

Leonardo Santiago

 Github |  LinkedIn |  Blog |  Email

Hello, I'm Leonardo, a strong and opinionated software engineer based in Brazil, focused on correctness, reliability and reproducibility. I daily drive NixOS and master a myriad of tools, including rust, nix, python, postgresql, emacs (and lisps!), being very quick to learn new languages and paradigms. My interests are in formal proof assistants, compilers, language design, linux systems and low level programming.

If you need a sharp thinker able to quickly identify flaws and inefficiencies in your processes and develop robust fixes, while always valuing correctness, then look no further. I'm willing to hear you out.

WORK EXPERIENCE

Software Engineer

Jun 2023 - present

Mixrank

 Remote

At Mixrank I create value through removing complexity and improving reliability of our systems in a fast paced environment, constantly looking for inefficiencies and inconsistencies. I've implemented key features in our rust build system in pure nix, improved reliability of our plethora of systems in NixOS and removed the need for a lot of legacy processes by leveraging nix's power, while keeping every server running 24/7. Also dealt with PostgreSQL and Python implementing shiny new features for our changing needs.

Software Engineer

Mar 2022 - May 2023

Kindelia

 Remote

At Kindelia I worked thoroughly with an avant garde open-source functional runtime, implemented entirely in Rust. I identified logical problems in the implementation, audited code of a prototype of a blockchain, and developed libraries to Kind, the internal dependently typed proof language. I also pioneered and successfully delivered an MVP of a transpiler of other languages' (mainly python and javascript) source code to our runtime.

Junior Developer

Aug 2020 - Mar 2023

Editora Epapers

 Rio de Janeiro, Brazil

As a junior in a small book publisher, I mainly helped automatizing away a lot of repetitive needs using python, both in XML and in Latex documents. They reduced the time to check each document in about 80% and greatly increased other workers' productivity. Even after leaving, most of my scripts are used until this day.

PROJECTS

lowestcase

I maintain a small lower case onlu blog to talk about topics that usually habit my mind for a little too long, ranging from parser implementations for relations between proofs and programs. It went through a few iterations, and in one of those I implemented my [open source own static site generator tool](#) (almost entirely in OCaml), but nowadays I've settled on a much simpler emacs + hugo setup.

My NixOS config

I maintain the configuration of my own systems as a public and open source repository, as a way to instruct new users on how to set it up. Everything from secrets management, window manager configuration and even my emacs configuration is all declaratively written in a way that lets me share most of the code between my two work machines.

My Emacs config

In fact, my own emacs configuration itself is a nixos module, that can be imported into any system directly, with no hassle. It is all done in a single [org file](#), which is then parsed by my modified [lisp parser](#), which then finally generates the correct derivation to be built in NixOS. Surprisingly, this works very well, and I'm able to quickly add and remove packages whenever I want, without any problems.

nesP

A small NES emulator written entirely in Common Lisp. It is able to run super mario at a staggering 8

FPS, albeit with a ton of bugs. I had a ton of fun writing it, and seeing it render a real game was very rewarding. It probably the reason why I use emacs till this day, and why I got so interested in systems programming in the first place.

Open source contributions

I've done a fair bit of open source contributions, with probably my main one being implementing missing capabilities in feature resolution in the amazing [no-cargo](#) nix build tool. I'm very proud of this contribution, because it re-enabled nocargo to build almost all rust written software. I still use it in all my local rust projects to this day without any problems.

EDUCATION

2020 - 2025 Bachelor's Degree in Computer Science @ Federal University of Rio de Janeiro (UFRJ) (GPA: 3.6/4.0).
Scientific research in type theory, compilers and computation theory.

SKILLS

Portuguese	Native
English	Fluent (Cambridge FCE)
Japanese	Advanced (JLPT N2)
Spanish	Conversational
Tools	Git, Linux, Latex, XML, Org files, excel, and more.