Sebastián Arduz

Ingeniero Full-Stack (TypeScript) · Automatización, integraciones y sistemas de outbound

Sucre, Bolivia (UTC-4, horario de EE.UU. este) · Remoto · Disponible desde noviembre de 2026

jsebastianarduzespa@gmail.com · +591 77124398 · linkedin.com/in/sebastian-arduz-46863a265 · github.com/Sebillasxd2 · sebillasxd2.github.io/PortafolioWeb

Resumen

Ingeniero full-stack en TypeScript. Durante un año fui el ingeniero interno y responsable de automatización de una agencia de ventas outbound B2B en EE.UU., trabajando junto a su fundador técnico: 80 pull requests mergeados a un monorepo en producción (Next.js, Prisma, PostgreSQL y Trigger.dev), 13 monitores programados con alertas por Slack y email, arreglos de causa raíz publicados en producción, y la capa de automatización sobre GoHighLevel, Instantly y Microsoft 365. Uso Claude Code a diario; nada se mergea sin tests de guarda, una revisión independiente y mi propia lectura del diff. Antes, Java y Spring Boot. Español nativo, inglés C1, UTC-4, disponible para horario de EE.UU. o Europa.

Experiencia

Ingeniero Full-Stack (TypeScript) · Leveraged Operations (Local GTM), EE.UU., contrato remoto

dic 2025 - oct 2026

Ingeniero interno y responsable de automatización, trabajando junto al fundador técnico, en una operación de ventas outbound (cold email en Instantly, llamadas en frío de SDRs en GoHighLevel, generación de leads) para negocios de servicios locales en EE.UU. De dic 2025 a jun 2026: integraciones de clientes y trabajo de datos, infraestructura de email en Microsoft 365 y pipelines de leads. De jul a oct 2026: 80 PRs mergeados a la plataforma central, 114 commits en total.

- **80 PRs mergeados al monorepo en producción en 114 commits (jul a oct 2026)**: Next.js 16 (App Router) / React 19, API con oRPC, Prisma sobre PostgreSQL (Supabase), jobs en Trigger.dev; 19 migraciones SQL y 16 archivos nuevos de tests de guarda en un código de 306 procedimientos oRPC y 227 modelos Prisma. Merge a cargo del owner hasta sep 2026, luego con permiso para mergear mis propios PRs bajo el mismo filtro (Biome, type-check, build y una revisión con IA independiente antes del merge), confirmando cada deploy contra el SHA en vivo de /api/version.
- **Reconstruí el chat base del fundador en un chat estilo Slack en 16 PRs en 5 días**: DMs, @menciones, reacciones, respuestas citadas, pegado de imágenes, notificaciones in-app y del sistema operativo, paleta de avatares WCAG-AA; después agregué presencia y avisos por email al manager ante un mensaje nuevo.
- **Mejoré el flujo de aprobación de leads de los SDRs en 8 PRs mergeados en un día** (3.843 líneas, 74/74 tests de guarda): primeros borradores con IA con guard de placeholders, presets de motivo de rechazo con prefill de IA, estadísticas por rep a 7 días, tareas de rellamada automáticas a 3 días.

- **Escribí 13 monitores programados en Trigger.dev con alertas por Slack y email**: un guardrail de rebotes (avisa sobre 3% con 200+ envíos en 7 días, informa y nunca pausa solo), caída del reply rate, reserva de leads, campañas vivas sin leads, clientes que pagan sin fulfillment, y chequeo nocturno de readiness del dialer.
- **Diagnostiqué incidentes de producción y publiqué arreglos con tests, varios mergeados el mismo día**: colas del dialer que acumulaban leads de otros clientes entre turnos (10 caminos de fuga confirmados por una auditoría de 36 agentes sobre mi propia rama, con la cola de un rep abarcando 14 clientes), arregladas el mismo día con un gate que no carga hasta verificar la cola vacía, un job nocturno que abortaba cada noche pero se reportaba como COMPLETED durante cuatro semanas, 127 dominios renovados marcados falsamente como vencidos (ahora contrastados con el registro vía RDAP), y un aviso de caller-ID reconstruido para Local Presence Dialing y validado contra registros de llamadas reales (6.882 evaluaciones, 0 discrepancias).
- **Aprovisioné la infraestructura de cold email**: 9 tenants de Microsoft 365 (unos 225 buzones compartidos) con SPF/DKIM/DMARC por la API de Cloudflare, con 26 scripts de PowerShell y 2 bots de Playwright (creación de buzones, conexión OAuth con Instantly al 97% de éxito al primer intento); audité 497 dominios y 2.646 IPs contra listas negras de DNS y demostré que el problema estaba en los dominios, no en las IPs de envío; purgué 46.192 contactos sin respuesta en 42 campañas por API, con backup a CSV de las campañas que valía conservar y verificando que cada una quedara con exactamente sus 1.538 respondedores.
- **Construí el pipeline de leads de punta a punta, del scraping a la campaña en borrador**: scraping con D7 y Apify con topes de costo, normalización y deduplicación, una cascada de verificación (MillionVerifier, luego una pasada SMTP con Reacher que agregué tras comprobar que la pasada única dejaba pasar 57 de 64 rebotes reales, luego Reoon), filtrado de alcance con LLM, y campañas en borrador creadas por la API de Instantly; 22.728 contactos geo-limpios en 29 builds para 32 clientes, 27 de 29 pasando un spot-check de 40 filas con controles.
- **Herramientas de campañas y respuestas sobre datos vivos de Instantly**: un linter de copy sobre campañas vivas y en borrador de Instantly (13 gates, con el spintax renderizado como lo ve el prospecto) más una pasada de 17 gates sobre los masters de copy antes de subirlos; el primer barrido marcó 71 borradores con 498 fallos y descubrió un bug de la API de Instantly que borraba en silencio los cuerpos formateados solo con etiquetas br), un tablero de escalación para respuestas de cold email, y una revisión de 90 días (302 campañas, 378.605 emails, 4.063 respuestas, con un LLM clasificando una muestra de 680 respuestas) que mostró que el ángulo de referral generó 39 de 48 oportunidades.
- **Endurecí endpoints públicos y de pago**: cuotas diarias por organización y un circuit breaker global en los fan-outs de jobs de pago, rate limits y topes de entrada en rutas públicas, headers de seguridad (HSTS, nosniff, X-Frame-Options) y un login por passcode con rate limit.
- **Ingeniería con IA usando Claude Code**: skills propias (un revisor de copy de cold email con un harness de regresión de 9 casos, evaluado contra 365 campañas reales ordenadas por resultados; una skill de triaje de respuestas positivas de clientes) y 62 scripts de operación; workflows multiagente para cuando una sola pasada del modelo no alcanza (scoping de leads con 10 agentes más 2 auditores, QA de copy con 24 jueces, auditoría del dialer con 36 agentes); y un índice del monorepo que redujo a la mitad el archivo de arranque del agente (de 921 a 463 líneas).
- **Soporte técnico al equipo de SDRs**: aprovisionamiento del dialer en GoHighLevel con permisos de mínimo privilegio, números locales y campos personalizados para cada cohorte nueva, aprobador designado de las colas de leads y emails (el gate de los leads pagables), screening de candidatos del embudo de contratación, y 10 runbooks y SOPs.

Ingeniero Full-Stack (Node.js) · Chatter Champions (agencia de talento de creadores), contrato remoto, en paralelo con el puesto anterior

may 2026 - ago 2026

Embudo de contratación y portal de capacitación para una fuerza de trabajo remota de soporte por chat: Express, PostgreSQL (Supabase) con dual-write a Google Sheets, Airtable, GoHighLevel, Trigger.dev, xAI Grok, Resend, Telegram, Dokploy.

- **83 PRs mergeados (100 commits en main, +11.3k líneas, 64 ramas)** en features de producto, automatizaciones y herramientas de administración, bajo un flujo de PRs con main protegida.
- **Resolví el incidente detrás de un día con 5 de 8 ausencias a entrevistas**: los emails de recordatorio de GoHighLevel llevaban links de Google Meet rotos. Creé el campo personalizado que faltaba por la API, rellené 28 reservas y publiqué el arreglo en Trigger.dev (sincronización del meeting-link más una regla de una reserva por contacto).
- **Construí el simulador de entrevista con IA y la capa de integridad de entrevistas**: chat de solo captura con tope de costo, revisión del transcript con decisión de contratar o no, invitaciones al entrevistador, jobs de recuperación de drop-off y de abandono en Trigger.dev; seguimiento de re-agendadores en serie, detección automática de ausencias con un resumen diario, baneo automático de postulantes rechazados o ausentes, blacklist en el momento de la reserva, y un screen-out de ingreso lanzado como test A/B.
- **Herramientas de administración y RR.HH.**: rol de manager con escrituras controladas, vista de contratados por fecha de contrato con badges de atraso, vista de completitud de trainees, alertas diarias de RR.HH., sincronización con Airtable, y un editor de plantillas de turnos que genera cualquier cantidad de semanas.

Desarrollador de Software Freelance (Java / Spring Boot) · Sucre, Bolivia

nov 2024 - nov 2025

- **Sistema de gestión para una concesionaria de autos (CrediAutos, feb a abr 2025):** app de escritorio en JavaFX sobre Spring Boot (Spring Data JPA) y MySQL: login de administrador y CRUD completo de clientes, vehículos, proformas, registros de mantenimiento y recordatorios, con proformas imprimibles. Repo: github.com/Sebillasxd2/ConcesionariaVehiculos.
- **Sistema de registros para un consultorio médico privado (nov 2024 a ene 2025):** app de escritorio en JavaFX sobre Spring Boot (Spring Data JPA) y MySQL: registros de pacientes, historia clínica, signos vitales y búsqueda. Repo: github.com/Sebillasxd2/ConsultorioMedico (subido como un único snapshot en abril de 2025).
- También: CajaVentas, un punto de venta en Python/SQLite (2025, repositorio privado).

Educación y certificaciones

- **Ingeniería de Sistemas (en curso, 8vo semestre)**, Universidad San Francisco Xavier de Chuquisaca, Sucre · 2021 - actualidad. El plan actual incluye un curso de IA basado en PyTorch.

- **Oracle Next Education F2 T6, pista de Back-end** (Oracle + Alura) · 2024 · Java, Spring Boot, REST APIs, bases de datos.
- **Inglés C1**, Centro Boliviano Americano (CBA), graduado del programa completo · 2020.

Habilidades

- **Lenguajes:** TypeScript / JavaScript (Node.js), SQL, Java, Python, PowerShell, Bash
- **Backend y datos:** Next.js 16 (App Router), REST APIs, oRPC (sobre Hono), Express, Prisma, PostgreSQL (Supabase, PostgREST), MySQL, Spring Boot, jobs en segundo plano y tareas programadas (Trigger.dev), webhooks, integraciones OAuth, pipelines de datos (ETL, deduplicación, verificación)
- **Frontend:** React, Tailwind CSS, shadcn/ui, UI responsive, contraste de color WCAG-AA
- **Sistemas de outbound y GTM:** infraestructura de cold email (tenants de Microsoft 365 / Exchange Online, aprovisionamiento de buzones, warm-up, SPF/DKIM/DMARC, auditorías DNSBL, análisis de placement en inbox), Instantly API v2 (creación de campañas, movimientos y purgas de leads, webhooks), cascada de verificación de emails (MillionVerifier, Reacher, Reoon), armado de listas y scoping de ICP (D7, Apify, Clay, filtro de alcance con LLM), clasificación y ruteo de respuestas, QA de copy y linting de spintax, CRM y power dialer de GoHighLevel (alta de usuarios, diseño y auditoría de workflows, campos personalizados, detección de deriva de caller ID con Local Presence Dialing), y automatizaciones con n8n al inicio del rol, antes de que ese trabajo pasara a Trigger.dev
- **Ingeniería con IA:** Claude Code (skills, hooks, workflows multiagente, rutinas programadas), APIs de Anthropic y OpenAI, xAI Grok, prompt engineering, evaluaciones con LLM como juez
- **Práctica de ingeniería e infraestructura:** flujo de PRs en Git / GitHub, code review, CI/CD (GitHub Actions), tests de guarda automatizados, Biome, monorepos con pnpm, Playwright, Dokploy (deploys basados en Docker), Nginx, VPS Linux (Hetzner), API de Cloudflare, Airtable, APIs de Google Sheets / Drive, Resend, Slack, Telegram, monitoreo y alertas de producción, respuesta a incidentes, documentación técnica (SOPs, runbooks)