

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, Paraná, Brasil | roneyrogerio.dev |

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

RESUMEN PROFESIONAL

Ingeniero de software con experiencia en aplicaciones web, APIs y microservicios en producción, trabajando de extremo a extremo entre backend, frontend y prácticas DevOps.

Trabajo principalmente con Go, PostgreSQL, Node.js y React, además de cloud (AWS, GCP y Oracle Cloud), contenedores y arquitectura orientada a servicios.

Enfoque en código simple, escalable y fácil de mantener, con interés continuo en rendimiento, concurrencia y sistemas distribuidos.

COMPETENCIAS TÉCNICAS

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

Frontend: React, Next.js (App Router), HTML y CSS, Astro, Vite, Three.js, Interfaces responsivas.

Backend y arquitectura: Microservicios (Microservices), APIs REST (REST APIs), GraphQL, Webhooks, Server-Sent Events (SSE), Vídeo en tiempo real (Cloudflare RealtimeKit, Streaming), Idempotencia (Idempotency), Gin, Arquitectura orientada a servicios (SOA), Versionado de APIs (Versionado de contratos), Pruebas automatizadas, Desarrollo seguro (Seguridad), Revisión de código (Code review), Desarrollo asistido por IA (AI-assisted development), Mensajería asíncrona (Comunicación asíncrona), Resiliencia (Tolerancia a fallos), Escalabilidad, Optimización de rendimiento (Performance).

Cloud y operaciones: AWS Lambda (Serverless), Amazon S3, Amazon RDS, AWS CloudFormation, AWS API Gateway, AWS Rekognition, Google Cloud Platform (GCP), Cloud Run, Cloudflare R2 (Almacenamiento de objetos), BigQuery, Pub/Sub, Vertex AI, Oracle Cloud Infrastructure (OCI, Oracle Cloud), Oracle Kubernetes Engine (OKE), Observabilidad (Monitorización), OpenTelemetry (Trazado distribuido, Tracing), Diagnóstico en producción, DevOps.

Datos y plataforma: PostgreSQL (Bases de datos relacionales), MongoDB, SQLite, Prisma, Modelado de datos, Optimización de consultas, Linux, Ejecución local de LLMs (Self-hosted LLM, Inferencia local), Llama (Ollama), API de OpenAI (OpenAI API, LLM), Salidas estructuradas con JSON Schema (Structured Outputs), Ingeniería de prompts (Prompt engineering), Redes para servicios, Docker, Kubernetes (K8s), Knative, CI/CD con GitHub Actions (Integración y entrega continuas), Terraform (Infraestructura como código, IaC), Git, GitLab, Jira y Bash.

EXPERIENCIA PROFESIONAL

Desarrollador Full Stack — Proaba

ago/2026 – sept/2026 · Freelance · Remoto

- Transmisión en directo y grabación de sesiones con Cloudflare RealtimeKit en una plataforma de clínica infantil: creación de la sala, emisión de token por participante, aprovisionamiento de presets y cierre al final, todo en el servidor.
- Grabación escrita directamente en Cloudflare R2, con transcripción automática y reproducción en la propia pantalla de la sesión.
- Webhooks del proveedor tratados con idempotencia: los eventos superpuestos dejaron de crear

dos registros de medios para una misma grabación, y el reenvío que la propia API pidió al responder 500 dejó de descartarse como duplicado.

- Stream SSE liberado mientras la pestaña está oculta y cerrado en try/finally, para que una sesión abierta no retenga una plaza en el proveedor.
- Control de acceso que exige vínculo con el niño para alcanzar una grabación, autorización grabación a grabación, y listado según la zona horaria de quien mira en lugar de la del servidor.
- Privacidad por diseño: ningún nombre de paciente se envía al proveedor — la sala se identifica por el id de la sesión.
- Pruebas end-to-end en Playwright y secretos del proveedor mapeados por entorno en el despliegue.

Ingeniero de Software Sénior — e-didatico

jul/2024 – may/2026 · Jornada completa · Remoto

- Trabajo de extremo a extremo en el ciclo de ingeniería, desarrollando y operando soluciones backend en cloud (AWS y GCP).
- Aplicación de prácticas DevOps y ágiles para garantizar entregas eficientes, estables y de alto valor.
- Evolución continua de una plataforma en microservicios, con foco en la fiabilidad operativa.

Ingeniero de Software Semisénior — e-didatico

ene/2022 – jul/2024 · Jornada completa · Remoto

- Implementación de funcionalidades con autonomía y dominio del sistema.
- Corrección de errores y mantenimiento evolutivo en entorno de producción.
- Desarrollo backend con Node.js y PostgreSQL, apoyando las entregas continuas del producto.

Ingeniero de Software Júnior — e-didatico

jul/2021 – ene/2022 · Jornada completa · Remoto

- Desarrollo de funcionalidades y corrección de errores.
- Participación en el ciclo de entrega con foco en la calidad y la estabilidad del software.

Analista de Soporte — e-didatico

mar/2021 – jul/2021 · Autónomo · Remoto

- Atención a los alumnos durante la aplicación de exámenes en línea.
- Soporte operativo para garantizar la continuidad y la calidad del servicio.

Desarrollador Full Stack — Early Denver

dic/2018 – dic/2019 · Jornada completa · Brasilia, DF, Brasil (Remoto)

- Desarrollo full stack de funcionalidades web.
- Trabajo con Node.js, React.js y PostgreSQL en el ciclo de entrega de producto.
- Colaboración remota con el equipo para la evolución continua de la aplicación.

Desarrollador de Software Júnior — StutzLab

ene/2018 – nov/2018 · Jornada completa · São Paulo, SP, Brasil (Remoto)

- Desarrollo de funcionalidades para aplicaciones web.
- Trabajo con PHP, JavaScript, MySQL, HTML y CSS en la implementación de pantallas y reglas de negocio.
- Uso de Bootstrap para estandarizar la interfaz y ganar productividad en el frontend.

Desarrollador Web — Proyectos propios

nov/2010 – dic/2017 · Autónomo · Remoto

- Desarrollo de proyectos propios con HTML, CSS, PHP, MySQL y Nginx.
- Monetización de productos y contenido digital mediante Google AdSense.

PROYECTOS

Soltar Pipa — soltarpipa.online

Proyecto personal · multijugador en tiempo real · En línea en: soltarpipa.online

- Servidor autoritativo en Go: toda la física de vuelo y el duelo entre hilos corren en el servidor, y el navegador solo dibuja y envía los controles, lo que elimina la superficie de trampa en el cliente.
- Protocolo binario propio sobre WebSocket, little endian, con un frame de referencia grabado por el Go y decodificado por la prueba del cliente en TypeScript: cambiar un lado sin el otro rompe la build.
- Área de interés que trata a cada jugador como un segmento de recta, de la mano a la cometa, en una rejilla espacial de 48 m con radio de 120 m e histéresis — 12 μ s por jugador, sin asignación de memoria.
- Optimización guiada por perfilado: barrido ordenado en lugar de comparar todos los pares (de 28 ms a ~1 ms por tick) y quickselect en lugar de ordenación al seleccionar cometas por envío (del 16% al 2,4% de la CPU del tick).
- Prueba de carga propia con 5.000 WebSockets simultáneos, sin fallos de conexión ni expulsiones; en producción la sala funciona con 140 jugadores a 15 fotogramas por segundo, límite impuesto por el ancho de banda del balanceador de 10 Mbps, no por la CPU.
- Producción en Knative sobre Kubernetes, con dos servicios y dos dominios, imágenes en GHCR y versionado automatizado con release-please.

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

Currículum multilingüe — roneyrogerio.dev

Proyecto personal · Repositorio: github.com/roneyrogerio/curriculum

- Adaptación del currículum a una oferta mediante LLM (API de OpenAI con Structured Outputs): el modelo recibe hechos identificados y devuelve solo selección, orden y redacción; el empleador, las fechas y los enlaces vienen de la fuente.
- Verificación automática de cada frase reescrita contra su origen, rechazando toda tecnología o cifra que el hecho no afirme.
- Sitio estático multilingüe con versión de lectura y versión de impresión A4, más exportadores de PDF y DOCX validados contra un simulador de análisis de ATS.
- Despliegue automatizado con GitHub Actions en cada release, con versionado y changelog por release-please e imagen multiarquitectura (linux/amd64 y linux/arm64) en GitHub Container Registry.
- Ejecución como Knative Service sobre Kubernetes (Oracle Kubernetes Engine), con namespace dedicado y dominio propio.

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

Receitex — receitex.com.br

Proyecto personal · sin mantenimiento, todavía en línea · En línea en: receitex.com.br

- Sitio de recetas con plugin propio, CulinAI: un panel en React dentro del admin de WordPress que genera la receta entera — título, ingredientes con fracciones, elaboración, categorías y etiquetas — y la publica como entrada.
- Texto y foto del plato generados por IA, con la imagen enviada a la biblioteca de medios mediante rutas REST propias del plugin.
- Empaquetado en contenedor a partir de la imagen oficial de WordPress, ejecutándose como Knative Service sobre Kubernetes, con MySQL, dominio propio y TLS.
- Ya no se mantiene, y sigue en línea a propósito: es el registro de un sitio entero escrito por IA cuando eso todavía era una novedad. Las imágenes son de la primera generación de los

modelos — reconocibles como el plato correcto, y visiblemente por debajo de lo que se produce hoy.

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

minishell

42 São Paulo · C · Repositorio: github.com/roneyrogerio/minishell

- Terminal Linux implementado en C con llamadas al sistema, sin biblioteca de lexer.
- Soporte para múltiples comandos, comillas simples y dobles, escape de caracteres, pipelines y redirecciones (<, > y >>).
- Built-ins (echo, cd, pwd, export, unset, env, exit), señales (ctrl-C, ctrl-D, ctrl-\) y variables de entorno.

C, Llamadas al sistema POSIX.

Pipefy Client Management API

Proyecto personal · 2026 · Repositorio: github.com/roneyrogerio/pipefy-client-management-api

- API REST en Go con Gin y persistencia en SQLite, para el registro de clientes y el procesamiento de patrimonio invertido.
- Integración GraphQL con Pipefy y recepción de webhooks con control de idempotencia: el mismo evento entregado dos veces no se procesa dos veces.
- Trazado distribuido con OpenTelemetry y Jaeger, y entorno local completo en Docker Compose — mock GraphQL, colector y API con recarga automática — levantándose con un solo comando.
- El mismo enrutador sirve HTTP local y AWS Lambda mediante proxy integration de API Gateway, con pruebas automatizadas que cubren las rutas.

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

cub3D y raycast-engine

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

- Juego demo pseudo-3D dibujado por raycasting en C, con el motor extraído a una biblioteca propia y el juego consumiendo solo su interfaz.
- La biblioteca expone cámara, minimapa, tamaño de ventana y actualización de movimiento, y dibuja la escena mediante un puntero a función que pasa quien la llama.

C, Raycasting, Makefile.

libbmp

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

- Biblioteca en C puro para leer, crear, modificar píxel a píxel y grabar imágenes BMP de 24 y 32 bits por píxel.
- Trabaja directamente sobre el formato del archivo y el orden de bytes little endian, sin dependencias externas.

C.

ft_server

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

- Imagen Docker única sirviendo NGINX, PHP, MySQL, WordPress y phpMyAdmin, configurados desde cero.

Docker, NGINX, PHP, MySQL, WordPress.

ft_services

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

- Proyecto de administración de sistemas y redes: aprovisionamiento y orquestación de servicios en contenedores.

Administración de sistemas, Redes, Contenedores.

libasm

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

- Funciones de la biblioteca estándar de C reescritas en Assembly, con las convenciones de llamada y el retorno de error del original.

Assembly, C.

libft

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

- Reimplementación de las funciones de la biblioteca estándar de C — memoria, cadenas, conversiones y salida por descriptor —, escritas desde cero para entender lo que cuesta cada una.
- Incluye una lista enlazada propia con creación, inserción por ambos extremos, iteración, mapeo y liberación.

C.

ft_printf

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

- Reimplementación de printf en C, con número variable de argumentos y despacho por especificador de formato, apoyada en la libft propia.

C, Funciones variádicas.

get_next_line

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

- Función que lee un archivo línea a línea desde un descriptor, usando solo read, malloc y free, con el búfer definido en tiempo de compilación.
- La segunda versión corrigió la fuga que dejaba la primera cuando el archivo no se leía hasta el final — el estado pendiente quedaba asignado sin dueño.

C, Llamadas al sistema POSIX.

Módulos de C++

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

- Primeros módulos de C++ de 42: lectura de argumentos de la línea de comandos y una agenda en memoria con los comandos ADD, SEARCH y EXIT.
- El módulo siguiente reúne cinco ejercicios más, en cpp_module_01.

C++.

FORMACIÓN ACADÉMICA

- **Técnico Superior en Análisis y Desarrollo de Sistemas** — UNOPAR — Universidade Norte do Paraná, ene/2022 — dic/2023 (Nota final 9,5) · [diploma](#)
- **Formación técnica en Ingeniería de Software** — 42 São Paulo, 2020 — 2021

CERTIFICACIONES

- **Next.js App Router Fundamentals** — Vercel, ago/2026 · [dashboard-app](#)
- **React Foundations for Next.js** — Vercel, ago/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](#)

CURSOS COMPLEMENTARIOS

- **Introducción al Lenguaje Python** — Unopar, 15 h, 1.er semestre de 2023 · [certificado](#)
- **Introducción al Análisis de Datos con Python** — Unopar, 15 h, 1.er semestre de 2023 · [certificado](#)
- **Estructuras de Datos en Python** — Unopar, 10 h, 1.er semestre de 2023 · [certificado](#)
- **Tecnologías de la Información Aplicadas al Derecho** — Unopar, 60 h, 2.º semestre de 2023 · [certificado](#)
- **Derecho Electrónico** — Unopar, 60 h, 2.º semestre de 2023 · [certificado](#)
- **Modelos de Gestión** — Unopar, 60 h, 2.º semestre de 2023 · [certificado](#)

IDIOMAS

- **Portugués** — Nativo
- **Inglés** — Básico (A2)

COMPETENCIAS ADICIONALES

Ingeniero de Software, Software Engineer, Desarrollador Back-end, Desarrollador Backend, Backend Engineer, Desarrollador Front-end, Desarrollador Frontend, Frontend Engineer, Desarrollador Full Stack, Full Stack Engineer, Sistemas distribuidos, WebSocket, SQL, NoSQL, Amazon Web Services, Alta disponibilidad, Fiabilidad, Metodologías ágiles, Scrum.