relational databases), MongoDB, SQLite, Prisma, Data modeling, Query optimisation, Linux, Self-hosted LLM inference (Local LLM), Llama (Ollama), OpenAI API (LLM), Structured Outputs with JSON Schema (JSON Schema), Prompt engineering, Service networking, Docker, Kubernetes (K8s), Knative, CI/CD with GitHub Actions (Continuous integration, Continuous delivery), Terraform (Infrastructure as Code, IaC), Git, GitLab, Jira and Bash.

PROFESSIONAL EXPERIENCE

Full Stack Developer — Proaba

Aug 2026 – Sep 2026 · Freelance · Remote

- Live session broadcast and recording with Cloudflare RealtimeKit on a children's clinic platform: room creation, per-participant tokens, preset provisioning and teardown, all server-side.
- Recording written straight to Cloudflare R2, with automatic transcription and playback inside the session screen itself.
- Provider webhooks handled idempotently: overlapping events stopped creating two media records for one recording, and the retry the API itself asked for by answering 500 stopped being discarded as a duplicate.
- SSE stream released while the tab is hidden and closed in try/finally, so an open session does

not hold a slot at the provider.

- Access control requiring the child relationship to reach a recording, per-recording enablement, and listing by the viewer's timezone rather than the server's.
- Privacy by design: no patient name reaches the provider — the room is identified by the session id.
- End-to-end tests in Playwright, and provider secrets mapped per environment in the deploy.

Senior Software Engineer — e-didatico

Jul 2024 – May 2026 · Full-time · Remote

- Worked end to end across the engineering lifecycle, building and operating backend solutions on AWS and GCP.
- Applied DevOps and agile practices to deliver stable, efficient and high-value releases.
- Drove continuous evolution of a microservices platform with a focus on operational reliability.

Mid-Level Software Engineer — e-didatico

Jan 2022 – Jul 2024 · Full-time · Remote

- Implemented features with autonomy and strong system ownership.
- Handled bug fixes and evolutionary maintenance in production environments.
- Built backend services with Node.js and PostgreSQL supporting continuous product delivery.

Junior Software Engineer — e-didatico

Jul 2021 – Jan 2022 · Full-time · Remote

- Developed new features and fixed software defects.
- Supported delivery cycles with a focus on quality and product stability.

Helpdesk Analyst — e-didatico

Mar 2021 – Jul 2021 · Self-employed · Remote

- Supported students during online exam operations.
- Helped ensure service continuity and operational quality.

Full Stack Developer — Early Denver

Dec 2018 – Dec 2019 · Full-time · Brasilia, DF, Brazil (Remote)

- Worked on full stack web feature development.
- Used Node.js, React.js and PostgreSQL in product delivery workflows.
- Collaborated remotely with the team to continuously evolve the application.

Junior Software Developer — StutzLab

Jan 2018 – Nov 2018 · Full-time · Sao Paulo, SP, Brazil (Remote)

- Developed software features for web applications.
- Worked with PHP, JavaScript, MySQL, HTML and CSS to implement UI flows and business rules.
- Used Bootstrap to standardize interface components and speed up frontend delivery.

Web Developer — Independent projects

Nov 2010 – Dec 2017 · Self-employed · Remote

- Built independent projects with HTML, CSS, PHP, MySQL and Nginx.
- Monetized digital products and content through Google AdSense.

PROJECTS

Soltar Pipa — soltarpipa.online

Personal project · real-time multiplayer · Live at: soltarpipa.online

- Authoritative Go server: all flight physics and line-versus-line duelling run server-side, with the browser only rendering and sending inputs, which removes the client-side cheating surface.
- Custom binary WebSocket protocol, little endian, with a reference frame written by Go and

decoded by the TypeScript client test: changing one side without the other breaks the build.

- Interest management that treats each player as a line segment, hand to kite, on a 48 m spatial grid with a 120 m radius and hysteresis — 12 μ s per player, zero allocation.
- Profile-guided optimisation: a sorted sweep replacing all-pairs comparison (28 ms to ~1 ms per tick) and quickselect replacing sorting in per-send kite selection (16% to 2.4% of tick CPU).
- Purpose-built load test with 5,000 concurrent WebSockets, no connection failures or drops; production runs the room at 140 flyers and 15 frames per second, a limit set by the 10 Mbps load balancer rather than by CPU.
- Runs on Knative over Kubernetes with two services and two domains, images on GHCR and automated versioning through release-please.

Go, TypeScript, Three.js, WebSocket, Vite, Docker, Kubernetes, Knative.

Multilingual CV — roneyrogerio.dev

Personal project · Repository: github.com/roneyrogerio/curriculum

- Résumé tailoring to a job posting with an LLM (OpenAI API with Structured Outputs): the model receives identified facts and returns only selection, order and wording; employer, dates and links come from the source.
- Automatic verification of every rewritten sentence against its source, rejecting any technology or number the fact does not state.
- Multilingual static site with a reading version and an A4 print version, plus PDF and DOCX exporters validated against an ATS parsing simulator.
- Automated deployment through GitHub Actions on every release, with versioning and changelog by release-please and a multi-architecture image (linux/amd64 and linux/arm64) published to GitHub Container Registry.
- Runs as a Knative Service on Kubernetes (Oracle Kubernetes Engine), with a dedicated namespace and custom domain mapping.

Astro, TypeScript, Node.js 24, Docker, GitHub Actions, OpenAI API, Kubernetes, Knative.

Receitex — receiptex.com.br

Personal project · unmaintained, still online · Live at: receiptex.com.br

- Recipe site with a plugin of its own, CulinAI: a React panel inside the WordPress admin that generates a whole recipe — title, ingredients with fractions, method, categories and tags — and publishes it as a post.
- Text and dish photo generated by AI, with the image uploaded to the media library through the plugin's own REST routes.
- Packaged into a container from the official WordPress image, running as a Knative Service on Kubernetes, with MySQL, its own domain and TLS.
- No longer maintained, and online on purpose: it is the record of a whole site written by AI back when that was new. The images are from the first generation of those models — recognisable as the right dish, and visibly short of what is produced today.

WordPress, PHP, React, OpenAI API, MySQL, Docker, Kubernetes, Knative.

minishell

42 Sao Paulo · C · Repository: github.com/roneyrogerio/minishell

- Linux shell implemented in C using system calls only, without lexer libraries.
- Support for multiple commands, single and double quotes, character escaping, pipelines and redirections (<, > and >>).
- Built-ins (echo, cd, pwd, export, unset, env, exit), signals (ctrl-C, ctrl-D, ctrl-\) and environment variables.

C, POSIX system calls.

Pipefy Client Management API

Personal project · 2026 · Repository: github.com/roneyrogerio/pipefy-client-management-api

- REST API in Go with Gin and SQLite persistence, for client records and invested-assets processing.
- GraphQL integration with Pipefy and webhook intake with idempotency control: the same event delivered twice is not processed twice.
- Distributed tracing with OpenTelemetry and Jaeger, and a complete local environment in Docker Compose — GraphQL mock, collector and hot-reloading API — brought up with one command.
- The same router serves local HTTP and AWS Lambda through API Gateway proxy integration, with automated tests over the routes.

Go, Gin, SQLite, GraphQL, OpenTelemetry, Jaeger, Docker Compose, AWS Lambda, API Gateway.

cub3D and raycast-engine

42 São Paulo · 2020 · Repository: github.com/roneyrogerio/cub3D

- Pseudo-3D game demo drawn by raycasting in C, with the engine extracted into its own library and the game consuming only its interface.
- The library exposes camera, minimap, window size and movement update, and draws the scene through a function pointer supplied by the caller.

C, Raycasting, Makefile.

libbmp

42 São Paulo · 2020 · Repository: github.com/roneyrogerio/libbmp

- Pure C library to read, create, edit pixel by pixel and write BMP images at 24 and 32 bits per pixel.
- Works directly on the file format and little-endian byte order, with no external dependency.

C.

ft_server

42 São Paulo · 2020 · Repository: github.com/roneyrogerio/ft_server

- A single Docker image serving NGINX, PHP, MySQL, WordPress and phpMyAdmin, configured from scratch.

Docker, NGINX, PHP, MySQL, WordPress.

ft_services

42 São Paulo · 2020 · Repository: github.com/roneyrogerio/ft_services

- System administration and networking project: provisioning and orchestration of services in containers.

System administration, Networking, Containers.

libasm

42 São Paulo · 2020 · Repository: github.com/roneyrogerio/libasm

- C standard library functions rewritten in Assembly, keeping the original calling conventions and error returns.

Assembly, C.

libft

42 São Paulo · 2020 · Repository: github.com/roneyrogerio/libft

- Reimplementation of the C standard library functions — memory, strings, conversions and file-descriptor output — written from scratch to learn what each one costs.
- Includes a linked list of its own, with creation, insertion at both ends, iteration, mapping and release.

C.

ft_printf

42 São Paulo · 2020 · Repository: github.com/roneyrogerio/ft_printf

- Reimplementation of printf in C, with a variable number of arguments and dispatch by format specifier, built on the libft above.

C, Variadic functions.

get_next_line

42 São Paulo · 2020 · Repository: github.com/roneyrogerio/get_next_line

- A function that reads a file line by line from a descriptor using only read, malloc and free, with the buffer size fixed at compile time.
- The second version fixed the leak the first one left when a file was not read to the end — the pending state stayed allocated with no owner.

C, POSIX system calls.

C++ modules

42 São Paulo · 2021 · Repository: github.com/roneyrogerio/cpp_module_00

- The first 42 C++ modules: reading command-line arguments, and an in-memory phone book with ADD, SEARCH and EXIT commands.
- The next module gathers five more exercises, in cpp_module_01.

C++.

EDUCATION

- **Technology degree in Systems Analysis and Development** — UNOPAR — Universidade Norte do Parana, Jan 2022 — Dec 2023 (Final grade 9.5/10) · [diploma](#)
- **Technical education in Software Engineering** — 42 Sao Paulo, 2020 — 2021

CERTIFICATIONS

- **Next.js App Router Fundamentals** — Vercel, Aug 2026 · [dashboard-app](#)
- **React Foundations for Next.js** — Vercel, Aug 2026 · [react-foundations](#)
- **Programming with Google Go Specialization** — University of California, Irvine, May 2026 · [2UUQCPO9MST7](#)
- **Concurrency in Go** — University of California, Irvine, May 2026 · [M2EKXK59LHYI](#)
- **Functions, Methods, and Interfaces in Go** — University of California, Irvine, May 2026 · [WIEGNHPO8TBM](#)
- **Getting Started with Go** — University of California, Irvine, May 2026 · [KFGD3IF75IOQ](#)

ADDITIONAL COURSES

- **Introduction to the Python Language** — Unopar, 15 h, first half of 2023 · [certificate](#)
- **Introduction to Data Analysis with Python** — Unopar, 15 h, first half of 2023 · [certificate](#)
- **Data Structures in Python** — Unopar, 10 h, first half of 2023 · [certificate](#)
- **Information Technologies Applied to Law** — Unopar, 60 h, second half of 2023 · [certificate](#)
- **Electronic Law** — Unopar, 60 h, second half of 2023 · [certificate](#)
- **Management Models** — Unopar, 60 h, second half of 2023 · [certificate](#)

LANGUAGES

- **Portuguese** — Native
- **English** — Basic (A2)

ADDITIONAL SKILLS

Software Engineer, Backend Engineer, Back-end Developer, Backend Developer, Frontend Engineer, Front-end Developer, Full Stack Engineer, Full Stack Developer, Go Developer, Golang Developer,

Distributed systems, Real-time systems, WebSocket, SQL, NoSQL, Amazon Web Services, High availability, Reliability, Agile, Scrum.