Argentina
gaellgonzalez10@gmail.com
+54 9 291 419 3031
linkedin.com/in/gael-gonzalez-2b0887273
github.com/gael689
gaelgonzalez.com.ar

GAEL GONZÁLEZ

Desarrollador de Software Full Stack

PERFIL PROFESIONAL

Desarrollador full-stack responsable del ciclo completo de más de quince plataformas web —y una aplicación móvil (Android/iOS) en curso— hoy en uso real, para clientes de doce rubros distintos: desde la venta y el relevamiento con el cliente hasta la arquitectura, el desarrollo, el despliegue y el mantenimiento. Trabajo las tres capas, aplicando arquitectura limpia y patrones de diseño en todas: interfaces en React, Next.js y React Native; backend en FastAPI y Node.js, con modelado de datos, APIs REST e integración de servicios de terceros; e infraestructura, con contenedores Docker, integración continua y despliegue en plataformas administradas (Railway, Vercel). Actualmente lidero la arquitectura de Plenia, una plataforma multi-tenant de gestión de seguros integrada con las APIs de doce aseguradoras argentinas e internacionales. Formo y coordino equipos de desarrollo, trabajo tanto sobre sistemas propios como sobre bases de código existentes, y tengo amplia experiencia sosteniendo la relación comercial y técnica con cada cliente. Incorporé el desarrollo asistido por IA al flujo diario, con formación continua y criterio para elegir la herramienta según la tarea, revisando yo mismo todo lo que llega a producción.

EXPERIENCIA

Desarrollador de Software Full Stack

Independiente (freelance, clientes propios) · Argentina · noviembre 2025 – Presente

- Diseñé y desarrollé más de quince plataformas web a medida —más una aplicación móvil (Android/iOS) en curso— para clientes de doce rubros distintos (seguros, gestión, RRHH, inmobiliario, industrial, comercio electrónico), aplicando Clean Architecture y principios SOLID para reducir la deuda técnica y sostener el mantenimiento en el tiempo.
- Construí las interfaces en React y Next.js (App Router, Server Components, renderizado en servidor) con TypeScript, Tailwind CSS y shadcn/ui sobre Radix UI, y las aplicaciones móviles en React Native con Expo — diseño responsivo, accesibilidad, estado con TanStack Query y Zustand, y animación con Framer Motion y GSAP.
- Desarrollé los backends en FastAPI y Node.js: APIs REST, autenticación y autorización basadas en roles (RBAC) con OAuth2, OIDC, JWT y Clerk, y webhooks para integrar servicios de terceros.
- Modelé bases de datos relacionales en PostgreSQL, MySQL y SQLite con foco en integridad referencial, indexación, transacciones y migraciones, y trabajé sobre Supabase (Postgres administrado, Auth, Storage y Row-Level Security) aplicando control de acceso por rol a nivel de base de datos, para sostener múltiples perfiles de usuario sobre una misma plataforma.
- Integré pasarelas de pago (Mercado Pago) y la API de Mercado Libre, con webhooks que disparan la entrega de contenido digital por email sin intervención manual, y construí pipelines de automatización con scraping (BeautifulSoup), normalización y deduplicación de catálogos de proveedores.
- Apliqué estrategias de caché e invalidación, optimicé consultas y sostuve el rendimiento en producción —compresión de imágenes en la subida con sharp, carga diferida y dimensionado responsivo— para catálogos de hasta 125.000 productos con accesos concurrentes.
- Escribí pruebas automatizadas —unitarias y de integración con Vitest y Testing Library, end-to-end con Playwright— sobre flujos críticos como el checkout y la autenticación.
- Desarrollé paneles de administración a medida para que el cliente publique productos, propiedades, búsquedas laborales y contenido sin intervención técnica.
- Construí sitios bilingües (español/inglés) con enrutamiento internacionalizado (next-intl) y mapas interactivos con agrupamiento de marcadores (Leaflet, Google Maps) para catálogos geográficos de más de 50 ubicaciones.
- Desarrollé formularios con validación robusta (React Hook Form + Zod) y anti-spam sin CAPTCHA —campo trampa invisible y límite de envíos por IP—, sin fricción para el usuario real.
- Implementé SEO técnico en todos los proyectos —metadatos por página, datos estructurados (schema.org), sitemaps y arquitectura de información por intención de búsqueda—, para buscadores tradicionales y con IA.
- Conduje el análisis previo al desarrollo con cada cliente: elicitación de requerimientos, glosario de dominio, definición de requerimientos no funcionales (rendimiento, escalabilidad, disponibilidad, seguridad) y modelo de datos, contemplando el cumplimiento de la normativa de protección de datos personales (Ley 25.326).
- Definí e instrumenté las métricas del negocio: tasa de respuesta por gancho, plantilla, rubro y horario, embudo por etapa y conversión hasta el ingreso facturado, bajo una única regla de atribución, así cada número responde a una sola pregunta y se puede comparar entre campañas.
- Conseguí por mi cuenta la totalidad de los clientes —prospección, primer contacto, propuesta, negociación y cierre—, sin intermediarios ni área comercial, y construí un sistema propio de captación y CRM para sostener ese trabajo.
- Formé y lideré equipos de trabajo para mis proyectos: delegué tareas de desarrollo, definí prioridades, revisé cada entrega por code review y organicé el trabajo con metodologías ágiles (Scrum/Kanban) — sprints cortos, backlog priorizado y entregas incrementales revisables por el cliente.
- Asumí el mantenimiento evolutivo del sistema de gestión interno de un cliente distribuidor, trabajando sobre una base de código existente en lugar de forzar un reemplazo.
- Incorporé el desarrollo asistido por IA al flujo diario desde abril de 2026, con Claude Code como herramienta principal y Gemini y Antigravity según la tarea: archivos de contexto por repositorio (CLAUDE.md/AGENTS.md) con la arquitectura y las reglas que no se pueden romper, skills propias que automatizan trabajo repetido, y varios agentes en paralelo sobre worktrees de git separados, con una regla de CI que falla si baja la cantidad de tests. También uso IA dentro de los productos: mi CRM redacta mensajes con la API de Claude, y estoy diseñando una arquitectura RAG para recomendar candidatos en el portal de empleo. Reviso yo mismo todo lo que llega a producción.

PROYECTO PRINCIPAL

Plenia — Plataforma de Gestión de Seguros

Arquitecto de software y líder técnico · Argentina · 2026 – Presente

Producto desarrollado en conjunto con un grupo de brokers de seguros, sobre necesidades reales del negocio

> **Doce aseguradoras integradas:** Experta Seguros · San Cristóbal Seguros · San Cristóbal ART · Sancor Seguros · La Holando Sudamericana · Seguros Rivadavia · Zurich Argentina · Integrity Seguros · MAPFRE Argentina · Meridional Seguros · Mercantil Andina · Allianz Argentina

- Lidero el proyecto completo —arquitectura, backend, frontend, mobile y testing— de una plataforma multi-tenant que maneja unas 2.400 pólizas y 1.600 clientes, en web, iOS y Android: backend en Python/FastAPI con SQLAlchemy, PostgreSQL y Redis; cuatro interfaces web según el perfil de usuario (administrador, asesor, cliente y profesional —contador o escribano con acceso de sólo lectura—) en Next.js 16 y React 19; y una aplicación móvil en React Native/Expo, con una sola base de código para las dos tiendas. El proyecto está en su etapa final, antes de la publicación en las tiendas. Coordino a un desarrollador colaborador: reparto tareas y reviso todo el código.
- Diseñé la capa de integración bajo un patrón adapter con capa anticorrupción (Anti-Corruption Layer) y un registry por proveedor: sumar una compañía es escribir su adapter y registrarlo, sin tocar el núcleo. Cada adapter declara las operaciones que su API soporta de verdad, y la interfaz habilita cada función sola. Cubre REST/JSON, SOAP/XML y OpenAPI —una suite de 17 especificaciones—, con un esquema de autenticación distinto por proveedor (OIDC, OAuth2, JWT, API key), credenciales cifradas por compañía y por ambiente, y una fuente de verdad definida para cada dato, así ninguna sincronización pisa información que le pertenece a la plataforma.
- Construí el multicotizador: una sola consulta va en paralelo a todas las compañías habilitadas, cada una con su propio timeout para que un proveedor lento no demore al resto, y vuelve como una tabla comparable. Reemplaza entrar a un portal distinto por cada compañía.
- Diseñé la caché de catálogos de las compañías después de medir 200 de 200 pedidos rechazados en una mañana por un proveedor que responde 429 a partir del tercer pedido seguido: un fallo nunca pisa un valor bueno, una respuesta vacía no se toma como éxito, cada entrada registra cuándo se trajo, los pedidos simultáneos disparan una sola llamada, y la caché se persiste y se precarga al arrancar.
- Endurecí las operaciones que comprometen dinero: exigen permisos explícitos, corren contra el ambiente de pruebas por defecto, y su payload se arma desde el registro guardado en el servidor y no desde lo que envía el cliente — así ningún usuario puede operar sobre el recurso de otro.
- Elegí Railway como PaaS administrado —backend en Docker, PostgreSQL, Redis y el frontend de Next.js como servicios aparte— porque las compañías exigen IPs de salida en una lista blanca y sus IPs estáticas lo cumplen; el VPS propio que se había planificado se descartó al leer el formulario real de la compañía. El CI en GitHub Actions corre unos 6.000 tests automáticos más typecheck y build, con un piso que hace fallar el build si baja la cantidad de tests, y la aplicación no arranca en producción si falta una variable obligatoria.

PROYECTOS DESTACADOS

Una selección de diez, sobre más de quince trabajos entregados. El portfolio completo, con capturas y el detalle de cada uno: **gaelgonzalez.com.ar**

Centro de Operaciones + leadgen · **2026, en uso diario** — Sistema propio de captación y CRM: 24 módulos sobre un servidor HTTP en Python sin dependencias externas y una interfaz propia sin framework. Un worker prospecta negocios, audita sus sitios, redacta los mensajes con IA y los agenda en Resend, lee las respuestas por IMAP y sincroniza los estados solo. Cubre embudo y seguimiento de clientes, campañas y mensajes automáticos, cotizaciones con propuesta en PDF, tareas y pendientes, llamadas y WhatsApp asistidos, equipo, reportes y monitoreo de salud del sistema, todo sobre SQLite.

Omega Distribuciones · **2026, publicado** — E-commerce (omegadistribuciones.com.ar) para una distribuidora de autopartes, repuestos agrícolas y ferretería que revende de seis mayoristas: unifica sus catálogos en unos 125.000 productos (unos 26.000 disponibles para comprar), un motor de puntaje tipo buy box elige a qué proveedor comprarle cada pieza, los precios incluyen la comisión de Mercado Libre consultada en vivo, y la tienda corre sobre una caché compartida en Redis.

Sistema MBK · **abril 2026** — Plataforma de gestión financiera con dos roles conectados, permisos por rol y reglas de seguridad a nivel de base de datos.

BBJobs · **junio–septiembre 2026, en línea desde julio** — Portal de empleo regional (bbjobs.com.ar) con tres roles; a septiembre de 2026, 1.611 candidatos, 10 empresas verificadas y 2.445 postulaciones. Las empresas se verifican antes de poder ver candidatos, cada una ve sólo a quienes se postularon a sus propias búsquedas, y una base de talento anónima se desbloquea con créditos.

Julián Tavano Arquitecto · **julio 2026** — Estudio de arquitectura bilingüe (español/inglés) con catálogo de modelos prefabricados y mapa interactivo de más de 50 obras construidas.

Syrius Ingeniería SRL · **agosto 2026** — Sitio institucional para el representante oficial de Emerson, con catálogo técnico reorganizado por familia de producto.

Neuhaus S.A. · **agosto 2026** — Industria gráfica del sector farmacéutico y cosmético (ISO 9001, BPM y FSC), con formularios de cotización específicos por tipo de producto.

NG Consultoría · **febrero 2026** — Motor de evaluación psicométrica propio, con lógica de scoring desarrollada a medida.

JLI Recruiters · **abril 2026** — Plataforma de RRHH con panel administrativo, permisos multiusuario y venta de contenido automatizada vía pasarela de pago.

Alport JF · **julio 2026** — Vidriera digital de siete rubros con carrito, ofertas con vencimiento automático y actualización de precios en lote desde el panel.

HABILIDADES TÉCNICAS

Lenguajes: TypeScript · JavaScript (ES6+) · Python · Java · SQL · C · Prolog · HTML5 · CSS3

Frameworks y librerías: Next.js (App Router, Server Components, SSR/ISR) · React · React Native · Expo · FastAPI · Node.js · SQLAlchemy · Tailwind CSS · shadcn/ui · Radix UI · TanStack Query · Zustand · React Router · React Hook Form · Zod · Framer Motion · GSAP · Three.js · Recharts · Leaflet · next-intl · Vite

Backend y APIs: Diseño de APIs REST · SOAP/XML · OpenAPI/Swagger · Webhooks · Uvicorn · Alembic · Autenticación y autorización por roles (RBAC) · Tareas programadas y procesamiento en segundo plano · Rate limiting · Estrategias de caché e invalidación · IMAP/SMTP

Bases de datos: PostgreSQL · MySQL · SQLite · Supabase · Redis · Modelado de datos relacional y no relacional (criterio para elegir el tipo de base según el problema) · Indexación y optimización de consultas · Transacciones · Migraciones · Row-Level Security

Datos y automatización: pandas · Web scraping (BeautifulSoup, lxml) · Normalización y deduplicación de catálogos · Pipelines de importación y enriquecimiento · Definición y medición de métricas · Análisis de embudo, atribución y conversión · Exportación a Excel (openpyxl) · Motores de reglas y de scoring propios

Infraestructura y DevOps: Docker · GitHub Actions (CI/CD) · Railway · Vercel · Cloudflare R2 · Linux · DNS y dominios (apex/www, certificados) · Configuración y secretos por ambiente · Despliegue con migraciones automáticas · Validación de configuración al arranque

Seguridad: OAuth2 · OIDC · JWT · API keys · Clerk · Supabase Auth · Control de acceso por rol (RBAC) · Row-Level Security · Cifrado de credenciales por ambiente · CORS · Rate limiting · Políticas de contraseñas · Protección anti-spam

Servicios de terceros integrados: Mercado Pago · API de Mercado Libre · Clerk · Supabase · Cloudinary · Cloudflare R2 · Resend · EmailJS · Google Maps y Google Places · API de Anthropic (Claude) · APIs de doce aseguradoras argentinas e internacionales (REST, SOAP y OpenAPI) · Portales B2B de seis proveedores mayoristas

Testing y calidad: Vitest · Testing Library · Playwright (end-to-end) · Pruebas unitarias y de integración · ESLint · Prettier · TypeScript en modo estricto · Code review

Arquitectura y diseño: Clean Architecture · Principios SOLID · Patrones de diseño (Factory, Strategy, State Machine, Observer, Adapter) · Anti-Corruption Layer · Arquitectura multi-tenant · Modelado de dominio · Arquitecturas distribuidas

Prácticas de frontend: Diseño responsivo · Accesibilidad · Internacionalización (i18n) · Optimización de carga y de imágenes · SEO técnico y datos estructurados (schema.org)

Análisis y modelado: Elicitación de requerimientos · Requerimientos no funcionales · Verificación y validación · Diagramas de entidad-relación (ER), de clases, de casos de uso, de estados y de secuencia · Protección de datos personales (Ley 25.326)

Fundamentos: Programación orientada a objetos · Programación funcional y lógica · Estructuras de datos y algoritmos · Análisis de complejidad · Programación concurrente y asíncrona (procesos, hilos, semáforos y mutex) · Sistemas operativos

Ingeniería de software: Git · GitHub · Metodologías ágiles (Scrum/Kanban) · Documentación técnica de decisiones · Desarrollo asistido por IA (Claude Code, Gemini, Antigravity)

HABILIDADES BLANDAS

Venta y relación comercial: la parte comercial la hago yo: prospección, presentación, cotización, negociación de alcance y precio, y cierre — sin intermediarios ni área comercial. Más de quince trabajos vendidos así, con clientes que volvieron a contratarme para un segundo proyecto y una relación que sostengo después de la entrega.

Comunicación y oratoria: es una habilidad que trabajo a propósito, no algo que di por dado: hice un curso de oratoria y lo sostengo con lectura continua. Es lo que me permite explicar una decisión técnica a alguien que no es del rubro y presentar yo mismo cada propuesta.

Traducción entre negocio y técnica: conduzco el relevamiento con gente no técnica, escribo las propuestas sin jerga y entrego paneles pensados para que el cliente opere solo, sin depender de mí en el día a día.

Liderazgo y autogestión: formé y coordiné equipos para mis proyectos —reparto de tareas, prioridades y revisión de cada entrega por code review—, sosteniendo la actividad profesional en paralelo a la carrera de grado.

Interlocución con equipos externos: sostengo la relación técnica con los equipos de ingeniería y soporte de doce aseguradoras: relevamiento, homologación en sus entornos y resolución de incidencias.

Apertura a la crítica: cuando escribo una decisión técnica dejo marcado qué medí y qué estoy suponiendo, y la doy a revisar antes de ejecutarla. Prefiero que un supuesto mío se caiga en la revisión y no en producción — ya me pasó, y me ahorró rehacer una infraestructura entera.

Compromiso con la entrega: acuerdo el alcance por escrito antes de arrancar y entrego eso, completo — no una versión recortada para llegar a la fecha. El trabajo se da por cerrado cuando el cliente está usando el sistema y conforme con el resultado, no cuando el código quedó subido.

IDIOMAS

Español: Nativo. **Inglés:** Nivel intermedio — lectura y escritura técnica fluida; examen de suficiencia de inglés de la UNS aprobado.

FORMACIÓN ACADÉMICA

Ingeniería en Sistemas de Información

Estudiante avanzado

Universidad Nacional del Sur (UNS) — Argentina

Materias aprobadas hasta la fecha: Algoritmos y Complejidad · Arquitectura y Diseño de Sistemas · Bases de Datos · Métodos Formales para Ingeniería de Software · Requerimientos de Sistemas · Lógica para Ciencias de la Computación · Arquitectura de Computadoras · Modelos de Software · Modelos Estadísticos para Ciencias de la Computación · Tecnología de Programación · Organización de Computadoras · Teoría de la Computabilidad · Estructuras de Datos · Introducción a la Ingeniería de Software · Lenguajes Formales y Autómatas · Introducción a la Programación Orientada a Objetos · Resolución de Problemas y Algoritmos, además de las materias de ciencias básicas del ciclo.

Proyectos académicos con código público: *Mario Bros* (Java) — videojuego 2D sobre patrones de diseño: máquina de estados para personaje y enemigos, fábricas de entidades y sistema de niveles desacoplado del motor. *M2Blocks* (Prolog + React/TypeScript) — motor de reglas declarativo conectado a una interfaz imperativa por contrato. Ambos en github.com/gael689.