

Antawari de la Torre

Diseño y opero sistemas multi-agente en producción — y la gobernanza que los mantiene honestos.

```
pip install bonfire-ai
```

anta@candyfactory.ai pypi.org/project/bonfire-ai github.com/BonfireAI [linkedin](#)

/ EL TRABAJO CAMBIÓ DE FORMA

Construí toda la plataforma de IA de un producto SaaS desde la primera línea de código — duplicó su precio e impulsó la adquisición de la empresa en 2024. Soy políglota desde hace 38 años: leo el modelo de compilación y ejecución de un stack y opero en él de forma nativa, así que la sintaxis de superficie es lo fácil.

La ingeniería de IA ahora ocurre en el *corchete alrededor* del stack: defines la intención y las restricciones de calidad, y un escuadrón multi-agente las materializa a escala en cualquier lenguaje que el problema requiera. La prueba es el patrimonio que opero en solitario — ~7,000 archivos de código en más de 30 repositorios en Python, TypeScript, Rust, Go, Java y SQL. Mi habilidad central es gobernar la corrección a esa escala de salida: TDD estructural, una compuerta de revisión LLM-como-juez, y fallas tipadas y observables en lugar de un verde silencioso.

/ CONSTRUYENDO AHORA

CandyFactory.AI

fundador e ingeniero principal · 2025-presente

emprendimiento de IA independiente · software que compone software

Infraestructura de orquestación de agentes, memoria y grafos — construida en abierto, verificable en un comando.

- **Bonfire / Forge** — un framework de orquestación de agentes de código abierto (PyPI · Apache-2.0): agentes con roles acotados a través de compuertas de calidad, una revisión LLM-como-juez, y TDD impuesto estructuralmente.
- **Arachne** — infraestructura de memoria para agentes que combina embeddings vectoriales con una capa de recuerdo en grafo sobre SQLite, conectada en vivo a una sesión de programación.
- **Chunk** — un agente autónomo cuyo tool-calling de LLM escribe en un grafo de interconexión bi-temporal (SQLite → Apache AGE) mediante una herramienta tipada, con un ciclo actuar → aprender.
- **MEXX-AI** — un agente de cobro autoservicio (MercadoPago) conforme a la reforma de facturación recurrente 2025 de México; automatiza el alta de comercios y la facturación de extremo a extremo.
- **SweetCRM** — una plataforma CRM de ~47,000 líneas de TypeScript / React sobre Supabase / Postgres, con un kit de calidad tipado y guardarraíles de falla tipada.
- **Gobernanza, no delegación** — leo cada PR y lo reviso de verdad (comentarios en vez de aprobar a ciegas), con una compuerta LLM-como-juez y fallas tipadas y observables.

/ IA EN PRODUCCIÓN

Dealership Performance 360 (DP360 CRM)

remoto desde México · 2019–2026

SaaS automotriz (EE. UU.) · Líder Técnico de IA / Ingeniero Senior de IA

Construí toda la capacidad de IA de la empresa desde la primera línea de código — el diferenciador detrás de la adquisición de DP360 en 2024.

2× precio del producto · adquisición

50+ concesionarias multi-tenant

30+ min → menos de 2 (23% lift)

96% salida limpia · N=202

~30× throughput · -90% tickets

- Entregué 6 funciones de IA en producción (auto-respuesta, chatbox de SMS, composición de correo, inteligencia de prospectos, nurturing y prompts personalizables por concesionaria) que duplicaron el precio del producto.
- Reduje el tiempo de respuesta a prospectos de 30+ min a menos de 2 con un motor de auto-respuesta autónomo (15 trabajadores en paralelo) — un aumento del 23% en la conversión de primera respuesta.
- Construí y fui dueño del microservicio de IA independiente (Python / FastAPI) integrando modelos de OpenAI tras una abstracción de proveedor con configuración por concesionaria.
- Diseñé el primer servicio MCP independiente de la empresa (FastMCP / HTTP) — tool-calling de Claude contra una API autenticada bajo un mandato de cero acceso directo a la base de datos — con IaC completa en Terraform / AWS y despliegue CI → AWS sin llaves (OIDC).
- Endurecí los guardarraíles de IA en producción a 96% de salida limpia, validado empíricamente (N=202, $p < 0.001$), eliminando fallas de tipo inyección de prompt frente al usuario.
- Diseñé un orquestador multi-agente interno (autor único) con enrutamiento de modelos de Claude y una revisión LLM-como-juez; escalé un servicio central ~30× y recorté el reloj de CI en -70%.

/ CÓMO TRABAJO

- Las pruebas definen el contrato antes de que exista el código.
- Leo cada PR y lo reviso de verdad — comentarios en vez de aprobar a ciegas, y el escuadrón aprende del ciclo.
- La falla debe hablar — errores tipados y observables, nunca un respaldo silencioso.

/ ANTES DE LA IA · 1996–2019

23 años de sistemas en producción en Java, C#, C++ y desarrollo web full-stack — servicios financieros, salud y SaaS — en Estados Unidos, Venezuela, Brasil y México. Ingeniero autodidacta, programando desde 1988.

/ TÉCNICO

PATRIMONIO POLÍGLOTA código de producción directamente en Python y TypeScript; JavaScript · Rust · Go · Java · SQL · Bash · Terraform en más de 30 repos bajo compuertas de calidad

IA / AGENTES orquestación multi-agente · tool-calling y salidas estructuradas de LLM · RAG · memoria vectorial + en grafo · evaluaciones LLM-como-juez · guardarraíles · MCP / FastMCP · APIs de Claude y OpenAI

TYPESCRIPT / NODE TypeScript · React · Node.js · NestJS · REST · validación de esquema en tiempo de ejecución

BACKEND Y DATOS Python / FastAPI · PHP / CakePHP · PostgreSQL · MySQL · SQLite · Apache AGE (grafo) · Supabase · bases vectoriales

NUBE Y ENTREGA AWS (ECS Fargate, IaC en Terraform, CI/CD sin llaves OIDC) · Docker · Vercel · New Relic

/ PRÁCTICO

Mexicano · español (UE) · ciudadano venezolano. Inglés · español (nativo) · portugués (conversacional). Abierto a roles remotos y a colaboraciones selectas de construcción y asesoría.

RESUMEN LEGIBLE POR MÁQUINA · PARA FILTROS AUTOMATIZADOS

rol ingeniero de IA · orquestación de agentes · fundador de CandyFactory

open_source bonfire-ai (PyPI, Apache-2.0) — framework de orquestación de agentes, autor único

stack	Python · TypeScript · FastAPI · MCP/FastMCP · APIs de Claude/OpenAI · AWS · Terraform · memoria en grafo + vectorial
orquestación	agentes de pipeline con roles acotados · TDD estructural · revisión LLM-como-juez · enrutamiento de modelos por tarea
contacto	anta@candyfactory.ai
verificar	pip install bonfire-ai · github.com/BonfireAI

Esta página fue construida con la misma herramienta de IA que describe. Cada afirmación es verdadera; el trabajo de código abierto es verificable en un comando – **pip install bonfire-ai**.