

**William Elesbão**  
**Full-Stack Developer**

[GitHub](#) | [LinkedIn](#) | E-mail: [william.elesbao.2000@gmail.com](mailto:william.elesbao.2000@gmail.com) | Phone: +55 51 99892-9854

## TECHNICAL SKILLS

- **Programming Languages & Principles:**
  - Node.js · TypeScript · OOP · SOLID · Clean Architecture · Feature-Based Architecture · Atomic Design
- **Front-End:**
  - Next.js · React · Tailwind CSS · Shadcn/Radix UI · Daisy UI · Fluent UI · Aceternity UI · NextUI · Framer Motion · React Hook Form · TanStack Query
- **Mobile Development:**
  - React Native · Expo · Reanimated · Gorhom Bottom Sheet
- **Back-End:**
  - Next.js · NestJS · Elysia.js · Fastify · Prisma ORM · TypeORM · Drizzle ORM · GraphQL · RestAPI · JWT Authentication
- **Databases & Caching:**
  - MySQL · MongoDB · SQL Server · PostgreSQL · Redis
- **DevOps & Infra:**
  - Docker · Docker compose · CI/CD · Drone · Github Actions · EAS Build · Stripe CLI · SigNoz · Jaeger
- **Testing & Code Quality:**
  - Jest · Vitest · Biome · SonarQube
- **API's & Integrations:**
  - OpenAPI · HeyAPI · Swagger · Stripe · Resend · Posthog · Google(account, calendar, maps, etc.)
- **Tools & Workflows:**
  - Git · GitHub · Figma · Builder.io · Bun · Husky
- **Internationalization:**
  - next-intl · i18next · react-i18next

## PROJECTS

- **Leko:** Leko is a React Native mobile application built with Expo that serves as a sports and coaching platform. The app connects users with coaches and enables:
  - User Profiles: Create and manage athlete and coach profiles
  - Booking & Appointments: Schedule training sessions and manage calendars
  - Group Activities: Join sports groups, participate in discussions, and share posts
  - Real-time Communication: Chat with coaches and group members via Pusher WebSocket
  - Payment Integration: Book and pay for coaching services
  - Location Services: Find coaches and activities near you with Google Maps integration
  - Multi-language Support: Available in multiple languages with i18n
    - Technologies used:
      - Mobile Framework: React Native, Expo (Expo SDK 52)
      - Navigation: Expo Router, React Navigation
      - Firebase Cloud Messaging (FCM)
      - State Management: Jotai
      - Data Fetching & Caching: TanStack Query (React Query)
      - Forms & Validation: React Hook Form, Zod
      - Real-time Communication: Pusher WebSockets
      - UI & Performance: Shopify FlashList, Gorhom Bottom Sheet, Reanimated
      - Maps & Location: React Native Maps, Expo Location
      - Authentication & Storage: AsyncStorage, Expo SecureStore
      - Internationalization: i18next, react-i18next
      - Animations: Lottie, React Native Reanimated
      - Device Features: Expo Camera, Image Picker, Haptics, Notifications
      - Analytics: PostHog
      - Testing: Jest, React Native Testing Library
      - Build & Tooling: TypeScript, Bun, EAS Build, Husky, Biome (lint/format), SonarQube
      - API Integration: OpenAPI code generation (openapi-ts)
- **Next Starter:** Development of a modern and scalable full-stack SaaS template using Next.js 16, React 19, TypeScript, and PostgreSQL, featuring authentication, billing, internationalization, email delivery, and Dockerized infrastructure designed for production-ready applications.
  - Framework/Libs/Tools:
    - Next.js16
    - React 19
    - TypeScript
    - Tailwind CSS v4
    - Shadcn/ui
    - Prisma ORM

- PostgreSQL
  - Better Auth
  - Stripe
  - Resend
  - React Email
  - TanStack Query
  - Zod
  - next-intl
  - Biome
  - Jest
  - Docker Compose
  - SonarCloud
  - Drone CI
- **Ignite Starter:** Scalable full-stack SaaS starter designed to build products with a production-ready foundation including frontend, backend, database, authentication, payments, queues, and CI. Features a monorepo architecture, typed OpenAPI client, and local Docker-based infrastructure
  - Framework/Libs/Tools:
    - Next.js16
    - React 19
    - Tailwind CSS v4
    - Shadcn/ui
    - Elysia.js
    - Bun
    - Zod
    - next-intl
    - Drizzle ORM
    - PostgreSQL
    - Better Auth
    - Stripe
    - BullMQ
    - Redis
    - Resend
    - React Email
    - HeyAPI (OpenAPI)
    - TanStack Query
    - Biome
    - Vitest
    - Docker Compose
    - SonarCloud
    - Drone CI
- **Ionic Partners:** Complete development of the institutional website.
  - Technologies used:
    - Builder.io with Figma integration to export designs and generate components using React and Tailwind CSS.
    - Framer Motion for component animations.
    - Next.js, React, Tailwind CSS.
    - Responsive design to ensure compatibility across devices.
    - Resend for sending contact data via email.
- **Event App:** A React Native application for event registration and credential management. Users can register for events, manage credentials, and check in. The backend API is built with Fastify and Prisma for data management.
  - Mobile:
    - react-native
    - expo
    - expo-router
    - axios
    - expo-image-picker
    - @expo/vector-icons
  - Backend:
    - fastify
    - fastify-type-provider-zod
    - prisma
    - @prisma/client
    - Zod
- **VerseMate (Open Source):** A web application offering a simplified and efficient way for users to read and understand the Bible, focusing on AI-generated translations of the books of Genesis and Matthew.
  - Key Features:
    - Full text of Genesis (Old Testament) and Matthew (New Testament) in a clean, navigable interface.

- GPT-generated translations, allowing users to compare the original text with AI-assisted versions.
    - AI conversation storage, enabling users to retain a history of their interactions for future reference.
    - Minimalist and responsive design ensuring usability across devices.
  - Backend Development:
    - Built using Bun and Elysia.
    - Database: PostgreSQL with Kysely.
    - Implementation of secure JWT authentication to ensure data privacy and security.
  - Technologies used:
    - Backend: Bun, Elysia, Kysely, Prisma (ORM).
    - Frontend: Next.js, React, CSS modules.
    - AI: Integration with ChatGPT API for translations and Bible verse explanations.
    - Container: Docker.
- **JokeApp:** A project developed to practice and explore modern web technologies. It features a Nest.js backend with a GraphQL API and a UI built with Next.js using NextUI and AceternityUI.
- **Check In & Out System:** A system developed to track the entry and exit of IT equipment, maintaining records linked to employees.
- **IoT Platform:** Development of an IoT platform with a Next.js frontend using ShadcnUI and NextUI libraries and a NestJS backend consuming WEG APIs via GraphQL services.

## EXPERIENCE

### Frontend Developer

@Deggex

March 2025 – Present

- Defined and documented frontend development standards and best practices, improving code consistency and collaboration across the team.
- Took key technical architecture decisions, migrating the project from Atomic Design to a Feature-Based Architecture after identifying maintainability and scalability issues with the previous structure
- Responsible for ensuring feature parity and stability across both iOS and Android platforms using React Native and Expo.
- Improved code robustness, maintainability, and scalability by introducing better architecture patterns and documentation.
- Implemented Expo OTA (Over-The-Air) updates, reducing mobile deployment costs by 34% and significantly improving the release cycle.
  - Previously, builds were performed weekly with associated build and distribution costs
  - After the OTA implementation, new versions became instantly available to QA as soon as pull requests were merged into the main branch, enabling faster testing cycles and eliminating the need for frequent store builds.
- Integrated PostHog analytics and product telemetry, enabling the team to:
  - Capture runtime errors and behavioral insights
  - Track user interaction metrics and feature usage
  - Improve data-driven decision making for product improvements
- Implemented type-safe API integration using HeyAPI (OpenAPI code generation) to automatically generate API clients based on backend specifications.
- Designed and implemented CI/CD pipelines to automate the mobile development workflow, including:
  - Automated build validation and testing during pull requests
  - Automated deployment workflows integrated with Expo/EAS
  - Continuous integration to ensure code quality, build stability, and faster release cycles
- Set up and maintained the mobile development environment, including integrations with third-party services such as Google Calendar, Apple Calendar, Google Maps, and Apple Maps.
- Collaborated with backend and design teams to ensure accurate implementation of product requirements and scalable frontend architecture.

### Full-Stack Developer

@Apegro

September 2024 – March 2025

- Responsible for the full-stack development of the Ionic Partners institutional website;
- Responsible for the full-stack development of VerseMate (MVP | Open Source), an interactive Bible reading application featuring AI-assisted translations and explanations.
  - Key contributions: Database design (PostgreSQL), modular and scalable backend using Elysia, integration with ChatGPT API, and implementation of secure JWT authentication.

### Full-Stack Developer

@TMSA

December 2021 – March 2025

- Analyzed processes and developed BPMN workflows and forms using SoftExpert.
  - Developed server-side rendered (SSR) and static site generated (SSG) applications with Next.js.
  - Built scalable and modular server-side services with Nest.js.
  - Implemented RESTful and GraphQL APIs using Nest.js and TypeScript, including token-based authentication.
  - Documented and tested APIs with Jest and Swagger.
  - Integrated internal, external, and cloud systems, including Senior ERP, Microsoft Azure, and SoftExpert.
  - Developed and integrated IoT platforms.
  - Optimized code for maintainability and scalability.
  - Performed unit, integration, and end-to-end testing.
  - Stayed updated on the latest technologies including Next.js, Nest.js, TypeScript, Clean Architecture, and SOLID principles.
- 

## **EDUCATION**

### **Bachelor's Degree in Systems Analysis and Development**

@Unicesumar

2022 – 2024

---

## **LANGUAGES**

- English: Intermediate