



# LUCAS NEIVA

Software Engineer — Computer Engineer

Belém, Brazil

✉ [lucasneiva07@gmail.com](mailto:lucasneiva07@gmail.com)  [linkedin.com/in/lucas-neiva](https://linkedin.com/in/lucas-neiva)  [github.com/Neiva07](https://github.com/Neiva07)

## Summary

---

Senior software engineer with 4y production Go and 7y TypeScript across backend, full-stack, and architectural ownership. At Seam, owned a cross-platform bridge service end-to-end (Windows/macOS/Linux) and drove a 94% cut in peak server load. At NTWRK, built Go microservices over MySQL and led a 4-engineer team through schema decoupling. Recent work at InventoryQuant: designed an event-sourced undo/redo system and a 30GB-file video pipeline on Next.js, Postgres/Drizzle, and Trigger.dev.

## Education

---

Federal University of Pará

Computer Engineering Major

2017 – 2022

Belém, Brazil

## Work Experience

---

InventoryQuant

Senior Software Engineer

Jan 2026 – Present

Remote

- Designed and implemented an **event-sourced change-tracking system** powering inventory undo/redo — `/api/inventory-changes/{undo,redo,status}` endpoints, a `useUndoRedo` React hook, and a History tab surfacing the change timeline. Backed by integration tests that replay event streams to verify state consistency.
- Enabled **30GB video processing** in production — parallelized frame extraction via **Trigger.dev** child tasks, resolved memory limits through memory-optimized buffering, and shipped frame-capture and “add from screenshot” flows on the client.
- Moved inventory pagination from client to backend — cut client memory pressure on large tables, eliminated full-dataset fetches, and paved the way for server-side filters, sorts, and sticky search. *Stack: Next.js 16 App Router, Drizzle ORM, PostgreSQL.*
- Re-architected file upload for large files and flaky mobile connections: **multipart video upload**, session-expiry error handling, mobile + multi-file upload, and a responsiveness pass across the app.
- Shipped multiple full-stack features: shareable read links with change-request flow, collaborator invitations (incl. non-registered users with auto-invite), **spreadsheet image extraction** with clickable image columns, team-based pricing configuration, and age/depreciation columns. *Stack: Next.js 16, Drizzle ORM, PostgreSQL, Trigger.dev.*
- Added **SSRF protection** for external image rehost — validate resolved IPs, pin connections, image allowlist. Strengthened **Sentry** observability: user context on all events, diagnostics wiring into file-processing captures, environment-gated init.

Seam

Senior Software Engineer

Sep 2023 – Jan 2026

San Francisco, CA (Remote)

- Drove architectural changes across production job orchestration and request flows in the **TypeScript / Next.js / Postgres** backend, eliminating critical scaling bottlenecks and reducing server load and external API calls by **94%** during peak usage.
- Designed and owned a cross-platform bridge service in **Go** for **Windows, macOS, and Linux** — secure cloud-to-on-premise communication for access-control systems. Built HTTP + SOCKS5 proxies with health checks, pairing flow with auth keys and session management, autostart-on-boot, and file-based log collection with in-dashboard download.
- Led backend development for 3rd-party integrations across 10+ hardware vendors — commercial ACS platforms (**Salto / Salto-KS, Dormakaba family, Visionline, Lenel**) and residential smart-lock APIs (**Lockly, Kwikset, Igloohome, Schlage, Latch, Assa Abloy**) — abstracting each behind unified Seam APIs with typed error maps. Built vendor fakes from scratch (e.g., a 20+ endpoint **Lenel** fake covering cardholders, access levels, and badges) to enable integration-test coverage. *Stack: TypeScript, Next.js, Postgres, REST, SOAP.*
- Built a production **Go logger** for the bridge service: **NDJSON** entries batched in memory with configurable flush interval and max batch size, HTTP export to a cloud telemetry endpoint with bearer-token auth, per-session metadata (Tailscale hostname, bridge-client-session ID), and goroutine-driven flush with graceful shutdown.
- Led stress and load testing initiatives: a dedicated API rate-limit testing harness, internal batch-user creation endpoints, and latency-simulation hooks into vendor fakes (Latch) — surfacing scaling limits and failure modes before production. *Stack: TypeScript, Next.js.*
- Drove **SOAP** → **REST migration** for the Dormakaba Ambiance integration and similar migrations for Dormakaba Community key-management endpoints, improving reliability over legacy transport.

## NTWRK

Mar 2022 – Aug 2023

Senior Software Engineer

Los Angeles, CA (Remote)

- Shipped the seller dashboard for NTWRK's live-commerce platform — product catalog with bulk editing and **Algolia** filter integration, seller onboarding, auth v2 migration, and team picker for multi-team users. *Stack: React, Next.js, TypeScript, Chakra, GraphQL.*
- Developed Go microservices and gRPC APIs in **droplive-server** — show, drawing, team, and customer services with bulk-edit endpoints, ownership-validation on writes, and per-seller secured **Algolia** key generation. Drove search + index consistency via real-time **Kafka** consumers, and later migrated customer search from Algolia to the primary **droplive** database for accuracy and cost. *Stack: Go, gRPC, Protobuf, MySQL, Algolia, Kafka.*
- Led a team of 4 engineers through legacy schema decoupling: evolved Protobuf schemas in **schema-registry**, added application-level data synchronization between services, and migrated write-path permissions to ownership-validation rather than session-based.

## Amazon

Jul 2020 – Mar 2022

Software Development Engineer

São Paulo, Brazil

- Built REST APIs using Python and AWS Lambda, served via API Gateway and consumed by React applications across multiple countries.
- Developed a Java-based validation engine running on AWS Fargate to validate inbound invoices against vendor documentation.
- Built serverless consumers triggered by SQS to classify inbound documents in Amazon warehouses.
- Set up CI/CD pipelines, integration tests, and monitoring for microservices using AWS CDK.
- Improved developer experience by enabling one-command personal stack deployment using infrastructure as code.

## Yogov

Feb 2020 – Jun 2020

Software Engineer

Oakland, CA

- Developed full-stack services for a web dashboard using React, Node.js, MySQL, and GraphQL.
- Built an integration layer to classify and email clients via Mailchimp APIs.
- Improved SEO through structured data using schema.org.

## BitX

Feb 2019 – Jul 2019

Software Engineer Intern

Belém, PA

- Shipped 3 web applications to manage users and company metrics using React and TypeScript.
- Developed a delivery tracking application using React and Google Maps API.

## Software Engineering Lab

Jan 2018 – Feb 2019

Software Engineer Intern

Belém, PA

- Designed and built a medical web system to manage patient flow at a university clinical lab.
- Developed backend infrastructure using Node.js and MongoDB following RESTful API patterns.

## Personal Projects

---

GitHub Portfolio | *Open Source*

[github.com/Neiva07](https://github.com/Neiva07)

- Public repositories showcasing full-stack and backend work across TypeScript, Go, and Node.js.

## Technical Skills

---

**Programming Languages:** Go, Python, JavaScript, TypeScript, Java, HTML, CSS

**Tools and Frameworks:** React, Node.js, AWS, GraphQL, MySQL, Elasticsearch, CI/CD, React Native, Unix, Redis, MongoDB, Kubernetes, Algolia, Redux

**Languages:** English, Portuguese