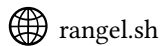


# LUCAS RANGEL

SÃO PAULO, BRAZIL

Senior Full Stack Software Engineer | Early-Stage Startups & AI/ML Systems



---

## WORK EXPERIENCE

### Yellowpad

05/2024–Present

*Founding Engineer*

Remote

[yellowpad.ai](https://yellowpad.ai)

Built an agentic document editor for contract review and drafting with human-in-the-loop workflows. Key technical contributions:

- Architected LangGraph-based agent workflows for contract drafting and review with retrieval-augmented generation for contextual document understanding.
- Built evaluation framework measuring agent goal and tool call accuracy.
- Built observability with OpenTelemetry for logging, metrics, and alerting in production.
- Built tool execution framework with state synchronization enabling simultaneous AI and manual document editing.
- Built Next.js application with real-time document collaboration UI.
- Developed type-safe database layer using Drizzle ORM with multi-tenant data isolation patterns and SOC 2 compliance.
- Architected GCP infrastructure with Terraform, including Cloud Run and Pub/Sub message queuing.
- Built bidirectional data transformation layer between Slate editor JSON format and Word Docx, preserving formatting and metadata.

### Plaza

07/2023–05/2025

*Founding Engineer*

São Paulo, Brazil

[useplaza.com.br](https://useplaza.com.br)

- As first engineering hire, built product from concept to hundreds of real estate companies, owning full technical architecture and implementation using Ruby on Rails, Hotwire, Stimulus Reflex, Phlex, and Sidekiq.
- Set up entire technical foundation from CI/CD to observability to infrastructure-as-code across Azure and AWS using Terraform and Docker.
- Shipped features weekly with automated testing, crucial for rapid iteration during product discovery phase.
- Built AI chatbot for real estate lead qualification through WhatsApp integration, improving closing rates by 30% with automated conversation flows and follow-up.
- Implemented RAG-based search to match real estate listings with user requirements.
- Drove product direction and roadmap, influencing key decisions and features based on customer feedback and market validation.

### Remoov

09/2020–07/2023

*Full Stack Software Engineer*

Remote

[remoovit.com](https://remoovit.com)

- Implemented new features end to end, utilizing Node.js, React, React Native and PostgreSQL for full-stack web and mobile applications.
- Developed internal tools using Rust to enhance developer productivity.
- Built code mods to aid in large refactoring projects, reducing technical debt while saving hundreds of engineering hours.
- Mentored junior engineers on best practices, code quality, and technical problem-solving.
- Solved Node.js performance and scaling issues by fixing memory leaks and improving memory usage, reducing startup time by ~2x and memory usage by ~4x.

### Nextel Brasil

06/2018–09/2020

*Data Engineer and Data Scientist*

São Paulo, Brazil

[Nextel Brasil](https://nextel.com.br)

- Developed a credit score model using machine learning algorithms to assess customer debt repayment likelihood.
- Built Hadoop data pipelines using Apache Airflow, Hive, Python, and Scala.

- Worked with query optimization for large datasets including geospatial data.

---

## EDUCATION

### Federal University of ABC - UFABC

2014–2019

- B.S. Computer Science
- B.S. Science and Technology
- Additional coursework:
  - Parallel computing, machine learning, and statistical inference
  - Computational neuroscience and bioinformatics
  - Functional and logic programming

---

## PERSONAL PROJECTS

### 🔗 **OxyLisp**, a Lisp interpreter in Rust.

- Tree-walk evaluator
- Interactive REPL with live demo at [rangel.sh/oxylist\\_repl](https://rangel.sh/oxylist_repl)

### 🔗 **phlex-emmet-lsp**, a Language Server Protocol implementation in Rust for [Phlex](#) templates.

- Uses a [PEG](#) parser
- Published on crates.io with VS Code and Neovim editor support

### 🔗 **plate-docx-deserializer**, a DOCX-to-Plate deserializer written in [Gleam](#).

- Compiles to JavaScript with TypeScript declarations
- Live demo at [plate-docx.vercel.app](https://plate-docx.vercel.app)

### 🔗 **ts-mcp-forge**, a TypeScript framework for building MCP servers.

- Decorator-based tool registration with automatic Zod schema generation from TypeScript types
- Toolkit system for grouping related tools with namespacing to avoid context bloating
- Published on npm with production-ready compliance test suite

### 🔗 **zmcp**, an MCP server bridging [Obsidian](#) vaults with AI assistants.

- Multiple tools for note management, task tracking, and flashcard generation
- Vault-based prompts and resources: Users can store and edit their prompts and resources in Obsidian and use them in any MCP client

### 🔗 **ankura**, a type-safe configuration tool for [Karabiner-Elements](#) using Apple's [Pkl](#).

- Declarative DSL for defining keymaps with autocomplete, type safety, validation, and editor support
- First-class support for popular macOS window managers and applications, live-reload for instant configuration updates
- Published as a Homebrew package