

Breno Fatureto de Bortolli

Curriculum Vitae

- Software Engineering Intern at Datarisk
- Computer Engineering student at Universidade de Brasília - UnB
- breno.afb@gmail.com
- <https://brenoafb.com>
- <https://github.com/brenoafb>

Summary

I have a deep interest in the complete spectrum of computing – from the architecture of hardware to low-level software in C and assembly, as well as the high degrees of abstraction provided by functional languages such as Haskell and Lisp. I am particularly fascinated by the elegance and productivity afforded by functional, strongly typed programming.

Skills and Experience

Programming Languages

- Haskell
- F#
- Javascript/Typescript
- Python
- Swift
- C/C++
- Java
- Scheme
- LaTeX

Tools and Software

- Linux
- Emacs
- Vim
- git
- Unix tools
- macOS
- Xcode

Software Engineering Intern – Datarisk (Nov 2021 - present)

Datarisk¹ develops custom data processing solutions and technologies. As part of the technology team, I am helping develop the company's model API platform. We use F# and the .NET environment for development, along with Azure for online hosting.

- Developed a serverless solution based on Azure Functions for monitoring our service
- Worked on maintaining the platform's API, including reworking database processes and adding new API endpoints

¹<https://datarisk.io/en>

- Added pages to the company's webapp frontend.

Software Engineering Intern – Axur (May 2021 - Nov 2021)

Axur² is a cybersecurity company aiming to make the web a safer place.

I worked on the company's threat tracking platform. Mainly Javascript/Typescript stuff using React and a plethora of other JS frameworks.

- Worked on a agile team which incorporates modern DevOps practices
- Did fixes and added functionality to the company webapp.
- Developed a page containing interactive graphs showing data regarding online threat tracking to customers
- Added extensions to Java-based microservice APIs

Undergraduate researcher – Analysis of Software Product Lines (November 2019 - present)

Under Prof. Vander Ramos, since late 2019 I am working on enhancing and developing software for the analysis of Software Product Lines. My team is currently implementing new techniques in the literature, as well as developing a solution for reusing previous results in order to speed up the analysis.

So far the work has included contributing to the ReAna-SPL³ tool, written in Java. as well as writing small tools in Haskell to interpret and analyze code and results. We've implemented the first instance of a Feature-Family-Product-based analysis as described in Castro et al⁴ in ReAna-SPL.

We are working on implementing reuse of previous results when running an analysis. This aims at reducing the time needed to run an analysis where an existing model has been altered in some way, and also invites us to think deeply about the evolution of the same model over time.

Teaching Assistant – Functional Programming in Haskell (January 2022 - Present)

I am currently a teaching assistant in an introductory Haskell course for Computer Science and Engineering Students. The course introduces functional programming in Haskell, going from basic functions to applicative parsers and monads. As part of the course, I am developing a CI/CD pipeline on Github for automatically grading student assignments.

Teaching Assistant – Programming Languages (2020-2021)

I was a teaching assistant for the Programming Languages course for two semesters. During this period, I worked on

²<https://axur.com/en/>

³<https://github.com/SPLMC/reana-spl>

⁴<https://repositorio.unb.br/handle/10482/39241>

developing front-ends and interpreters for programming languages of multiple paradigms in Haskell.

Selected Projects

- Programming Language Implementation⁵

A sequence of interpreters and compilers of increasing complexity in Haskell, both in the language specification as well as the functional programming features used in the implementation, such as applicatives and monad transformers.

- Lisp Interpreters I have a thing for writing simple Lisp interpreters using different languages and techniques. I wrote two in Haskell, one written from scratch using only the standard library⁶, and a more advanced one⁷ using more elaborate libraries and techniques.

In order to practice writing in a more conventional, commercial language, I also wrote a version in Swift⁸. This one implements a more extensive standard library, as well as HTTP request and JSON capabilities.

One of the most rewarding parts of building Lisp interpreters is then writing a metacircular evaluator. That is, running a Lisp interpreter inside the Lisp interpreter you just wrote. It feels like you unlocked the deepest concept in computing.

- Site generator in Haskell⁹

Built using the Scotty web framework, this project serves Markdown files as a static website.

- ReAna Data Analysis¹⁰ As part of my undergraduate research program, I was tasked with writing code that performs runtime data analysis for the ReAna-SPL¹¹. This was done using Jupyter Notebook in Python.
- JVM¹² In a Systems Programming discipline, I led the construction of a reduced JVM written in C capable of running simple programs.

Natural Languages

- English – fluent
- Portuguese – native language

⁵<https://github.com/brenoafb/ProgrammingLanguageImplementation>

⁶<https://github.com/brenoafb/hlisp>

⁷<https://github.com/brenoafb/haskell-lisp>

⁸<https://github.com/brenoafb/SwiftLisp>

⁹<https://github.com/brenoafb/site-generator>

¹⁰<https://github.com/brenoafb/reana-data>

¹¹<https://github.com/SPLMC/reana-spl>

¹²<https://github.com/brenoafb/jvm>

Education

Computer Engineering – Universidade de Brasília (2017-present)

I am studying Computer Engineering at Universidade de Brasília. The course includes study of the entire range of computing, while also including some Electrical Engineering courses.

Most of the computing courses utilize C/C++ for the study of programs, data structures, and algorithms.

In the multiple laboratory courses I worked with FPGAs and Microcontrollers for the study of digital circuits, as well as with analog circuits and filters for the study of electrical engineering fundamentals. Highlights include developing a simulator for a RISC-V processor with the RV32I architecture in C, as well as implementing the CPU in VHDL.

As a student and teaching assistant at the programming languages course, I worked in the implementation of multiple small programming languages using Haskell. This experience has motivated me to deepen my understanding of functional programming and programming language implementation in general.

Complementary Courses

I have done various courses on interesting topics that are not necessarily covered in my University education.

- Build a Modern Computer from First Principles: From Nand to Tetris (Project-Centered Course) by Hebrew University of Jerusalem on Coursera. Certificate earned at Tuesday, January 15, 2019 2:10 AM GMT
- Machine Learning by Stanford University on Coursera. Certificate earned at Wednesday, July 24, 2019 3:19 AM GMT
- Neural Networks and Deep Learning by deeplearning.ai on Coursera. Certificate earned at Sunday, July 28, 2019 1:34 PM GMT
- Deep Neural Networks: Hyperparameter tuning, Regularization and Optimization by deeplearning.ai on Coursera. Certificate earned at Saturday, August 3, 2019 9:16 PM GMT
- Structuring Machine Learning Projects by deeplearning.ai on Coursera. Certificate earned at Tuesday, August 6, 2019 8:31 PM GMT
- Convolutional Neural Networks by deeplearning.ai on Coursera. Certificate earned at Friday, September 13, 2019 8:21 PM GMT
- Sequence Models by deeplearning.ai on Coursera. Certificate earned at Tuesday, December 31, 2019 8:04 PM GMT
- Deep Learning by deeplearning.ai on Coursera. Certificate earned at December 31, 2019