Egas Carvalho Perdigão Gonçalves Ribeiro

Lisbon, Portugal
egas.g.ribeiro@gmail.com — github.com/riogu

GCC contributor with merged patches across the C++ and Rust frontends and Static Analyzer. Interested in compiler middle-ends, static analysis, and language implementation.

Education

PhD Preparatory Phase (MSc), Computer Science *Starting Oct 2026*
Saarland Informatics Campus, Saarland University — Saarbrücken, Germany

Bachelor of Science in Computer Science & Engineering *Graduated July 2026*
Instituto Superior Técnico — Lisbon, Portugal

Open Source Contributions

Google Summer of Code 2026 — GNU Compiler Collection *May 2026 – Aug 2026*
Extending C++ Support in GCC's Static Analyzer — mentored by David Malcolm (Red Hat)

- Virtual call devirtualization (PR97114): devirt `OBJ_TYPE_REF` calls by decoding vtable initializers
- Exception subclass matching (PR119697): new language hook delegating to the C++ frontend's `can_convert_eh`, keeping the analyzer language-agnostic
- `dynamic_cast` support (PR110578), beyond the original proposal scope: statically evaluates the [expr.dynamic.cast]/9 runtime check over the class hierarchy

GNU Compiler Collection *Oct 2025 – Present*
Contributor with write-after-approval access, working across the C++ frontend, the Rust frontend (gccrs), and the static analyzer. Attended GNU Tools Cauldron 2025 and FOSDEM 2026.

- **C++ frontend:** 15+ merged patches fixing internal compiler errors across template instantiation, lambdas, CTAD, SFINAE, partial specialization, constexpr evaluation, and name lookup
- **gccrs:** implemented irrefutability checking for let bindings and function parameters
- Full commit history: gcc.gnu.org/cgit — Egas Ribeiro
- Developed Neovim plugin (gcc1plus) automating DejaGNU test execution and `cc1plus` debugging

Projects

henceforth — github.com/riogu/henceforth *October 2025 – Present*
Typed stack-based programming language compiler in Rust, exploring middle-end compiler design with focus on SSA-form IR, dataflow analysis, and optimization.

- Implemented complete compiler frontend (lexer, recursive descent parser, semantic analysis) and SSA-form MIR with CFG construction
- Built optimization pipeline: mem2reg via iterated dominance frontiers, worklist-driven DCE, Cooper-Harvey-Kennedy dominator tree computation
- Designed generational arena-based IR with def-use chains, RAUW support, and per-function analysis infrastructure

fumo-compiler — github.com/riogu/fumo-compiler *July – Sept 2025*
Statically-typed systems programming language compiler targeting LLVM IR, built from scratch.

- Implemented complete compiler pipeline: hand-written lexer and recursive descent parser, semantic analyzer, type checker, and LLVM IR code generator
- Implemented C++-style namespaces, structs with member functions, type inference, and `libc` interoperability

fumo-lib — github.com/riogu/fumo-lib *2025*
Single-header C library exploring compile-time type safety through C23 features: type-safe variant and Result types with pattern matching via macro-based generic programming.

Technical Skills

Specialization: Low-level systems programming, compiler implementation, static analysis
Languages: C++ (advanced), C, Rust, Python, OCaml
Tools: LLVM, GCC, Git, GDB, CMake, DejaGNU, Linux

Languages

Portuguese (native), English (fluent, C2), Japanese (fluent, JLPT N2)