

Mi lab casero para Dev IA: IA soberana con hardware reciclado

¿Será posible... o será utopía?

● **Prof. Ricardo Darío Matas** · SysAdmin y profesor de tecnología



¿Puede una PC de oficina correr un LLM?



¿Necesito una GPU de USD 2.000 para usar IA?



¿O alcanza con el fierro que su empresa dio de baja?



No lo vamos a discutir: lo vamos a medir en vivo.

Regla del taller: cero promesas en láminas, solo tokens en pantalla.



—●—●—● DESDE DÓNDE HABLO

Una mirada desde la operación, no desde la teoría



Territorio de investigación

Entrenar los modelos, derivar la teoría, empujar el estado del arte. Un trabajo enorme y necesario.



Territorio de operaciones

Hacer que las cosas funcionen con el presupuesto que hay: virtualización, redes, medir, poner en producción.

Unos entrenan los modelos; **otros los ponemos a trabajar. Las dos cosas hacen falta.**

Todo lo que voy a afirmar hoy se verifica con una división y un cronómetro.



El costo real de depender de APIs externas



Dólares por token

Cada consulta se paga en moneda dura. Con inflación local, el costo es impredecible.



Tus datos viajan

Historias clínicas, contratos, expedientes: cada prompt sale a un servidor extranjero.



Permiso revocable

Cambios de términos, geobloqueos, sanciones. El servicio puede desaparecer mañana.

Soberanía = el modelo corre en TU fierro, con TUS datos, bajo TUS reglas.



Tres fierros, tres roles, cero dólares nuevos



Ryzen 5 3400G · 16 GB dual channel · Proxmox 8.4

Llama 3.1 8B · Qdrant · Open WebUI

Servidor de inferencia + RAG



APU A4 · 8 GB · año 2013

Llama 3.2 1B · Whisper base

Nodo edge



Core Duo · 4 GB · sin AVX

Grafana: tok/s, RAM, temperatura

Monitoreo en vivo

Cada fierro viejo tiene un rol. Eso es economía circular aplicada.



Así se ve la soberanía: fierro reciclado, real



Gabinete abierto, sensor a 37°, terminal viva y un ping a 8.8.8.8. Sin stock, sin photoshop: soberanía de verdad.

La inferencia es mover bytes, no calcular

Para generar CADA token, el CPU debe leer el modelo COMPLETO desde la RAM:

Token 1 → leer 4,9 GB de RAM

Token 2 → leer 4,9 GB de RAM

Token 3 → leer 4,9 GB de RAM

...

La caché L3 (4 MB) no salva: el modelo es mil veces más grande.



El cuello de botella

No son los GHz del CPU. Es el ancho de banda de la memoria.

El CPU pasa la mayoría del tiempo esperando a la RAM.

Calculá tu techo de tokens con una división

$$\text{tok/s (techo)} = \text{ancho de banda de RAM} \div \text{tamaño del modelo}$$

Ryzen 3400G: $38,4 \text{ GB/s} \div 4,9 \text{ GB (Llama 8B Q4)} = 7,8 \text{ tok/s de máximo físico}$

Hardware	Ancho de banda	Techo con 8B Q4
DDR4-2400 dual (este lab)	38 GB/s	~8 tok/s
DDR5-6000 dual	96 GB/s	~19 tok/s
Tesla P40 (2016, usada)	346 GB/s	~70 tok/s
RTX 3090 (usada)	936 GB/s	~190 tok/s

Una GPU de 2016 le gana 9× a un CPU moderno — por su memoria, no por sus núcleos.



Cuantización: el JPEG de los modelos



FP16 original

16 GB

Calidad 100%. No entra en tu RAM.



Q8

8,5 GB

Como un JPEG calidad 95:
indistinguible.



Q4_K_M ★

4,9 GB

~97% de calidad, 1/3 del tamaño. El
sweet spot.

Doble beneficio: entra en tu RAM + menos bytes que leer = más tokens por segundo.



Stack profesional en hardware modesto

Open WebUI

Interfaz multiusuario · el celular de la audiencia se conecta acá

Ollama · llama.cpp

Motor de inferencia · sirve los modelos por API REST

Qdrant + embeddings

Base vectorial para RAG · los documentos nunca salen

Proxmox 8.4 · LXC

Virtualización: un contenedor por servicio, como un datacenter

Las mismas herramientas que en producción, a una fracción del costo.



El chatbot soberano, midiendo tokens

● EN VIVO

```
$ ollama run llama3.1:8b --verbose
```

```
"Explicá qué es la soberanía  
tecnológica en 3 oraciones"
```

```
prompt eval rate: 45 tok/s  
eval rate: 6 tok/s ←
```

El goteo de palabras que ven = generación memory-bound en acción.



Lo que están viendo

Un ChatGPT privado...

...corriendo entero en MI Ryzen, sin nube.



Antes de comprar, auditá lo que tenés



dmidecode -t memory

¿Dual channel? ¿Velocidad real?

2×8 GB ✓ • a 2400 de 2667
MT/s



XMP / DOCP en BIOS

Activar lo que ya pagaste

+10% de tokens, gratis



num_thread = 4

Físicos vs. lógicos en Zen+

hasta +20% según carga

Benchmark antes/después medido en ESTE hardware — no en el blog de alguien con una 4090.



RAG: tus documentos, sin re-entrenar nada

1

Ingesta

Documentos → chunks de 300-500 tokens → embeddings → base vectorial

2

Búsqueda

La pregunta se vuelve vector → top 3-5 fragmentos más cercanos por significado

3

Generación

Los fragmentos se inyectan en el contexto → el LLM responde citando TU material

La clave: **todo el pipeline corre local. Historias clínicas, expedientes, contratos: nada sale de tu red.**



Matar la alucinación con la ley argentina

● EN VIVO

SIN RAG

"¿Qué dice el artículo 9 de la ley 25.326?"

El modelo inventa un texto plausible, con número de artículo y todo. No sabe que no sabe.

CON RAG

Misma pregunta, ley indexada en Qdrant.

Responde citando el artículo textual correcto, con la fuente. El antes/después más convincente que existe.

Los números que cierran en LatAm

USD 0

●
en hardware nuevo

Todo reciclado: remates, bajas de oficina,
e-waste

~30 W

●
consumo en inferencia

Menos que una lámpara vieja. Medido,
no estimado.

0 datos

●
salen de tu red

Sin API keys, sin términos de servicio, sin
permiso de nadie

Próximo upgrade coherente: una Tesla P40 usada de 24 GB — ancho de banda de datacenter a precio de remate.



●—●—● EL DATO QUE IMPORTA

¿ERA UTOPIA?

6 tok/s

Sin la nube.
Sin dólares.
Sin permiso de nadie.

La IA soberana no es utopía: es una decisión de arquitectura.



Tu lab arranca este fin de semana



La fórmula: ancho de banda ÷ tamaño del modelo = tu techo de tok/s



Auditá tu fierro antes de comprar: dmidecode, dual channel, XMP



Stack mínimo: Proxmox (o Docker) + Ollama + Open WebUI en una tarde



Q4_K_M es tu amigo: 97% de calidad, un tercio del tamaño



RAG local para datos sensibles: la killer app de la soberanía





¡Gracias, LatAm!

Cada fierro viejo es un nodo de soberanía esperando su rol.

ricardodariomatas.ar

in/ricardo-dario-matas

