

Eliott Velarde

(+51) 944 181 210 | eliottvelardeverde5@gmail.com | Lima, Peru

[linkedin.com/in/eliott-velarde](https://www.linkedin.com/in/eliott-velarde) | github.com/EliottV17

Educación y Certificaciones

- Computación e Informática, Cibertec Estado - Graduado
- Masterclass Diseño de sistemas, Nodejs & SQL (Academia X)

Historial Laboral y Proyectos

Pasante de Ingeniería de Software, SimplyTech, Remoto

Julio 2025 - Diciembre 2025

- Participé en la construcción de una plataforma web bajo metodologías Scrum, integrando servicios backend y consumiendo APIs.
- Consumo de APIs REST e integraciones se implementaron para conectar las interfaces con los microservicios del servidor, asegurando un flujo de datos rápido y sin interrupciones para el usuario final.
- Trabajo colaborativo y gestión de repositorios (Git/GitHub) garantizaron el cumplimiento de los *sprints*, aplicando flujos de integración y revisión constante de código para el despliegue exitoso de nuevas funcionalidades.
- Resolución de problemas y capacidad analítica se aplicaron diariamente para estructurar la lógica cliente-servidor, resultando en un sistema altamente mantenible.

Desarrollador Backend, Meevent, Remoto

Abril 2026 – Agosto 2026

- Colaboré en la construcción y estabilización de la lógica del servidor aplicando Java, Spring Boot y Arquitectura de Microservicios.
- Desarrollo y refactorización orientada a objetos se ejecutó para resolver tareas críticas del *backlog*, asegurando la correcta integración de las REST APIs con los sistemas cliente.
- Pruebas unitarias y funcionales (JUnit/Mockito) se implementaron exhaustivamente para validar nuevas *features* antes de producción, aplicando el patrón AAA para garantizar la calidad y resiliencia del código.
- Principios de Clean Code, SOLID y diseño de bases de datos (SQL) se mantuvieron como estándar en cada *Pull Request*, superando rigurosas revisiones en un flujo de control de versiones basado en ramas.

- Construí el núcleo de una plataforma integrando TypeScript y NestJS bajo principios de Arquitectura Hexagonal, preparando el sistema para un entorno de microservicios.
- Desarrollo de REST APIs y enfoque API First se aplicaron utilizando NestJS y PostGIS, optimizando el consumo de tablas espaciales y asegurando respuestas ligeras para la interfaz.
- Buenas prácticas de desarrollo (SOLID, Clean Code) y seguridad se implementaron mediante JWT y OAuth 2.0, protegiendo el acceso a las rutas y garantizando la integridad de las transacciones.
- Herramientas de desarrollo asistido (OpenCode, Copilot) y control de versiones (Git/GitHub) se utilizaron para acelerar la escritura de código, generar pruebas y mantener un repositorio colaborativo impecable.