

Jonathan Sousa de Jesus

Software Engineer (.NET | Flutter)

Brazil · Open to Remote Work and International Relocation (LATAM / Global)

github.com/jonathan-jesus · linkedin.com/in/jon-jesus · sousjonathan@gmail.com

Profile

Software engineer who progressed from software automation in live broadcast operations to owning the architecture and delivery of business-critical software systems across sales, finance, and field operations.

My work spans financial automation, an AI sales assistant, an offline-first field operations platform, and production automation. The common thread: replacing manual, error-prone processes with maintainable, well-tested software. I do my best work close to operations, where constraints are real and the tradeoffs matter.

Professional Experience

Estância Bahia Leilões — Brazil

Auction house / Agribusiness marketplace

2020–2026

Software Developer | 2024–2026

Owned the architecture, implementation, and delivery of new business-critical software systems across sales, finance, and field operations while integrating with externally maintained legacy platforms.

- Architected and shipped a full-stack field operations platform with **.NET Web API**, **Flutter Web**, and an offline-capable **Flutter mobile** app, replacing manual paper-form registration. Scaled the system from a pilot project to manage 74% of total operations (processing over 900 livestock cataloging sessions, 3,000 lots, and 191,000 animals year-to-date in 2026). Achieved 79% team adoption, drastically reducing manual form transcription and unlocking geographic tracking and operational data analytics.
- Designed and built an internal AI sales assistant using a **FastAPI (Python)** backend and provider-agnostic **LLM orchestration**, enabling sales teams to match products with potential clients and query customer data, credit limits, and purchase history in natural language with real-time **SSE** streaming. Currently in production and under active monitoring ahead of a formal impact review.
- Built bank file-based integrations for electronic payments, eliminating duplicate data entry in the banking portal and manual status updates, significantly reducing operational errors.
- Delivered an integration layer around the core business system using **.NET Web API**, creating a stable boundary for external consumers and enabling integration across multiple internal and third-party applications while isolating them from the underlying legacy platform.

Broadcast Technician | 2020–2024

Software automation and systems integration for time-critical production workflows.

- Developed a **.NET automation tool** for mass video overlay generation, reducing production processing time by **87.5%** during peak digital catalog periods.
- Built secure data-extraction integrations between the auction management system and downstream production systems, eliminating manual data entry.

Projects

Overlayer (2026)

Serverless video processing pipeline on AWS · github.com/jonathan-jesus/overlayer · overlayer.jonsousa.dev

A database-free system that composites custom image overlays onto videos using FFmpeg, with direct S3 uploads and fully asynchronous processing.

- **Architecture:** Event-driven pipeline using **S3** → **SQS** → **AWS Fargate**, with a **.NET 10 Minimal API** on **AWS Lambda** for upload/status operations and a containerized worker for FFmpeg rendering.
- **Frontend:** **Astro** + **React** frontend with an interactive canvas designer.
- **Infrastructure:** **Pulumi** IaC with separate development and production stacks.
- **Engineering:** Implemented **idempotency**, **distributed locking**, **retries**, and **failure handling** across S3/SQS, with Outside-in TDD using **WebApplicationFactory** and **LocalStack**. Documented key architectural decisions with ADRs.

Skills

Mobile: Flutter, offline-first architecture, local persistence, background synchronization

Backend: .NET, ASP.NET Core, Minimal APIs, Entity Framework, FastAPI

Frontend: React, Vue 3, Astro

Languages: C#, Python, Dart, TypeScript, SQL

Cloud (AWS): Lambda, ECS/Fargate, S3, SQS, IAM, S3 Event Notifications

Architecture: Event-driven systems, serverless, distributed locking, idempotency, offline-first

Infrastructure: Docker, Pulumi (IaC), CI/CD

AI / LLM Integration: Tool calling, LLM orchestration, provider-agnostic abstractions, SSE streaming, OpenRouter

Testing: Outside-in TDD, WebApplicationFactory, LocalStack, unit/integration/acceptance testing

Databases: SQL Server, PostgreSQL, MySQL

Education

Computer Science Studies — UFMT (2010–2012)

Technical High School in Computer Science — IFMT (2007–2009)

English: Professional working proficiency.