

Marcos Brito

Brazil, São Paulo | marcos.brito0507@gmail.com | github.com/marcos-brito | linkedin.com/in/marcos-brito07 | marcos-brito.github.io/nnago/

Education

FATEC Mogi das Cruzes

Technologist in Systems Analysis and Development

Brazil, São Paulo

Expected July 2026

Projects

BookList

github.com/marcos-brito/booklist

- Designed an event system using Go and Redis streams to decouple service communication from API endpoints, which reduced code complexity and enabled real-time updates to users and search indexes via sidecar processes
- Designed a scalable, cloud-agnostic read-tracking system architecture using open-source tools such as, PostgreSQL, Meilisearch, Gorse and Ory, enabling cheap initial deployment on a \$40/month VPS
- Developed a GraphQL API using Go and gqlgen, integrating Ory Kratos for authentication and planning Ory Hydra for OAuth support, enabling secure public access for CLIs and mobile applications with user-controlled permissions
- Built an integration test suite for API resolvers using Docker and Testcontainers, allowing real service connections while maintaining fast execution times, avoiding mock code and increasing reliability

Henactor

github.com/marcos-brito/henactor

- Implemented a desktop file manager using Tauri's RPC and WebView, enabling users to apply custom CSS themes and configuration files at runtime while laying the foundation for JavaScript plugins to extend functionality
- Developed an interpreter with Rust and the pest parser generator that could transform a domain specific language into a boolean expression, enabling efficient breadth filtering and deep recursive searches for file system entries
- Designed a keyboard driven interface by creating a context-aware command system with Typescript, DOM events and a single listener that allowed custom keyboard shortcuts and was capable of intelligently resolve conflicts while also allowing manual command calls

Theris

github.com/marcos-brito/theris

- Developed a CLI tool using Rust and Clap to automate theme management for UNIX-based environments, allowing users to apply themes across multiple applications simultaneously using YAML configuration files, templates, and scripts
- Implemented a backup system using tarballs, symbolic links, and YAML manifest files, enabling automatic restoration of modified files while keeping track of the latest backup for seamless, unparameterized restores
- Created documentation using Markdown and mdBook as a static site generator to help users configure their environment, then automated deployment with GitHub CI to host the guide on GitHub Pages

Skills

- **Programming languages:** Go (Golang), Rust, Java, Typescript, Javascript, C, Haskell,
- **Technologies:** Git, Docker, Kubernetes (K8S), AWS, GraphQL, Redis, Supabase, PostgreSQL, Ory, OpenAPI
- **Frameworks and libraries:** Fiber, Echo, GORM, gqlgen, testify, sqlx, Bubble Tea, SvelteKit, Koa, Hono, Express, Drizzle ORM, Tauri, axum, clap