Rodrigo Amarilla

Formosa, Argentina
+54 9 370 4516054 • amarillarodrigo14@gmail.com
LinkedIn, GitHub, rodrigoamarilla.tech

Backend Engineer con foco en cloud engineering. Diseño y mantengo APIs REST y microservicios en producción, y los despliego en AWS con Kubernetes, Terraform y CI/CD, con observabilidad de trazas, métricas y alarmas. Trabajo con Java y Spring Boot, Node.js y TypeScript, Python y FastAPI, PostgreSQL y Docker, y TDD estricto.

Competencias Técnicas

- **Backend:** Java 25, Spring Boot 3.5, Spring Security, Node.js, TypeScript, Fastify, Express, NestJS, FastAPI (Python), APIs REST, OpenAPI 3.1
- **Datos:** PostgreSQL 15/16, MySQL, MongoDB, Supabase, Prisma, Flyway, SQLAlchemy, Redis
- **Cloud e infraestructura:** AWS (EC2, EKS, VPC, IAM, ECR, S3, Secrets Manager, SSM, CloudWatch, X-Ray, CloudTrail, AWS Config, SNS), Kubernetes, k3s, Terraform, OpenTelemetry, GitHub OIDC, Docker, Docker Compose, Ansible, Nginx, Railway, Render
- **IA y análisis:** SGLang (XGrammar, RadixAttention), inferencia de LLMs con salida estructurada, Gemini, pandas, scraping con Cheerio
- **Calidad y herramientas:** TDD (JUnit 5, Mockito, pytest), Git, runners self-hosted, OpenSpec, Linux, PowerShell

Experiencia Profesional

Outlier AI — **Remoto**
Revisión de calidad de salidas de IA — *2026*

- Revisión y puntuación de respuestas de modelos de lenguaje para clientes de IA generativa, sobre cinco ejes: calidad, contexto, utilidad, veracidad y potencial de daño. En mi proyecto de Journal API implemento el mismo análisis con salida estructurada validada contra esquema y defensa contra prompt injection.

NÓRDICO — **Formosa, Argentina**
Backend Engineer — *2026-07 - 2026-10*

- Implementé el microservicio de autenticación en Java 25 y Spring Boot 3.5: login unificado con JWT (HS256), rotación de refresh token, BCrypt cost 12, OAuth2 resource server, rate limiting y control de acceso por cinco roles.
- Modelé el esquema relacional en PostgreSQL 15 con migraciones versionadas en Flyway, incluida la regla de arranque del primer administrador de IT.
- Desplegué y mantuve APIs de pedidos, equipos, comodatos y visitas, con evidencia fotográfica almacenada en S3 mediante AWS SDK v2.
- Endurecí la API en producción: mapeé errores 4xx que devolvían 500, optimicé el listado de pedidos y corregí un fallo por `lower(bytea)` en PostgreSQL.
- Despliegue con imágenes Docker multi-stage, healthcheck contra el puerto de la plataforma y CORS por entorno, en un equipo de tres desarrolladores.
- Documenté la API con OpenAPI 3.1 y expuse actualizaciones en tiempo real con WebSocket.

Instituto Politécnico Formosa — **Formosa, Argentina**
Desarrollador Backend (Prácticas Profesionales) — *2025*

- Tres prácticas profesionales en proyectos de clientes, con un promedio de 9.66 sobre 10: desarrollo y despliegue de backends, modelado de datos e integración con frontends.

Proyectos Independientes

- **Python, FastAPI, PostgreSQL, AWS, Terraform, Kubernetes, EKS, OpenTelemetry**
  Journal API - Capstone de Cloud Engineering *2026-10*
  - Infraestructura como código con 79 recursos de Terraform en AWS (VPC, EC2, EKS, IAM, S3 con KMS, Secrets Manager, CloudTrail, SNS, CloudWatch); las credenciales se siembran fuera de banda y nunca están en el repositorio ni en el estado de Terraform.
  - Identidad de servicio propia: rol de runtime con permisos sobre dos ARN exactos, `Deny` explícito de lectura de secretos para el rol de la instancia y `ExternalId` atado al ServiceAccount del pod.
  - Tres fallas de producción resueltas hasta la causa raíz: un ConfigMap que se actualizaba sin que el proceso lo leyera (inodo y *swap* del symlink `..data`), un plan de Terraform que proponía reiniciar producción por finales de línea CRLF, y una alarma que disparaba durante su propia investigación.
  - Pipeline con GitHub OIDC sin claves permanentes, trazas con OpenTelemetry hacia X-Ray y CloudWatch, y alarma verificada rompiendo la dependencia a propósito. Cerrado con un procedimiento de limpieza escrito antes de ejecutarlo.
- **JavaScript / TypeScript, NestJS, Prisma, PostgreSQL, Redis, BullMQ, SGLang, Docker**
  PostulaDo - API de postulaciones con IA *2026*
  - API NestJS con análisis estructurado de ofertas mediante SGLang (XGrammar y RadixAttention), con salida validada contra esquema y scraping con Cheerio.
  - Prisma sobre PostgreSQL, colas con BullMQ y Redis, y suite e2e contra PostgreSQL y Redis reales en Docker Compose con GitHub Actions bloqueando el merge.
- **Node.js, Express, MongoDB, Docker, Terraform, Ansible, Nginx, GitHub Actions**
  Multi-Container App - API contenerizada *2026*
  - API REST containerizada con infraestructura como código, Nginx como reverse proxy y pipeline de CI/CD con runners self-hosted.

Formación Académica

- **Universidad del Gran Rosario**
  Licenciatura en Ciencia de Datos *2026 - Presente*
  Segundo cuatrimestre en curso. Aprendizaje Automático, Aprendizaje por Refuerzo y Metodología de la Investigación, con proyecto de scraping y análisis de datos.
- **Instituto Politécnico Formosa** **Formosa, Argentina**
  Técnico Superior en Desarrollo de Software Multiplataforma *2022 - 2025*
  Promedio general 8.60 y promedio 9.66 en Prácticas Profesionales. Egreso 05/12/2025; título emitido 20/03/2026. Formación complementaria en Minería de Datos, Modelos y Aplicaciones de la IA, React Native y Flexmind.

Idiomas

- Español (Argentina) - nativo. Inglés - nivel B2, profesional.

Certificaciones

- Learn to Cloud - capstone de Cloud Engineering (Journal API), completado 10/2026: AWS, Kubernetes, Terraform, CI/CD seguro y observabilidad.
- AWS Certified Developer - Associate - en curso.

Referencias

- Disponibles a pedido.