PORTAFOLIO DE PROYECTOS

Cristian Pinilla

AI Automation Architect / Administrador Infraestructura Cloud

13
PROYECTOS

5
RAG EN PRODUCCIÓN

Administrador de Infraestructura Cloud y AI Automation Architect, con más de 20 años en IT y 5+ especializados en Cloud/SysOps. En el último año llevé cinco sistemas RAG a producción sobre AWS, PostgreSQL/pgvector y n8n, construí consolas propias de gestión cloud con asistentes de IA (niveles Observer/Advisor/Executor) para AWS, Azure y GCP, y un stack de 8 agentes autónomos que monitorean, diagnostican y remedian infraestructura 24/7 con aprobación humana explícita. No son demos: son sistemas con usuarios y datos reales, medidos con métricas de retrieval verificables (golden sets, recall@k) en vez de afirmaciones sueltas.

Email: cpinilla74@gmail.com **LinkedIn:** linkedin.com/in/cristian-pinilla-lira-24125358 **Portafolio:** cpinilla74.com
Generado: 03-08-2026

DESTACADO · EN PRODUCCIÓN

Consulta Normativa CMF

Asistente RAG especializado en la normativa de la Comisión para el Mercado Financiero (CMF) de Chile. Indexa 850+ documentos oficiales incluyendo la Recopilación Actualizada de Normas (RAN), Compendio de Normas Contables para Bancos, Manual del Sistema de Información, Circulares de Basilea III, normativa de Tarjetas y Operadoras de Pago, y regulación de Cooperativas y SGR.

El chatbot responde exclusivamente con información verificada de los documentos, citando fuentes con enlaces directos a los archivos originales en Google Drive. Incluye rate limiting, validación de seguridad y respuestas en Markdown estructurado.

DOCUMENTOS
841 con contenido real (851 procesados)

CHUNKS
20.620 vectores

MODELO
GPT-4.1-mini

EMBEDDING
text-embedding-3-small

BASE DE DATOS
PostgreSQL + pgvector (Neon)

ORQUESTACIÓN
n8n self-hosted

n8n · PostgreSQL/pgvector · OpenAI · AWS · CloudFront · S3 · Neon Tech · RAG

VALIDACIÓN MEDIDA

0.867	0.667	Cosine
RECALL@15	BASELINE ANTERIOR	SIMILARITY

Pipeline de ingesta v6 (chunking corregido + filtro de período por regex): golden set fijo de 30 preguntas generadas por LLM a partir de chunks reales (ground truth), medido sobre el corpus completo de producción (no una muestra). Recall subió de 0.667 a 0.867 (+30% relativo) combinando top_k=15 con un filtro de período (mes/año) extraído del nombre de archivo — sin costo extra de LLM. El patrón de falla remanente sigue siendo preguntas por cifras/fechas exactas en informes periódicos casi idénticos entre sí, limitación estructural de embeddings densos más que del chunking.

DESTACADO · EN PRODUCCIÓN

Portal PJUD

Asistente RAG del Poder Judicial de Chile, especializado en documentos institucionales del sistema judicial. Cubre manuales de la Oficina Judicial Virtual (OJV), actas de la Corte Suprema, leyes de tramitación electrónica, instructivos de transparencia judicial y protocolos de atención al público.

Permite consultar procedimientos como agendamiento de audiencias, tramitación digital, y protocolos de la Mesa de Ayuda OJV. Responde con citas textuales de los documentos fuente.

DOCUMENTOS **33 manuales/actas/leyes OJV**	CHUNKS **604 vectores**
MODELO **GPT-4.1-mini**	EMBEDDING **text-embedding-3-small**
BASE DE DATOS **PostgreSQL + pgvector (Neon)**	ORQUESTACIÓN **n8n self-hosted**

n8n · PostgreSQL/pgvector · OpenAI · AWS · Node.js · Neon Tech · RAG

VALIDACIÓN MEDIDA

1.0	0.833	Cosine
RECALL@15	BASELINE ANTERIOR	SIMILARITY

Pipeline de ingesta v6, mismo protocolo que CMF: golden set fijo de 30 preguntas, recall medido contra la query real de producción (top_k=15). Recall perfecto (30/30) — mejor resultado medido de todo el portafolio. Corpus más acotado (33 documentos, 604 chunks) que CMF, con menos competencia por posiciones del top-k entre documentos casi idénticos.

DESTACADO · 3 COMUNAS ACTIVAS

Asistente Municipal

Sistema de consulta normativa municipal con inteligencia artificial. Tres comunas de la Región Metropolitana con sus ordenanzas indexadas y disponibles para consulta pública:

• Puente Alto — Ordenanzas sobre ferias libres, tenencia de mascotas, ruidos molestos, aseo y ornato, tránsito, subvenciones municipales y microempresas familiares.

• La Florida — Ordenanzas de participación ciudadana, regulación de comercio en vía pública, patentes comerciales, tenencia de animales y uso de espacios públicos.

• Til-Til — Ordenanzas de medio ambiente, protección de recursos naturales, actividades agrícolas y rurales, agua potable y servicios básicos.

Cada chatbot responde citando la ordenanza específica (nombre y número), con montos de multas textuales sin redondear.

COMUNAS
Puente Alto, La Florida, Til-Til

MODELO
GPT-4.1-mini

EMBEDDING
text-embedding-3-small

BASE DE DATOS
3 DB pgvector independientes (Neon)

ORQUESTACIÓN
n8n self-hosted

DATOS
Ordenanzas municipales públicas

n8n · PostgreSQL/pgvector · OpenAI · AWS · Neon Tech · RAG · Gobierno Local

VALIDACIÓN MEDIDA

0.933	0.8	0.867	Cosine
TIL-TIL RECALL@6	LA FLORIDA RECALL@15	PUENTE ALTO RECALL@6	SIMILARITY

Las 3 comunas corren sobre el pipeline de ingesta v6, evaluadas con golden set fijo de 30 preguntas por comuna (antes N=8, muestra poco confiable). Til-Til subió de 0.875 a 0.933; La Florida, el salto más grande, de 0.375 a 0.8 (+113% relativo) combinando el pipeline corregido con top_k=15. Puente Alto subió de 0.5 a 0.867 (+73% relativo) tras recuperar las 37 ordenanzas desde el portal de transparencia municipal y reprocesarlas con el pipeline v6. El patrón de falla remanente en las 3 comunas es el mismo: preguntas por datos exactos (número/fecha de ordenanza, monto de multa) que la similitud semántica no siempre ubica en el top-k.

DESTACADO · EN PRODUCCIÓN

Asistente Médico

Portal completo Médico-Paciente: canal de comunicación y seguimiento que continúa el contacto más allá de la consulta física. Registro/login con SSO de Google + reCAPTCHA, perfil y ficha clínica con historial cronológico, directorio de médicos, mensajería con notificación por correo, videollamadas WebRTC P2P (con servidor TURN propio), agenda de citas con disponibilidad semanal y anti-doble-booking, y un chatbot de intake clínico (function-calling) que entrevista al paciente y arma el caso para el médico sin dar diagnóstico.

Backend propio en Python/Flask (migrado desde n8n) con PostgreSQL en Neon, desplegado en EC2 con gunicorn + nginx.

FUNCIONALIDADES
Registro, Login SSO, Ficha, Mensajería, Video, Agenda, Chat IA

BACKEND
Python/Flask (gunicorn + nginx)

BASE DE DATOS
PostgreSQL (Neon)

VIDEO
WebRTC P2P + TURN (coturn) propio

AUTH
SSO Google + reCAPTCHA v2

ESTADO
Operativo — en producción

Python/Flask · PostgreSQL · WebRTC · Google OAuth · AWS · OpenAI

DESTACADO · PRODUCTO PROPIO

CloudConsole — "Jarvis para AWS"

Plataforma propia de gestión cloud construida desde cero: 14 módulos cubriendo Inventory, Monitoring, Costos/FinOps, Seguridad (con AI Score en 2 capas), Backup & DR, SLA & Uptime, Architecture, IaC Export (CloudFormation/Terraform) y Reportes ejecutivos multi-módulo.

Incluye un asistente de voz (Whisper STT + OpenAI TTS) multi-LLM (OpenAI/Anthropic/Google) con 3 niveles de autonomía: Observer (solo lectura), Advisor (propone comandos) y Executor (ejecuta acciones con aprobación humana explícita). Monitor de alarmas 24x7 con diagnóstico AI y notificación por email.

Diseñada con potencial comercial como producto independiente para gestión y automatización cloud.

MÓDULOS
14 + asistente de voz

STACK
Python/Flask + boto3

LLM
OpenAI, Anthropic, Google

VOZ
Whisper STT + OpenAI TTS

AUTONOMÍA
Observer / Advisor / Executor

ESTADO
Operativo — Demo bajo solicitud

Python/Flask · boto3 · Multi-LLM · Voice AI · AWS · IaC Export

Proyecto confidencial / sin demo pública

DESTACADO · PRODUCTO PROPIO

Aether — Orquestador Multi-Cloud

De la evolución de CloudConsole nacieron tres agentes bot independientes de administración cloud: Ember (AWS), Céfiro (Azure) e Iris (GCP). Cada uno opera y asiste en su propia nube — pero administrar 3 agentes por separado en un entorno multi-cloud no escala.

Por eso nació Aether: el orquestador central que interactúa directamente con Ember, Céfiro e Iris bajo un mismo modelo Dispatcher — propone la acción, espera aprobación humana explícita y ejecuta — permitiendo administración multi-cuenta y multi-nube desde un solo punto de control.

Mismo modelo de autonomía en capas (Observer/Advisor/Executor) que CloudConsole, extendido ahora a 3 proveedores cloud coordinados por un solo cerebro central.

AGENTES ORQUESTADOS
Ember (AWS), Céfiro (Azure), Iris (GCP)

MODELO
Dispatcher: propone → aprueba humano → ejecuta

ALCANCE
Multi-cuenta / Multi-nube

AUTONOMÍA
Observer / Advisor / Executor

ORIGEN
Evolución de CloudConsole

ESTADO
Operativo — validado con acciones reales

Multi-Cloud · AI Agents · AWS · Azure · GCP · Orquestación

DESTACADO · EN PRODUCCIÓN

Denuncia Vecinal

Portal donde un vecino reporta un problema urbano (bache, basural, semáforo caído, accidente) subiendo texto libre + fotos + geolocalización. Antes de construirlo, evalué las 3 APIs de Vision clásicas (Azure AI Vision, AWS Rekognition, GCP Vision) contra una foto real de microbasural — las 3 fallaron: son clasificadores cerrados que solo devuelven tags fijos (truck, car, tree) y no aceptan texto guía.

El motor real es Gemini 2.5 Flash multimodal: recibe foto + texto en el mismo prompt, usa el texto del vecino como contexto para orientar el análisis de la imagen, y es resistente a falsos reportes — probado con texto exagerado + foto inocua (lo refutó correctamente) y con foto de choque real + texto no relacionado (detectó el incidente igual).

El resultado estructurado (categoría, urgencia, confianza, alerta de falso reporte) se guarda en Postgres y dispara un correo automático a la municipalidad.

MOTOR IA
Gemini 2.5 Flash multimodal (foto + texto)

POR QUÉ NO VISION APIS
Azure/AWS/GCP Vision son clasificadores cerrados sin texto guía — fallaron en la prueba real

BACKEND
n8n (webhook → Gemini → Postgres → email)

DB
Neon Postgres (denuncia_vecinal)

ANTI FALSO-REPORTE
Validado con 2 casos cruzados (texto exagerado/foto inocua y viceversa)

ESTADO
Operativo — validado end-to-end en producción

Gemini Multimodal · n8n · PostgreSQL/Neon · AWS · SES

ARQUITECTURA

MCP Architecture

Arquitectura de orquestación de múltiples knowledge bases vía webhooks. Diseñada para exponer capacidades RAG como servicio, con API propietaria de síntesis que protege la propiedad intelectual de las bases de conocimiento subyacentes.

Soporta tres modelos de despliegue: SaaS (multi-tenant), híbrido (KB del cliente + orquestación centralizada) y on-premise completo. Cada knowledge base se conecta como un microservicio independiente con su propio embedding y vector store.

PATRÓN
Orquestación multi-KB

API
Webhooks REST

MODELOS
SaaS, Híbrido, On-premise

ORQUESTACIÓN
n8n

VECTOR STORE
pgvector por KB

ESTADO
Diseño completado

n8n · Webhooks · RAG · API Design · pgvector · Microservicios

Proyecto confidencial / sin demo pública

EMPRESA PERSONAL

ConsultaIA

Marca y plataforma personal para ofrecer servicios de consultoría TI especializados en inteligencia artificial, automatización de procesos y optimización cloud. Origen del producto Consulta-CMF.

Foco en soluciones RAG a medida para empresas, integraciones con n8n, y arquitecturas serverless en AWS. Incluye portal web con presentación de servicios y canal de contacto.

TIPO
Empresa personal / Marca

SERVICIOS
Consultoría IA, RAG, Automatización

INFRAESTRUCTURA
AWS (cuenta propia)

URL
consultaia.cpinilla74.com

CI/CD
S3 + CloudFront

ESTADO
Activo

n8n · PostgreSQL/pgvector · AWS · CloudFront · OpenAI · Consultoría

Proyecto confidencial / sin demo pública

INFRAESTRUCTURA

AWS Multi-Region Infrastructure

Infraestructura cloud completa desplegada en 3 regiones AWS (us-east-1, us-east-2, us-west-1) con 8+ instancias EC2. Incluye:

- WAF Free activado en distribuciones CloudFront para protección web
- Savings Plan de $0.30/hr (1 año) con ~85% de utilización
- AWS Backup automatizado con vault semanal
- DLM (Data Lifecycle Manager) para snapshots EBS diarios
- Dumps SQL nocturnos para bases de datos críticas
- Route 53 con múltiples dominios y certificados ACM
- SES para envío de emails transaccionales
- Tags de costos (Project/Env/Owner) en ~50 recursos con Cost Category activa

REGIONES
us-east-1, us-east-2, us-west-1

INSTANCIAS
8+ EC2 activas

CDN
CloudFront + WAF

BACKUP
DLM diario + dumps SQL

DNS
Route 53 multi-dominio

AHORRO
Savings Plan ~85% utilización

AWS · EC2 · WAF · CloudFront · Route 53 · S3 · SES · DLM · FinOps

Proyecto confidencial / sin demo pública

MICROSERVICIO

Sistema Extracción PDF V6

Microservicio FastAPI que forma parte del Pipeline RAG de Ingesta v6. Extrae texto de PDFs complejos usando una cascada de tres motores:

1. pymupdf4llm — Primera opción para PDFs con texto embebido. Rápido y preciso.
2. pdfplumber — Fallback para PDFs con tablas complejas o layouts mixtos.
3. Tesseract OCR — Último recurso para PDFs escaneados o con imágenes de texto (procesa 45 páginas escaneadas en ~5:30 min).

El sistema selecciona automáticamente el mejor motor según la calidad del texto extraído. También soporta extracción de PPTX (python-pptx). Desplegado en EC2 Spot bajo demanda vía PM2/systemd, integrado con el pipeline de ingesta RAG v6 que orquesta: Google Drive → Extracción → Chunking + LLM → Embedding → pgvector.

La v6 corrige dos bugs reales del pipeline (tamaño de chunk mal calculado y arquitectura de loop) y mejora la calidad de retrieval de forma medible en las 4 bases regeneradas. Validado con 4 KBs en producción: CMF (841 docs, 20.620 chunks), PJUD (33 docs, 604 chunks), Til-Til (19 docs, 616 chunks) y La Florida (43 docs, 610 chunks).

Ejemplo medido (golden set fijo de 30 preguntas, mismo top_k en ambas versiones): PJUD pasó de recall@15 = 0.833 (25/30, pipeline v5) a recall@15 = 1.0 (30/30, pipeline v6) — recall perfecto tras la corrección de chunking.

FRAMEWORK
FastAPI (Python)

OCR ENGINES
pymupdf4llm, pdfplumber, Tesseract

DEPLOY
EC2 Spot bajo demanda + PM2/systemd

PIPELINE
Ingesta RAG v6 (n8n)

VALIDACIÓN
4 KBs en producción: CMF, PJUD, Til-Til, La Florida

EJEMPLO REAL
PJUD: recall@15 0.833 → 1.0 (N=30)

MEJORA RECALL@K
+3.7% a +60% vs pipeline v5 (golden sets N=30)

ESTADO
Producción

Python · FastAPI · OCR · PM2 · pymupdf4llm · pdfplumber · Tesseract · n8n

Proyecto confidencial / sin demo pública

EN PROCESO · COMPUTER VISION (NO GENAI)

BioGate — Verificación Biométrica de Identidad

Portal de validación de identidad con 4 capas de defensa en profundidad: SSO con Google (identidad declarada), reCAPTCHA v3 invisible (anti-bot), liveness básico vía cámara web y verificación biométrica facial con AWS Rekognition (CompareFaces) contra un set de fotos de referencia.

Evalué las tres nubes antes de elegir el motor: Azure Face API está en Limited Access desde 2023 (requiere aplicar y justificar el caso de uso ante Microsoft), y GCP Vision API solo detecta rostros pero no tiene un endpoint de matching 1:1/1:N. AWS Rekognition fue la única opción que resolvió el caso de uso directamente, dentro del free tier.

Importante: este es un proyecto de Computer Vision / ML clásico (embeddings + similarity score), no un proyecto GenAI — no hay ningún LLM en el flujo de verificación. Lo mantengo separado de mis proyectos de RAG/agentes para no mezclar categorías.

Desplegado 100% serverless: Lambda + API Gateway con dominio y certificado SSL propios. Probado end-to-end con resultado real de 10/10 matches, similarity ~99.9%.

MOTOR BIOMÉTRICO
AWS Rekognition CompareFaces

HOSTING
Lambda + API Gateway (serverless)

AUTH
Google OAuth (SSO) + JWT de sesión corta

ANTI-BOT
reCAPTCHA v3 invisible

CATEGORÍA
Computer Vision / Seguridad — NO GenAI

ESTADO
En proceso — prueba de concepto validada

AWS Rekognition · Lambda · API Gateway · S3 · Google OAuth · reCAPTCHA v3 · Computer Vision

VALIDACIÓN MEDIDA

10 / 10	~99.9%	4
MATCHES	SIMILARITY PROMEDIO	CAPAS DE VALIDACIÓN

Prueba real end-to-end (2026-07-03): login SSO → captura de cámara → reCAPTCHA v3 → comparación contra las 10 fotos de referencia en S3. Resultado: aprobado, coincidencia en el 100% de las fotos de referencia.

DESTACADO · EN PRODUCCIÓN

Bitácora — Base de Conocimiento que se Documenta a Sí Misma

Durante 10 meses documenté cada decisión de arquitectura al construir una plataforma RAG real (CMF): por qué pgvector y no Bedrock, qué funcionó y qué no (búsqueda híbrida, reranking), qué medí antes de confiar en cada mejora — con la misma disciplina de un ADR de ingeniería, no notas sueltas.

Reuní toda esa documentación y construí Bitácora: un asistente RAG que la ingiere completa y responde preguntas de cómo/cuándo/por qué con citas verificables a los documentos reales, sin inventar — y cuando el contexto no alcanza, lo dice explícitamente en vez de adivinar.

También genera sus propios documentos y diagramas bajo demanda: le pides un resumen para cierta audiencia o un diagrama de arquitectura, y lo sintetiza sola, citando las fuentes que usó. Los dos ejemplos enlazados abajo — la presentación en primera persona y el diagrama de arquitectura — fueron generados así, sin editar el contenido.

CORPUS
63 documentos, 688 chunks, metadata enriquecida

RETRIEVAL
Cosine similarity, top_k=15, recall@15 medido 0.967

GENERACIÓN
gpt-4.1-mini, temperature=0, citas de identificador inline

CAPACIDADES
Q&A conversacional + generación de documentos/diagramas on-demand

RUNTIME
n8n self-hosted, Postgres/pgvector en Neon

ESTADO
Operativo — webhooks con autenticación por API key

RAG · n8n · pgvector · OpenAI · Documentación técnica · Self-hosted

VALIDACIÓN MEDIDA

0.967 RECALL@15 (GOLDEN SET, 30 PREGUNTAS)

0.900 RECALL@8

63 (688 chunks) DOCUMENTOS INGERIDOS

Recall medido con golden set generado por LLM sobre chunks reales (mismo método que el resto del portafolio): se samplean chunks al azar, se genera una pregunta única por chunk, y se mide si la búsqueda real de producción devuelve ese chunk dentro del top-k.