

Brian Benegas

Desarrollador Backend · Arquitectura de sistemas
Ituzaingó, Buenos Aires, Argentina · Disponible para remoto
contactobrianleft@gmail.com · +54 11 3155 6315
github.com/Brianleft28 · linkedin.com/in/brian-benegas-44770729b · brianleft.com

PERFIL

Desarrollador backend enfocado en sistemas de sector público que funcionan todo el día, desatendidos, y no se pueden caer. Trabajo de punta a punta: backend y frontend, y también el Linux donde corren, el CI/CD que los despliega, el nginx que los publica y el certificado que los asegura. La mayor parte de mi trabajo consiste en blindar sistemas de tiempo real bajo carga: escalado horizontal, comportamiento de memoria en clientes de larga duración y acceso a base de datos en hora pico. Actualmente lidero el equipo de innovación del Municipio de Ituzaingó.

EXPERIENCIA

Municipio de Ituzaingó — Secretaría de Innovación y Tecnología

Buenos Aires, Argentina · Septiembre 2023 – Presente
Líder del Equipo de Innovación (Octubre 2024 – Presente) · **Desarrollador Backend** (Septiembre 2023 – Octubre 2024)

Desarrollador principal de VUS (Ventanilla Única Simplificada), un sistema llamador de turnos en tiempo real que funciona a diario en oficinas municipales: tótems de autogestión, impresión térmica de boletos, pantallas de sala de espera, puestos de atención, dashboard en vivo y analítica histórica, desplegado sobre Kubernetes.

- Resolví la pérdida intermitente de eventos en tiempo real al escalar a múltiples réplicas en Kubernetes, incorporando Redis como adapter pub/sub de Socket.IO y sacando de la memoria de cada proceso el estado de las conexiones, que en silencio impedía que los pods fueran intercambiables.
- Eliminé los timeouts de la API en el pico diario reescribiendo una consulta no-SARGable (`DATE(campo) = CURDATE()`) como un rango puro, devolviéndole el índice al motor y reduciendo un Full Table Scan a milisegundos.
- Dejé plana la huella de memoria de las pantallas que nunca se apagan, reemplazando un Set sin límite por un Map tipo LRU que expira a los 60 segundos, y saqué los watchdogs artesanales de reconexión para delegar en el exponencial backoff nativo de Socket.IO.
- Eliminé los llamados duplicados por audio con un debounce por hash de ID y estado, con ventana anti-repetición de 15 segundos.
- Desacoplé el navegador de los puertos del sistema operativo construyendo un daemon local en .NET que traduce JSON sobre HTTP a comandos ESC/POS nativos, aislando el driver de las impresoras térmicas en un único componente reemplazable.
- Desarmé un monorepo falso —cuatro microservicios compartiendo un solo .git— separándolo en repositorios independientes con su propio pipeline de CI/CD en GitLab, terminando con las colisiones de dependencias y los builds cruzados.
- Separé el dashboard de operación en vivo, alimentado por sockets, de la analítica histórica, para que las consultas agregadas pesadas nunca corran sobre el camino de tiempo real.
- Llevé oficinas, servicios y comportamiento del llamador a parametría configurable en runtime: lo que cambia según la institución es dato y no un build nuevo.
- Construí un ETL recurrente en Node.js que migra el padrón mensual de un programa social desde planillas Excel a MySQL, dejando los archivos con datos personales fuera del repositorio por diseño y exigiendo selección explícita del archivo mediante una CLI interactiva, para evitar reprocesos silenciosos.
- Lidero el equipo de innovación, coordino la entrega entre áreas técnicas y administrativas, y capacito a personal municipal en los sistemas implementados.

Pidgeon Solutions — Cofundador

Buenos Aires, Argentina

- Cofundé un estudio de servicios de software, junto a otro desarrollador y un diseñador gráfico.

PROYECTOS DESTACADOS

VUS — Sistema llamador de turnos municipal

NestJS · SvelteKit · TypeScript · MySQL · TypeORM · Socket.IO · Redis · Kubernetes · .NET · GitLab CI/CD

Ecosistema completo de atención al público: tótem táctil, API orquestadora, pantallas de sala de espera, panel del operador, daemon de impresión ESC/POS, dashboard en vivo y analítica. Pensado para hardware heterogéneo y operación desatendida, donde cada componente falla distinto y todos tienen que degradar sin tirar el resto.

brianleft.com — Portfolio interactivo con RAG escrito a mano

SvelteKit · NestJS · TypeScript · MySQL · Redis · Docker · nginx

Portfolio que se navega como una consola de Linux y responde preguntas sobre mi trabajo mediante un pipeline de recuperación escrito paso a paso en TypeScript —ingesta de documentos, extracción de keywords con LLM, recuperación por relevancia, armado de contexto y streaming— sin motor de workflows en el medio.

Autogestionado de punta a punta en mi propio VPS, incluyendo nginx, TLS y despliegue.

SØREN — Visión por computadora sobre un proceso desatendido

Python · OpenCV · Tesseract · Discord Webhooks

Lee el estado de un proceso externo por OCR y análisis de píxeles, sin tocar su memoria ni su tráfico de red, y actúa en consecuencia. Corre horas sin supervisión: lo interesante no es automatizar, sino que se recupere solo cuando el mundo cambia debajo.

TECNOLOGÍAS

Lenguajes · TypeScript, JavaScript, C#, Python, SQL

Backend · Node.js, NestJS, .NET, TypeORM, Socket.IO, APIs REST

Frontend · SvelteKit, Svelte

Datos · MySQL, MongoDB, Redis

Infraestructura · Docker, Kubernetes, nginx, Linux (Debian, Ubuntu, Arch), GitLab CI/CD, PM2, administración de VPS

Prácticas · SOLID, clean code, programación defensiva para sistemas 24/7, Git, Jira, Postman

FORMACIÓN

Tecnicatura en Desarrollo de Software — Centro Universitario Regional Ituzaingó · 2023 – Presente

IDIOMAS

Español (nativo) · Inglés (avanzado)

CURSOS

Platzi, 2024 — Programación con C#, fundamentos de .NET, manejo de datos con LINQ, principios SOLID, programación orientada a objetos, buenas prácticas y clean code, tipos y funciones avanzadas de TypeScript.