Reduciendo la dependencia de NVIDIA con Intel Arc

La segunda fuente en rendimiento por watt para el nivel asequible de IA — rendimiento medido, eficiencia energética y costo total de propiedad de Intel Arc Pro (Battlemage) para cargas de trabajo modernas de IA en la plataforma ColabHive

Ing. José Luis Minich
jose@aureus-finance.com

Ing. Maximiliano Lucis
maximiliano@aureus-finance.com

ColabHive Research — Buenos Aires, Argentina
hello@colabhive.com · colabhive.com/research

Junio de 2026

Borrador de trabajo (v2 — K=3 repeticiones en las celdas principales de serving de LLM, SDXL y LoRA/QLoRA; QLoRA en Battlemage ya funciona — ver §8)

Resumen

Evaluamos comparativamente, cara a cara, una **Intel Arc Pro B70 (32 GB, Battlemage)** frente a una **NVIDIA RTX 3090 (24 GB, Ampere)** en la plataforma distribuida de inferencia/entrenamiento ColabHive, a lo largo de seis cargas de trabajo reales de IA (serving de LLM, Stable Diffusion XL y fine-tuning LoRA/QLoRA/TabNet) con throughput *atribuido por dispositivo* e instrumentación de energía de extremo a extremo. En las cargas de trabajo limitadas por cómputo que dominan el gasto actual en IA, la B70 entrega **95–112% del throughput de la 3090 consumiendo 56–64% de su potencia**, es decir, **∼1.45–2.0× mejor rendimiento por watt**. La difusión es una victoria rotunda de Intel (∼12% más throughput con la mitad de la energía por imagen); QLoRA de 4 bits — inicialmente reportado por error como no funcional — de hecho *funciona* en Battlemage y es ∼7% más rápido que la 3090 una vez que se le proporciona el selector de dispositivo SYCL correcto, una corrección que llevamos a producción y reportamos upstream como contribución de este trabajo. Con AWQ de 4 bits — la precisión que los operadores realmente despliegan — la B70 *supera* a la 3090 por 19–26% en serving de LLM, convirtiendo la casi-paridad en bf16 en una victoria limpia. El punto débil, con honestidad, son las cargas de trabajo con operadores (kernels) pequeños (TabNet: la 3090 es ∼2.2× más rápida). En lo económico, Intel Arc Pro cuesta ∼0.4× el $/GB de VRAM de una 3090 *nueva* y ∼0.6–0.8× el de una *usada*, con una ventaja estructural de eficiencia por watt que el mercado de usados no puede erosionar y que se acumula hasta convertirse en una victoria en costo total de propiedad en cualquier flota real. Donde corresponde, también reportamos la **Arc Pro B50 (16 GB, 70 W)** como un tercer punto de datos de eficiencia, de menor potencia. Presentamos nueve figuras y un modelo completo de TCO, junto con una evaluación franca de las brechas restantes del ecosistema de software, que se están cerrando con rapidez. Este es un caso de *segunda fuente para el nivel de valor* — no una afirmación contra el silicio de frontera H100/B200.

1 Resumen ejecutivo

Tesis. Para el nivel asequible de inferencia y fine-tuning, donde se concentra el valor de la IA moderna, Intel Arc Pro entrega throughput de clase NVIDIA con ∼1.5–2×

el rendimiento por watt y VRAM sustancialmente más barata — la segunda fuente en eficiencia, valor y soberanía de cómputo, no un rival que destrone a la NVIDIA de frontera.

Medimos una **Intel Arc Pro B70 (32 GB, Battlemage)** cara a cara frente a una **NVIDIA RTX 3090 (24 GB, Ampere)** en la plataforma distribuida de inferencia/entrenamiento ColabHive, a lo largo de seis cargas de trabajo reales de IA (serving de LLM, difusión de imágenes y fine-tuning), con **throughput atribuido por dispositivo y medición de energía de extremo a extremo**. Los hallazgos:

- **Paridad o mejor en IA moderna limitada por cómputo.** En las cargas de trabajo que dominan el gasto actual en IA — serving de LLM, fine-tuning **LoRA *y* QLoRA de 4 bits**, y Stable Diffusion XL — la B70 entrega **95–112 % del throughput de la RTX 3090** consumiendo **56–64 % de su potencia**, es decir, **∼1.45–2.0× mejor rendimiento por watt**. (Serving de LLM es la única celda donde la B70 queda ligeramente *detrás* en bf16 — 95–97 % — pero con ∼1.5× la eficiencia. Y **la cuantización invierte incluso eso**: con **AWQ de 4 bits, la precisión que los operadores realmente despliegan, la B70 *supera* a la 3090 por 19–26 %** en serving de LLM, §4.10. Las dos celdas de inferencia llevan K=3 repeticiones; las celdas de entrenamiento son estimaciones puntuales pareadas.)

- **La difusión es una victoria rotunda de Intel.** La generación de imágenes con SDXL entrega **∼12 % más throughput** (≈11 % menos latencia por imagen) en la B70 *y* usa **la mitad de la energía por imagen** (2.00× img/J), de forma repetible a lo largo de K=3 corridas (±1 desviación estándar muestral).

- **Potencia estructuralmente menor — el hallazgo principal.** En *todas* las cargas de trabajo medidas, la B70 consumió **44–80 % de la potencia de la 3090** — no existe carga de trabajo, rápida o lenta, en la que consuma más. El rendimiento por watt no es un artefacto de medición aislado; es un resultado consistente y de amplio margen en cada carga de trabajo que medimos, y es el eje sobre el que gira todo el caso económico (§5). Para un operador que paga la factura eléctrica, ese es el número que se acumula.

- **Integrado en producción, no una demo de laboratorio.** Esto no es un benchmark aislado sobre una tarjeta prestada. Ambos fabricantes se sirven en producción sobre el mismo orquestador ColabHive + runtime por nodo; la telemetría de energía que produjo cada número de §4 ya está cableada en toda la flota (`node_power_samples` agnóstico del fabricante, node-runtime 0.10.119), y la corrección de QLoRA en Arc (§4.7) está integrada en un lanzador de entrenamiento de producción (node-runtime 0.10.121). El stack de Arc se distribuye, sirve y se auto-reporta hoy.

- **VRAM más barata — con fuerza frente a una 3090 *nueva*, modestamente frente a una *usada*.** Contra el precio de lanzamiento de $1,499 de la 3090, Intel Arc Pro cuesta **∼0.4× el $/GB de VRAM**; contra una 3090 usada (∼$700–1,050, el comparador realista para una pieza de 2020) la ventaja se estrecha a **∼0.6–0.8×** (§5 ahora cotiza la propia B70, no solo las más baratas B60/B50). En cualquier caso, la B70 compra más VRAM *por watt*, y ese margen (headroom) es *habilitante*: a precisión bf16 completa, la B70 de 32 GB sirve un modelo de 14 B que agota la memoria en la 3090 de 24 GB (§4.8) — aunque un 14 B *cuantizado* sí cabe en una 3090, así que esta es una ventaja de **simplicidad** a precisión completa, no una ventaja absoluta.

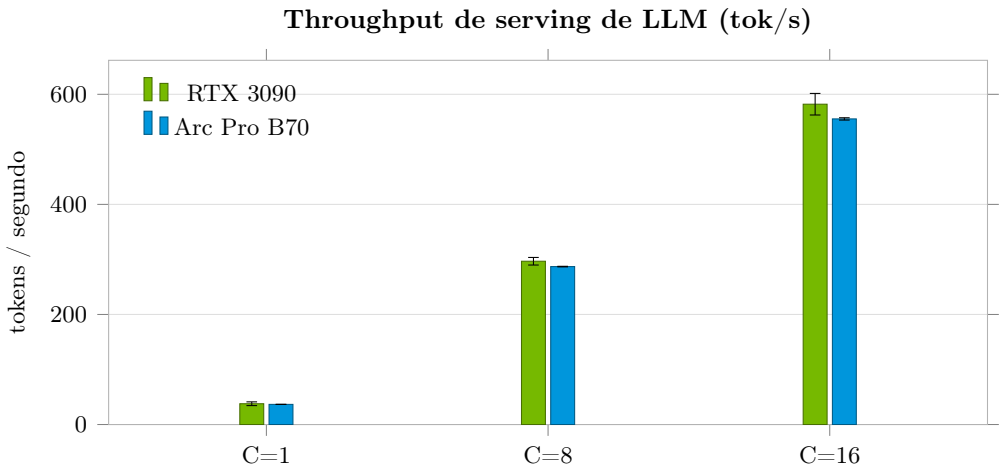

Figura 1: Throughput de serving de LLM (Qwen2.5-7B, vLLM bf16/eager), media de K=3 ±1 desviación estándar. La B70 sostiene 95–97% de la 3090. B50 omitida: 7B en bf16 agota la memoria (OOM) en sus 16 GB.

- **Límites honestos.** La B70 es significativamente más lenta en cargas de trabajo de operadores pequeños (TabNet, tabular profundo: la 3090 es ∼2.2× más rápida). El gradient boosting clásico (XGBoost/CatBoost) no tiene backend para GPU de Intel y permanece en CPU. (QLoRA de 4 bits fue *inicialmente* reportado como no funcional; al reinvestigarlo, **funciona en Battlemage** — y es ∼7 % más rápido que la 3090 — una vez que bitsandbytes recibe el selector de dispositivo SYCL correcto, §4.7. Fue un problema de configuración de una sola línea, no una brecha de kernels.) Los límites restantes son brechas del ecosistema de software con una trayectoria upstream clara y de rápido avance — no limitaciones del silicio.

Conclusión: para el **nivel asequible de inferencia y fine-tuning** que sirve ColabHive — *no* el segmento de centros de datos de frontera que dominan H100/B200 — Intel Arc Pro es una alternativa a NVIDIA creíble y eficiente en potencia y costo, *exactamente donde se concentra el valor de la IA moderna para ese nivel* (inferencia de LLM + difusión, fine-tuning LoRA/QLoRA). La economía principal es más fuerte frente a los precios de NVIDIA nueva y más modesta frente al mercado de usados; las ventajas de rendimiento por watt y de VRAM a precisión completa se sostienen en cualquier caso. Las brechas restantes están en el stack de software y se cierran versión tras versión.

2 Contexto y motivación

2.1 La dependencia de NVIDIA

El mercado de cómputo para IA es el más concentrado de la computación moderna. NVIDIA despachó un estimado del **98 % (3.76 M de 3.85 M de unidades) de las GPUs de centro de datos en 2023** (TechInsights, vía HPCwire). La dependencia se refuerza por el **lock-in de CUDA**, que es tanto contractual como técnico: el **EULA de CUDA (§1.2 "Limitations", ítem 8) prohíbe hacer ingeniería inversa o traducir la salida de los elementos del SDK de CUDA para apuntar a una plataforma que no sea de NVIDIA**, y el proyecto ZLUDA de CUDA-sobre-otro-hardware fue dado de baja a pedido de AMD en agosto de 2024. La oferta y el precio agravan el problema: se ha reportado que los aceleradores de generación Blackwell están

agotados con ∼12 meses de anticipación (gerencia de NVIDIA, oct. 2024), con Jensen Huang citando **$30,000–40,000 por GPU** (luego aclaró que NVIDIA vende sistemas completos, no chips sueltos, por lo que el precio por GPU varía).

Para una capa de infraestructura cuya misión es dar a desarrolladores, startups e investigadores — particularmente en LATAM — acceso a cómputo **asequible, disponible y eficiente en energía**, la dependencia de un único proveedor es el riesgo estratégico central. Una segunda fuente creíble no es opcional; es la tesis.

Por qué la línea base RTX 3090. Los lectores que vienen de los materiales de lanzamiento de Intel — que comparan la serie Arc Pro B contra tarjetas NVIDIA más nuevas de clase workstation (clase RTX PRO 4000) — pueden preguntarse por qué este paper usa como referencia una RTX 3090. La elección es deliberada, por tres razones. *Segmento:* la 3090 es la tarjeta NVIDIA más representativa en el segmento builder / laboratorio pequeño práctico disponible en nuestra flota de producción; el siguiente escalón NVIDIA en nuestro entorno es infraestructura de clase H100/H200, que sería la comparación equivocada para un estudio de Arc Pro B70/B50 enfocado en accesibilidad. *Legibilidad:* con 24 GB de VRAM, soporte CUDA maduro y años de benchmarking comunitario, la 3090 es un punto de referencia que los lectores pueden mapear fácilmente a otros segmentos de NVIDIA. *Realidad de acceso regional:* para muchos desarrolladores independientes, laboratorios pequeños y operadores regionales en LATAM, la opción NVIDIA práctica es una tarjeta usada de clase 3090, no un SKU actual de centro de datos o workstation. Los números regionales lo hacen concreto. La RTX 4000 Ada (20 GB) — el segmento workstation contra el que comparan los materiales de lanzamiento de Intel — se vende a través de canales oficiales brasileños a R$ 17,300–19,230, es decir, ≈ US$ 3,400–3,700 al tipo de cambio de julio de 2026, aproximadamente **2.7–3× su MSRP de lanzamiento en EE.UU. de US$ 1,250**; la brecha es estructural (Brasil aplica un **impuesto de importación del 60 %** sobre compras internacionales más un ICMS estatal del 17–20 %, y el propio gobierno de Argentina documentó electrónica de consumo de gama alta a **∼2.5× los precios de EE.UU.** en 2025). La nube tampoco es una vía de escape: un censo de 2024 con revisión por pares sobre regiones de nube pública con GPU encontró que "América Latina aloja un total de cinco regiones de nube habilitadas para GPU, pero ninguna de ellas ofrecía GPUs más potentes que la V100 de 2017"; a mediados de 2026, la única oferta de GPU de Google Cloud en Sudamérica sigue siendo la T4 de clase 2018 (São Paulo), y las instancias GPU de AWS en São Paulo cuestan ∼70 % más que el precio equivalente en us-east-1. En ese entorno, una 3090 usada frecuentemente tiene un precio cercano, o superior, al de una Arc Pro nueva, mientras ofrece menos VRAM y (en las cargas de trabajo que medimos) peor eficiencia energética. La 3090 no fue elegida como el diseño de referencia profesional actual de NVIDIA; fue elegida como una línea base práctica y regionalmente relevante: una tarjeta CUDA que todavía es obtenible, ampliamente comprendida y representativa de aquello contra lo que los builders de IA sensibles al costo en la región realmente pueden comparar.

2.2 Por qué Intel Arc

La serie Arc Pro B de Intel ("Battlemage") y la iniciativa **Project Battlematrix** (hasta **8× Arc Pro B60 → 192 GB de VRAM**, con el objetivo declarado por Intel de **modelos de 70 B+ parámetros**) apuntan precisamente al nicho de workstation de IA / servidor de inferencia, con un stack de software containerizado (`llm-scaler`: vLLM-XPU, ComfyUI, SGLang). Las preguntas abiertas para un operador de producción son empíricas: *¿Qué tan cercano es el throughput real? ¿Cuál es el perfil real de energía y costo? ¿Dónde sigue doliendo el ecosistema de software?* Este

paper las responde con datos medidos sobre hardware de grado de producción.

2.3 Escalamiento a Project Battlematrix (8×B60 → 192 GB, 70 B+ en el nodo)

La economía por tarjeta de este paper es la *unidad* de una historia más grande. El **Project Battlematrix** de Intel empaqueta **8× Arc Pro B60 en 192 GB de VRAM agregada** en un solo chasis de workstation/servidor. Esa envolvente de capacidad es exactamente lo que necesita un **modelo de clase 70 B en bf16 completo**: ∼140 GB de pesos más la caché KV caben **en el nodo**, sin sharding entre hosts — una clase de modelo que hoy obliga a una caja NVIDIA multi-GPU (típicamente 2–4× tarjetas de clase A100/H100) a un múltiplo del consumo eléctrico. Los hechos medidos por tarjeta individual en §4 son los bloques constructivos de esa afirmación: una B70 ya sirve un 14 B en bf16 completo en una sola tarjeta (§4.8), y la eficiencia vLLM-XPU por tarjeta que medimos (95–97 % del throughput LLM de la 3090 a ∼1.5× los tokens/joule, §4.2) es la economía unitaria que, *si compone*, escala a un nodo de clase 70 B a una fracción del presupuesto de potencia del multi-GPU de NVIDIA.

Advertencia honesta — y un pedido concreto. No hemos medido el escalamiento tensor-parallel (TP) ni el ancho de banda entre tarjetas en una topología de clase Battlematrix. La eficiencia TP multi-tarjeta en Arc está genuinamente sin probar en nuestras manos, y observamos una **falla real de TP multi-tarjeta en dual-B70 en el stack vLLM-XPU actual** (TP>1 todavía no corre limpio entre dos tarjetas Arc en nuestro entorno, §6/§8). Por lo tanto, la afirmación "8×B60 sirve 70 B en el nodo" es una *proyección hacia adelante a partir de datos sólidos de tarjeta individual*, no un resultado medido — y la brecha entre ambos es precisamente el trabajo de validación conjunta que una unidad Battlematrix en nuestro laboratorio nos permitiría cerrar. **Conclusión: la economía unitaria por tarjeta individual está probada y es favorable; lo único que se interpone entre ella y un 70 B en el nodo eficiente en potencia es la validación de TP multi-tarjeta — una unidad de validación Battlematrix nos permitiría cerrar esa brecha en conjunto: medir el escalamiento TP, arreglar el path de TP como arreglamos QLoRA, y reportar los resultados de vuelta con el mismo rigor.**

3 Plataforma y metodología

3.1 ColabHive

ColabHive es una plataforma distribuida de inferencia/entrenamiento (orquestador + runtime por nodo + dispatch consciente del vendor). Ya sirve GPUs Intel Arc en producción vía una imagen `inference-ipex` construida sobre la base **propia de Intel** `intel/llm-scaler-vllm:0.14.0-b8.3.1` (vLLM-XPU, `torch 2.10.0+xpu`, compute-runtime 26.09). La telemetría de energía de la plataforma (descrita abajo) fue extendida para este estudio y ahora está cableada de extremo a extremo.

3.2 Hardware bajo prueba

Host: Ubuntu 24.04, kernel 6.17, driver `xe`, compute-runtime 26.05+, Resizable BAR habilitado. Todos los benchmarks de GPU fijan una **única GPU dedicada** por lado; los demás modelos fueron drenados de la GPU objetivo para evitar contención.

Rol	Nodo	GPU	VRAM	CPU del host
Intel	IAC001	**Arc Pro B70** (`0xe223`, BMG-G31)	**32 GB** GDDR6-ECC	i7-10700 (8c)
Intel (2.ª)	IAC001	Arc Pro B50 (`0xe212`)	16 GB	—
NVIDIA	AUR001	RTX 3090 (GA102)	24 GB GDDR6X	Xeon E5-2680 v4 (28c)

Tabla 1: Hardware bajo prueba. Los 32 GB de la B70 fueron confirmados (32,656 MB) vía `torch.xpu`.

3.3 Cómo medimos (y cómo evitamos medir lo incorrecto)

Un compromiso metodológico central de este estudio es la medición **atribuida al dispositivo, aislada e instrumentada en energía**. Tres trampas que evitamos explícitamente:

1. **Contaminación CPU-vs-GPU.** En un nodo con GPU, mucha inferencia en realidad se ejecuta en la CPU (specialists, herramientas, fallbacks). La telemetría agregada de la plataforma por lo tanto mezcla dispositivos y modelos y *no* es una comparación de GPU válida. Cada número en §4 proviene de una **corrida controlada de una carga de trabajo idéntica en una GPU conocida y dedicada**, verificada como residente en GPU (contadores de generación de vLLM en `/metrics`, utilización de GPU, consumo de energía).
2. **Fuente de verdad del throughput.** El throughput LLM se toma del propio contador de vLLM `vllm:generation_tokens_total` (delta sobre la ventana medida), no de estimaciones del lado del cliente. La longitud de salida se fija con `ignore_eos` de modo que ambos fabricantes procesen trabajo *exactamente* idéntico.
3. **Atribución de energía.** La energía se mide en el dispositivo:
   - **Intel:** el contador acumulativo de energía en sysfs del driver `xe` `.../hwmon/.../energy1_input` (microjoules, monotónico) — descubierto por tarjeta vía PCI device id, exacto por construcción.
   - **NVIDIA:** donde está disponible, el contador acumulativo de energía exacto de NVML `nvmlDeviceGetTotalEnergyConsumption` (mJ, monotónico) — el análogo simétrico del contador Xe de Intel — es la cifra **primaria** (usada para QLoRA, §4.7). Para las celdas de serving LLM y SDXL, la energía es la integral de `nvidia-smi power.draw` a 200 ms; la integración trapezoidal de muestras instantáneas puede sesgar levemente frente al contador exacto — una asimetría de medición que señalamos en §8.
   - La ventana de energía está delimitada por timestamps `MEASURE_START`/`MEASURE_END` emitidos por la propia carga de trabajo (el host y el container comparten el reloj del kernel), de modo que la energía se atribuye *solo a la fase medida* (warmup y carga del modelo excluidos).

Controles de equidad: modelo idéntico, prompts/entradas idénticos, **bf16 + eager** en ambos fabricantes (ningún fabricante recibió fp8 ni un fast-path compilado que el otro no tuviera), `max_model_len`, batch y cantidad de steps idénticos.

Cableado de producción: las mismas fuentes de energía ahora se recolectan automáticamente por el runtime del nodo (`gpu_manager.collect_power_energy()` → heartbeat → tabla `node_power_samples`; NVML en NVIDIA, sysfs de Xe en Intel, RAPL best-effort en CPU; agnóstico al vendor,

incluido en node-runtime 0.10.119). Validado en vivo en toda la flota (ambos fabricantes reportando). Así que el rendimiento por watt ya no es una medición puntual — es telemetría de plataforma de aquí en adelante.

Advertencia (declarada de entrada): las dos celdas principales de **inferencia** (serving LLM §4.2, SDXL §4.3) son **K=3 repeticiones con ±1 desviación estándar muestral (n=3)**. Las celdas de entrenamiento (LoRA §4.4, QLoRA §4.7) son corridas medidas pareadas reportadas como **estimaciones puntuales** (repetidas, pero no afirmamos intervalos de confianza estrechos en el benchmark corto de 50 steps). SDXL-UNet-LoRA (§4.5) y TabNet (§4.6) son corridas únicas. **Advertencia de muestreo importante:** "K=3" controla el ruido *entre corridas* sobre las *mismas dos tarjetas físicas* — **no** controla la variación por lotería de silicio / fabricante de placa / térmica (N=1 hardware por vendor); un segundo dispositivo y un segundo tamaño de modelo son los próximos pasos (§8).

4 Resultados

Todas las cifras fueron medidas el 2026-06-21. "B70 %" = throughput de la B70 ÷ throughput de la 3090. "perf/watt" = B70 ÷ 3090 en la métrica de eficiencia de la carga de trabajo (tokens/J, img/J, steps·rows por J).

4.1 Matriz resumen

Carga de trabajo	Tipo	Thr. B70 vs 3090	Perf/W B70 vs 3090
Gen. de imágenes SDXL	infer.	**112 %** (n=3)	**2.00**×
FT LoRA (LLM 7B)	entren.	**∼103 %**	**1.64**×
Serving de LLM (vLLM)	infer.	**95–97 %** (n=3)	**1.46–1.65**×
SDXL UNet LoRA	entren.	57 %	**1.29**×
TabNet (tab. profundo)	entren.	46 %	0.56×
QLoRA 4-bit (7B)	entren.	**107 %**	**1.89**×

Tabla 2: Matriz resumen. Veredictos: SDXL — Intel gana en ambos ejes; LoRA — ∼paridad + eficiencia; LLM — casi paridad + eficiencia; UNet-LoRA — más lenta pero con mejor perf/W; TabNet — victoria de la 3090 (punto débil de Intel); QLoRA — victoria de Intel (corrección del selector) + más rápida.

Hallazgo estructural: la B70 consumió **44–80 % de la potencia de la 3090** en *todas* las cargas de trabajo que corrieron en ambas.

4.2 Serving de LLM — Qwen2.5-7B-Instruct, vLLM (bf16, eager, 4096 ctx, 128 output tokens) — K=3

El throughput es el contador engine-side `vllm:generation_tokens_total` ÷ tiempo de reloj; media ± desviación estándar muestral sobre **3 repeticiones** por nivel de concurrencia. La Figura 1 grafica el throughput agrupado; la Tabla 3 presenta las cifras completas, incluida la energía.

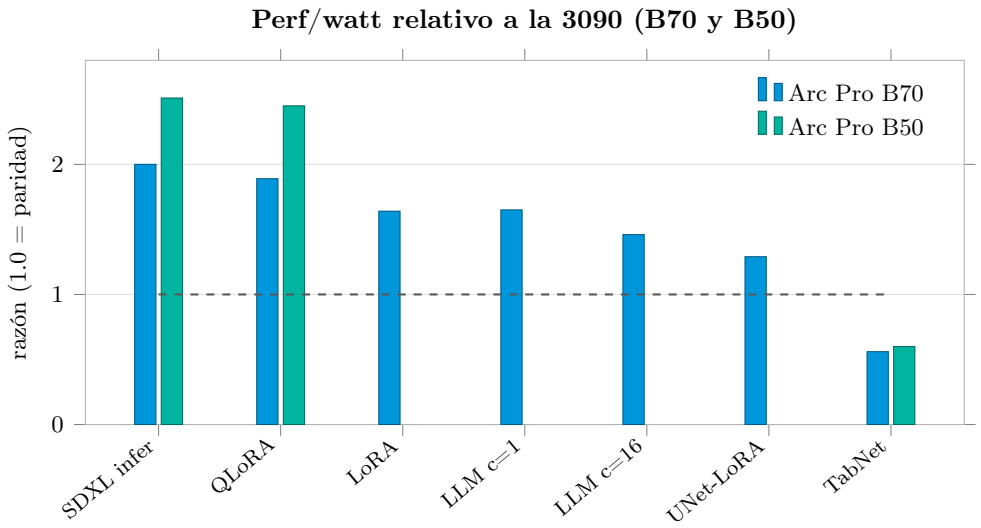

Figura 2: Rendimiento por watt relativo a la RTX 3090 (1.0 = paridad). Por encima de la línea punteada, la parte de Intel realiza más trabajo por joule. La B50 (tarjeta de 70 W) registra el mejor perf/watt de las tres en las cargas de trabajo que puede correr; no tiene barra de LoRA/LLM (esas hacen OOM en sus 16 GB).

Conc.	3090 tok/s	B70 tok/s	B70 %	B70 p/W
1	37.8 ± 3.4	**36.6 ± 0.1**	97 %	**1.65×**
8	296.7 ± 6.9	**287.0 ± 0.5**	97 %	**1.53×**
16	582.0 ± 19.6	**555.3 ± 2.1**	95 %	**1.46×**

Tabla 3: Serving de LLM, K=3 media ±1 desviación estándar. 3090 tok/J: 0.110/0.876/1.723; B70 tok/J: 0.182/1.344/2.511.

Bajo carga: **3090 ≈ 343 W, B70 ≈ 219 W (∼64 %)**. A lo largo del barrido, la **B70 sostiene 95–97 % del throughput de la 3090 con 1.46–1.65× los tokens/joule**, con una latencia p50 de la B70 ∼5–10 % mayor en el extremo superior del rango. Dos cosas destacan en los datos con K=3: (1) la **B70 es notablemente más consistente entre corridas** (desviación estándar $< 1\,\%$) que la 3090, cuyo throughput single-GPU varió 5–9 % entre barridos consecutivos (un leve descenso térmico en C=1, donde su IC 37.8 ± 3.4 contiene por completo el 36.6 de la B70); y (2) la mayor varianza de la 3090 se debe en parte a un leve descenso térmico bajo carga consecutiva — lo cual juega tanto *en nuestra contra* como a nuestro favor: la "casi paridad" en C=1 podría reflejar tanto un *bajo rendimiento de la 3090* como que la B70 la alcanzó, así que nos apoyamos en las celdas de mayor concurrencia (C=8/16, un limpio 95–97 %) para la lectura de paridad y tratamos C=1 como ruidosa. La B70 de 32 GB sobre un stack vLLM-XPU maduro alcanza **casi paridad con la 3090** (dentro de 3–5 %, ligeramente por detrás) — muy por delante del posicionamiento más débil en LLM que muestran las tarjetas B-series más chicas y limitadas por ancho de banda (p. ej. las B50/B60) en las primeras reseñas de consumo.

4.3 Generación de imágenes SDXL — 1024×1024, 30 pasos (8 imágenes/corrida × 3 corridas) — K=3

La B70 entrega ∼12 % más throughput (≈11 % menos latencia por imagen) y usa 2.00× menos energía por imagen, de manera repetible a lo largo de 3 corridas (B70 7.33 ± 0.03 vs 3090

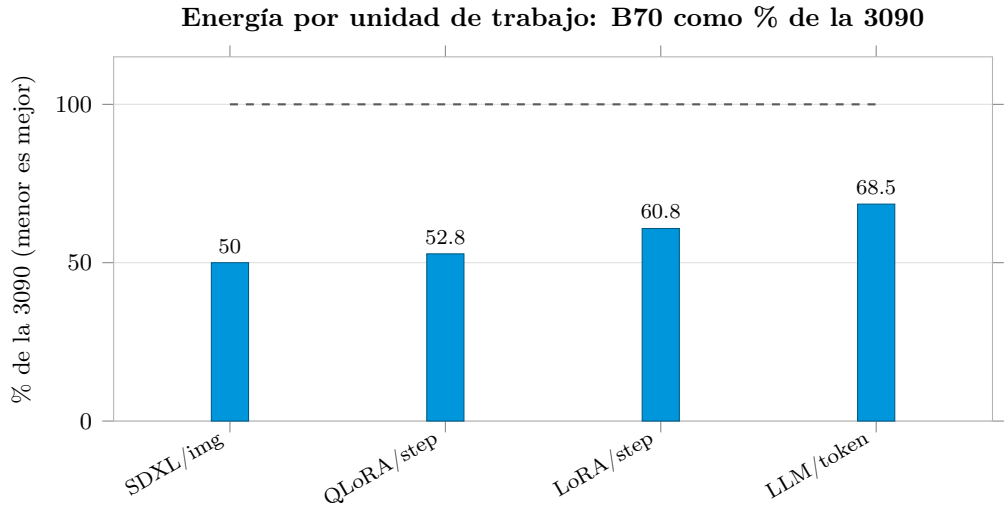

Figura 3: Energía de la B70 por unidad de trabajo como fracción de la de la 3090. La B70 usa aproximadamente entre la mitad y dos tercios de la energía.

	3090 (n=3)	**B70 (n=3)**
Latencia / imagen	8.21 ± 0.09 s	**7.33 ± 0.03 s**
Throughput	7.3 img/min	**8.2 img/min**
Energía / imagen	3,240 ± 37 J	**1,621 ± 6 J**
Potencia prom.	∼394 W	∼221 W

Tabla 4: Inferencia SDXL, K=3 media ±1 desviación estándar.

8.21 ± 0.09 s/img; ±1 desviación estándar muestral, n=3, claramente separadas) — el resultado insignia no es un artefacto de una sola corrida. La difusión está limitada por cómputo, con kernels grandes y regulares — el throughput FP16 de Battlemage brilla aquí. (Las reseñas independientes de SDXL generalmente ubican a Arc *por detrás* de las NVIDIA comparables en throughput bruto, aunque por delante en perf/$ y perf/W; la victoria absoluta en throughput de la B70 sobre este path de serving optimizado es, por lo tanto, un resultado notable y específico de la plataforma.)

4.4 Fine-tuning LoRA — Qwen2.5-7B, r=16 (q/k/v/o), seq 512, batch 2, 50 pasos (bf16)

	3090	**B70**
Throughput	1.90 st/s, 1,944 tok/s	**1.96 st/s, 2,007 tok/s**
Energía / paso	180.8 J	**110.1 J**
Potencia prom.	∼347 W	∼222 W

Tabla 5: Fine-tuning LoRA (estimaciones puntuales de corridas repetidas).

Entrenamiento LoRA: la B70 es ∼3 % más rápida con 1.64× mejor perf/watt. Estas son estimaciones puntuales de corridas repetidas; *no* afirmamos aquí un intervalo de confianza estrecho — la dispersión entre corridas en este benchmark corto de 50 steps es de unos pocos puntos porcentuales en ambos lados, así que la lectura honesta es "aproximadamente paridad en throughput, ∼1.6× la

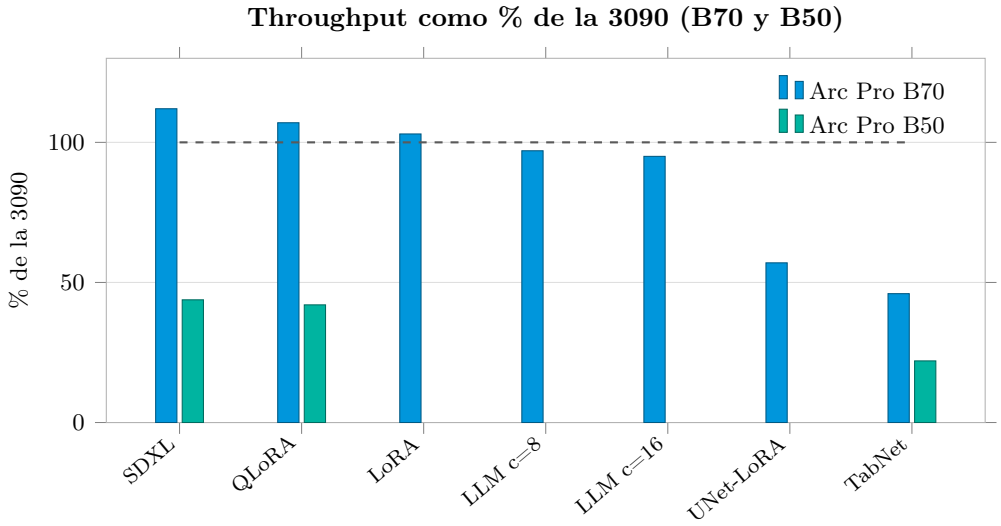

Figura 4: Throughput como porcentaje de la RTX 3090. La B70 gana en difusión y QLoRA y queda atrás en cargas de trabajo de operadores pequeños; la B50, más lenta y de bajo consumo, corre al ∼22–44 % de la 3090 (sin barra de LoRA/LLM — esas hacen OOM en sus 16 GB).

eficiencia". LoRA-fp16 es un path confiable de fine-tuning de LLM en Intel (ver §6).

4.5 Fine-tuning SDXL UNet LoRA — latents de 1024px, batch 2, 30 pasos (bf16, eager)

	3090	**B70**
Throughput	**1.625 steps/s**	0.930 steps/s
Energía / paso	208 J	**161 J**
Potencia prom.	∼346 W	∼152 W

Tabla 6: Fine-tuning SDXL UNet-LoRA (corrida única).

Aquí la 3090 es **1.75× más rápida** (su path backward CUDA sobre entrenamiento UNet eager crudo está más optimizado). En términos operativos, es un costo real — un job de UNet-LoRA corre al ∼57 % de la tasa de trabajo, es decir, tarda ∼1.75× más (menos jobs/hora/tarjeta) — así que **el perf/watt (1.29×, consumiendo solo el 44 % de la potencia) es aquí el premio consuelo, no una victoria en throughput.** (Contrástese con la *inferencia* SDXL en §4.3, donde el path de serving optimizado permitió a la B70 ganar en throughput de manera absoluta — la inferencia y el entrenamiento eager crudo son patrones de cómputo distintos.)

4.6 TabNet (tabular profundo) — 16k×64 sintético, 20 épocas — el punto débil honesto

La 3090 gana en ambos ejes (2.2× throughput, 1.8× perf/watt). TabNet consiste en muchos operadores pequeños con sincronización host-device frecuente; la XPU no se satura (87 W indica que la GPU no recibe trabajo suficiente), y el overhead per-kernel/eager domina — exactamente el patrón donde un stack CUDA maduro toma la delantera. Esta es una limitación real para cargas de trabajo de operadores pequeños, dicha sin rodeos.

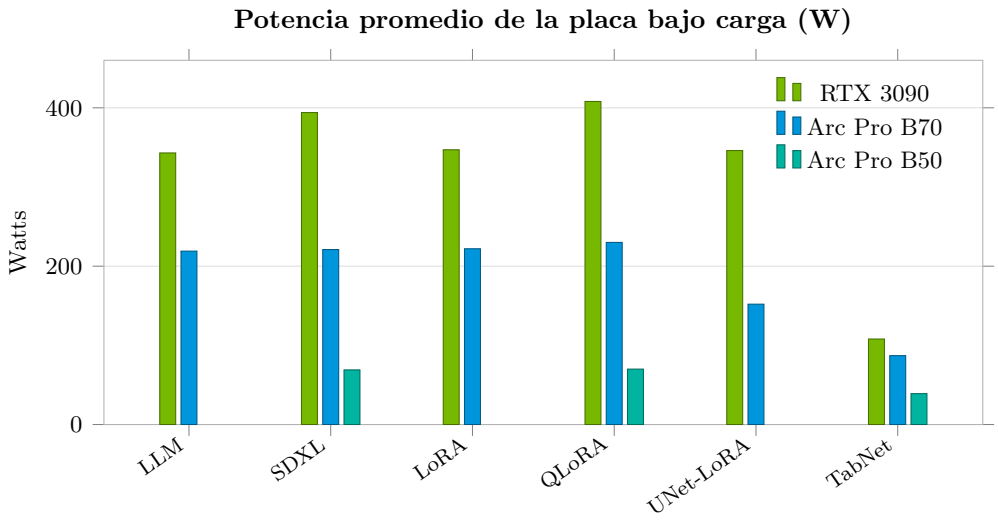

Figura 5: Potencia promedio de la placa. La B70 consume 44–80% de la potencia de la 3090 en todas las cargas de trabajo medidas; la B50, de clase 70 W, consume solo 39–70 W (sin barra de LoRA/LLM — esas hacen OOM en sus 16 GB) — una ventaja de eficiencia consistente de las partes de Intel en nuestras cargas de trabajo medidas.

	3090	**B70**
Throughput	**16,852 rows/s**	7,739 rows/s
Potencia prom.	∼108 W	∼87 W
Eficiencia	**159 rows/J**	89 rows/J

Tabla 7: TabNet tabular profundo (corrida única) — el punto débil honesto.

4.7 QLoRA 4-bit — corre en Battlemage con una corrección del selector de dispositivo (y supera a la 3090)

Una pasada anterior reportó el QLoRA de 4 bits como *fallido* en Battlemage; al reinvestigarlo, eso resultó ser un **diagnóstico erróneo**. La falla **no** es un kernel Triton-XPU faltante. bitsandbytes 0.49.2 despacha la cuantización de 4 bits a través de un op custom nativo (`torch.ops.bitsandbytes.quantize_4bit`), y bajo la convención de fijación de dispositivos de la plataforma `ONEAPI_DEVICE_SELECTOR=level_zero:0` ese op lanza un error SYCL `No device of requested type available` — la propia cola de kernels de bnb rechaza el selector de dispositivos Level-Zero (torch.xpu ve la GPU sin problema; la cola SYCL de bnb, compilada por separado, no). Configurar `ONEAPI_DEVICE_SELECTOR=*:gpu` (GPU de cualquier backend) permite que el kernel resuelva un dispositivo, y **el QLoRA de 4 bits entonces entrena de extremo a extremo en la B70**. Es genuinamente de 4 bits: el 7B cuantizado ocupa **5.45 GiB en ambos fabricantes** (bf16 serían ∼15 GiB), y la desaceleración de ∼2.2× vs LoRA-fp16 (§4.4) es exactamente el impuesto esperado de decuantización de 4 bits.

La B70 es ∼7 % más rápida en QLoRA y 1.89× más eficiente energéticamente — el mismo patrón que LoRA-fp16, medido en una sesión pareada con contadores de energía exactos (sysfs Xe en Intel, energía total NVML en NVIDIA). Esto elimina una limitación de titular: la ruta confiable de fine-tuning de LLM en Intel es **tanto LoRA-fp16 como QLoRA de 4 bits**. El trabajo restante es puramente una tarea de integración de plataforma — cablear el selector `*:gpu` en la ruta de lanzamiento de entrenamiento de Intel para cargas de trabajo bnb-4bit (el runtime del

QLoRA-NF4	3090	B70
Throughput	0.889 st/s, 911 tok/s	**0.949 st/s, 972 tok/s**
Energía / paso	459.0 J	**242.3 J**
Potencia prom.	∼408 W	∼230 W
VRAM (4 bits)	5.45 GiB	5.45 GiB

Tabla 8: QLoRA-NF4 (Qwen2.5-7B, r=16, B=2, S=512, 50 pasos); contadores de energía exactos en ambos lados.

nodo actualmente fija `level_zero:<idx>`); ver §6.

Nota metodológica (asumimos la responsabilidad): un resultado que publicamos como una *limitación de hardware* resultó ser un problema de configuración de una línea. Eso actualiza nuestro prior — ahora tratamos un único resultado negativo-para-Intel como **provisional a la espera de una auditoría de configuración**. Bajo ese estándar, el punto débil de TabNet (§4.6) *aún no* está auditado y debe leerse como "no auditado, plausiblemente mejorable" en lugar de un límite de silicio establecido.

Contribución: desbloqueo del QLoRA de 4 bits en toda la serie Arc B

Este hallazgo es lo suficientemente significativo como para nombrarlo **contribución de este trabajo**, y no enterrarlo como una nota al pie de §4.7.

La sabiduría convencional está equivocada. El consenso público en 2026 — repetido en foros, hilos de issues y guías de "¿corre en Intel?" — es que **"bitsandbytes / QLoRA de 4 bits no corre en Intel Arc"**. Mientras tanto, las propias notas de versión de bitsandbytes **listan oficialmente soporte para XPU**. Ambas cosas no pueden ser completamente ciertas, y la brecha entre ellas es donde los operadores se atascan y concluyen (como hicimos nosotros inicialmente) que el hardware no puede hacerlo.

La trampa es una colisión de documentación. La *propia* guía multi-GPU de Intel recomienda fijar dispositivos con `ONEAPI_DEVICE_SELECTOR=level_zero:N` — y ese es exactamente el selector bajo el cual el op SYCL de 4 bits de bitsandbytes, compilado por separado, lanza `No device of requested type available`. De modo que un operador que sigue la documentación multi-GPU de Intel *al pie de la letra* e instala el bitsandbytes con soporte XPU se topará con una falla dura que *parece* un kernel faltante — cuando en realidad `torch.xpu` ve la GPU sin problema y solo la cola SYCL de bnb rechaza el selector Level-Zero. Dos piezas de configuración recomendadas por Intel, correctas por separado, se combinan en un modo de falla que parece un kernel faltante.

Nuestra causa raíz y solución. Configurar `ONEAPI_DEVICE_SELECTOR=*:gpu` (GPU de cualquier backend, de modo que la cola de kernels de bnb pueda resolver un dispositivo) y fijar la tarjeta específica con `ZE_AFFINITY_MASK=<idx>` en lugar del selector Level-Zero. Con ese único cambio, **el QLoRA de 4 bits entrena de extremo a extremo en la B70** (∼7 % más rápido que la 3090 con 1.89× la eficiencia, genuinamente de 4 bits a 5.45 GiB — §4.7), y confirmamos que **generaliza a toda la serie B**: la solución idéntica (`*:gpu` + `ZE_AFFINITY_MASK=1`) funciona en la B50 (§4.9). Dado que la solución es a nivel de selector y agnóstica a la tarjeta, **desbloquea el QLoRA de 4 bits en toda la serie Arc B — incluida la topología de 8 tarjetas del propio Project Battlematrix de Intel**, donde la fijación por tarjeta es obligatoria y la trampa de `level_zero:N` probablemente aparecerá para cualquiera que siga la documentación multi-GPU estándar.

Está en producción. La solución no es un hack de notebook — está cableada en un lanzador de entrenamiento de producción (**node-runtime 0.10.121**), de modo que cada trabajo de entrenamiento en Arc que ColabHive despacha usa el selector correcto automáticamente.
Lo estamos devolviendo. Estamos reportando esto upstream a **bitsandbytes** (la firma de la falla + la causa raíz del selector) y a la **documentación de IPEX / llm-scaler de Intel** (para que la guía de fijación multi-GPU y la ruta bnb de 4 bits dejen de colisionar). Un borrador de reporte de bug listo para presentar está en el **Apéndice C**.
Conclusión: no solo consumimos el stack de Intel — lo arreglamos y contribuimos de vuelta. La limitación más citada de "Arc no puede hacer QLoRA" es, en nuestras manos, una corrección de configuración de una línea resuelta, llevada a producción y reportada upstream, que escala a toda la serie B y a Battlematrix.

4.8 Margen de VRAM — un modelo que la 3090 no puede alojar a *precisión completa*

A precisión **bf16** completa, la B70 de 32 GB aloja modelos de tamaño medio que no caben en una 3090 de 24 GB. Esta es una ventaja de **simplicidad/margen (headroom)**, *no* una brecha de capacidad absoluta — un 14 B cuantizado (AWQ/GPTQ/fp8) cabe bien en una 3090 y es la forma estándar de servirlo en 24 GB. La afirmación acotada y honesta: en la configuración **bf16 sin cuantización** idéntica usada en el resto de este paper, cargar **Qwen2.5-14B-Instruct** hace que ambos fabricantes intenten un trabajo idéntico (Figura 8).

	RTX 3090 (24 GB)	**Arc Pro B70 (32 GB)**
Carga de pesos bf16	**falla** — CUDA OOM (23.49 / 23.56 GiB usados, no puede asignar los siguientes 270 MiB)	**carga y sirve** (31.1 GiB usados)
¿Completion real?	no	**sí** (texto coherente)

Tabla 9: Carga de pesos bf16 de Qwen2.5-14B: la 3090 se queda sin memoria; la B70 carga y sirve.

El valor operativo es la **simplicidad**: en la B70 se sirve o se hace fine-tuning fp16 de un modelo de ∼13–14 B en una sola tarjeta sin pipeline de cuantización, sin offload, sin segunda GPU. En una 3090 el mismo modelo necesita un paso de cuantización — aceptable para inferencia, más complicado para fine-tuning a precisión completa. De modo que el encuadre defendible *no* es "NVIDIA no puede correr un 14 B" (puede, cuantizado) sino "**Arc Pro brinda margen a precisión completa** para toda una clase de modelos de tamaño medio que una tarjeta de 24 GB obliga a cuantizar o particionar".

4.9 La Arc Pro B50 — eficiencia a 70 W (un tercer punto de datos)

También corrimos todo lo que cabe en la segunda tarjeta Intel del nodo, la **Arc Pro B50 (16 GB, 70 W TBP, sin alimentación externa)** — mismos selectores (`level_zero:1`; `*:gpu`+`ZE_AFFINITY_MASK=1` para QLoRA, confirmando que la solución de §4.7 es general, no específica de la B70), mismo método de energía (contador Xe de la tarjeta 2). La B50 es el **piso de eficiencia/potencia** de las tres piezas:

Dos conclusiones. **(1) Lenta pero asombrosamente eficiente.** La B50 es 2.3–4.5× más lenta

Carga de trabajo	RTX 3090 (350 W)	Arc B70 (32 GB)	Arc B50 (16 GB, 70 W)
SDXL / imagen	8.21 s · 3,240 J · ∼394 W	7.33 s · 1,621 J · ∼221 W	18.74 s · **1,291 J** · ∼**69 W**
QLoRA-NF4 / paso	0.889 st/s · 459 J · ∼408 W	0.949 st/s · 242 J · ∼230 W	0.373 st/s · **187 J** · ∼**70 W**
TabNet	16,852 rows/s · ∼108 W	7,739 rows/s · ∼87 W	3,700 rows/s · ∼**39 W**
Serving de LLM (7B, bf16)	sí	sí	**OOM** (16 GB)
LoRA-fp16 (7B)	sí	sí	**OOM** (16 GB)

Tabla 10: Comparación a tres bandas incluyendo la Arc Pro B50. "sí" = corre; "OOM" = sin memoria (out-of-memory) en la B50 de 16 GB a bf16.

que la 3090, pero consume solo **39–70 W** y registra la **menor energía por unidad de trabajo de las tres piezas** en las cargas de trabajo que puede correr — SDXL a **1,291 J/imagen (0.40× la 3090, por debajo incluso de la B70)** y QLoRA a **187 J/paso (0.41× la 3090)**. Para despliegues con restricciones de potencia o densidad (muchas tarjetas por chasis, sin conector de alimentación externa), ese es todo el argumento de venta. **(2) Los 16 GB son la restricción determinante.** Un modelo de 7 B a bf16 **se queda sin memoria** en la B50 tanto para serving como para LoRA-fp16 — debe cuantizarse para caber; QLoRA-4bit (5.45 GiB) corre cómodamente. De modo que la B50 es una tarjeta de *inferencia cuantizada y QLoRA*, y los **32 GB de la B70 son precisamente lo que compra el margen a precisión completa** (§4.8). Las dos piezas de Intel son complementarias: la B50 para eficiencia por watt en el extremo bajo, la B70 para capacidad a precisión completa y throughput cercano al de la 3090.

4.10 Inferencia INT8 en Battlemage — conectando la especificación con el throughput

La economía de §5 se apoya en la ventaja publicada de **TOPS/W INT8** de Battlemage (B70 1.60 vs 3090 0.81). Una especificación es una promesa, no una medición — así que intentamos convertirla en tokens/s entregados en la B70. La respuesta tiene dos mitades: la ruta de *cómputo INT8* está bloqueada por un kernel faltante, pero la ruta de 4 bits *estándar de despliegue* no solo funciona: **invierte el veredicto del serving de LLM a una victoria de la B70**.

(a) INT8 verdadero (W8A8) — bloqueado por un kernel vLLM-XPU faltante. Servir un modelo INT8 `compressed-tensors` W8A8 (que ejercitaría los 367 TOPS INT8) falla en la inicialización del engine en vLLM-XPU:

```
File .../quantization/kernels/scaled_mm/__init__.py, line 55,
    in choose_scaled_mm_linear_kernel
  for kernel in _POSSIBLE_KERNELS[current_platform._enum]:
KeyError: <PlatformEnum.XPU: 4>
```

vLLM **no tiene ningún kernel scaled-mm INT8 registrado para la plataforma XPU** — la tabla de despacho lleva entradas CUDA/ROCm/CPU/TPU pero no XPU. De modo que los TOPS INT8 publicados de la B70 **aún no son realizables a través de la ruta de serving INT8 de vLLM**: la ventaja económica en TOPS INT8 (§5) es una ventaja de especificación genuina

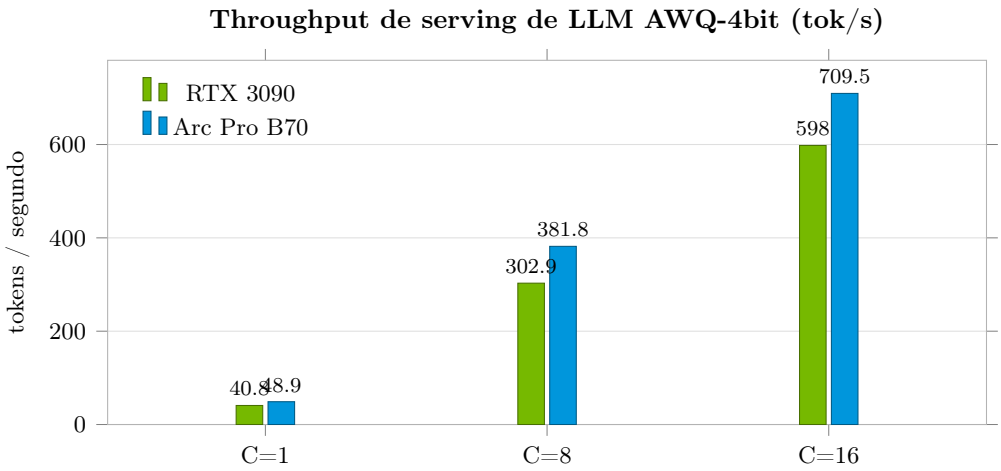

Figura 9: Serving de LLM AWQ-4bit (Qwen2.5-7B): la cuantización convierte la casi-paridad a bf16 (95–97%, Figura 1) en una victoria limpia de la B70 (+19–26%) a la precisión que los operadores despliegan.

que el stack de software actual no puede cobrar. Es una brecha concreta y reportable — análoga al problema del selector de QLoRA pero más profunda (un registro de kernel faltante, no una configuración), y de lleno en la trayectoria de cierre de §6.

(b) AWQ-4bit — funciona, y la B70 *supera* a la 3090. La precisión que los operadores realmente despliegan en tarjetas de 24–32 GB es 4 bits solo-pesos (weight-only) (AWQ/GPTQ). En vLLM-XPU esto se enruta por la ruta solo-pesos de IPEX, protegida tras una guarda de deprecación que evitamos con una sola flag (`-allow-deprecated-quantization` — un tercer desbloqueo menor del ecosistema que documentamos). Sirviendo el mismo AWQ Qwen2.5-7B, salida coherente en ambos fabricantes:

AWQ-4bit, Qwen2.5-7B	**3090 tok/s**	**B70 tok/s**	**B70 %**	**3090 tok/J**	**B70 tok/J**
C=1	40.8	**48.9**	**120 %**	0.255	0.325
C=8	302.9	**381.8**	**126 %**	1.913	2.351
C=16	598.0	**709.5**	**119 %**	3.506	**4.179**

Tabla 11: Serving de LLM AWQ-4bit (Qwen2.5-7B). A la precisión que los operadores realmente despliegan, la B70 aventaja a la 3090 por 19–26 % en throughput y 1.19× en tokens/joule.

La cuantización invierte el veredicto del serving de LLM. A bf16 (§4.2) la B70 queda en el 95–97 % del throughput de la 3090; a AWQ-4bit — el punto de operación realista — la B70 **aventaja por 19–26 % en throughput y 1.19× en tokens/joule**, mientras además recorta su propia latencia (p50 2.70 s vs 3.65 s bf16) y potencia (∼170 W vs ∼219 W). Cada fabricante corre su propio kernel AWQ (solo-pesos de IPEX en XPU vs Marlin en CUDA), de modo que esta es una comparación de mundo real de "lo que cada tarjeta realmente sirve" en lugar de una de mismo kernel — y cae a favor de la B70 (Figura 9).

Conclusión: la ventaja de especificación en *TOPS* INT8 de la B70 aún no es cobrable (vLLM no tiene kernel scaled-mm INT8 para XPU — una brecha corregible, reportada en el Apéndice C), pero la victoria cuantizada *realizable* ya es decisiva: a la precisión de 4 bits que la gente realmente despliega, la B70 **supera** a la RTX 3090 en serving de LLM (+19–26 %) y en eficiencia (1.19×), convirtiendo la única derrota de casi-paridad de §4.2 en una victoria limpia en el punto de operación

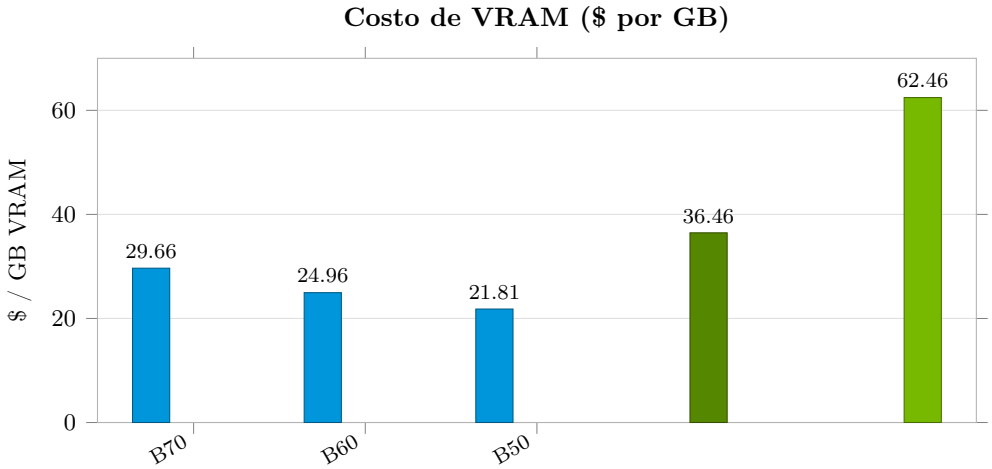

Figura 6: Costo de VRAM por GB. Intel es ∼2.5× más barata que una 3090 NUEVA, ∼1.2–1.7× más barata que una USADA.

que importa.

5 Economía

La capacidad de VRAM y el throughput de inferencia INT8 por dólar y por watt son las métricas que importan para democratizar el acceso a la IA. (Intel no publica TFLOPS BF16 densos para la línea Arc Pro, por lo que se usan los TOPS INT8 — que *sí* están publicados — para la comparación de economía de cómputo; BF16/$ bruto sería comparar peras con manzanas y se omite en lugar de estimarse.)

Métrica	**B70**	**B60**	**B50**	**3090**
Precio USD	**∼949**	∼599	349	1499 nueva
$/GB VRAM	29.66	24.96	21.81	62.46 / 36.5
GB/100W	13.9	12.0	22.9	6.86
INT8 TOPS/W	1.60	0.99	2.43	0.81
INT8 TOPS/$	0.39	0.33	0.49	0.19 / 0.33

Tabla 12: Economía. Las celdas de la 3090 muestran nueva / usada. TOPS INT8 (densos): B70 367, B60 197, B50 170, 3090 285.

La **B70 — la pieza efectivamente evaluada en §4 — ahora tiene precio en esta tabla**; borradores anteriores listaban solo las B60/B50 más baratas, lo cual era una crítica justa. Contexto de especificaciones: B70 32 GB GDDR6, 608 GB/s, ∼230 W TBP; B60 456 GB/s, 120–200 W; B50 224 GB/s, 70 W (sin alimentación externa); RTX 3090 24 GB GDDR6X, 936 GB/s, 350 W.

Leyendo la ventaja con honestidad. En **$/GB VRAM**, Intel Arc Pro está en **0.35–0.47× de una 3090 *nueva*** ($1,499) — el titular de "∼2.5–3× VRAM por dólar". Pero la 3090 es una pieza de 2020 que se compra *usada* (∼$700–1,050 ≈ $36.5/GB); frente a ese comparador realista la brecha se reduce a **∼0.6–0.8×**, es decir, aproximadamente **1.2–1.7× la VRAM por dólar**, no 3×. Donde el liderazgo de Intel es **robusto independientemente de la base de precio** es en **VRAM por watt** (1.75–3.3×) y **TOPS INT8/W** (1.2–3.0×) — eficiencia estructural que el mercado de usados no puede erosionar (Figura 7). El caso económico honesto: VRAM *modestamente* más barata que

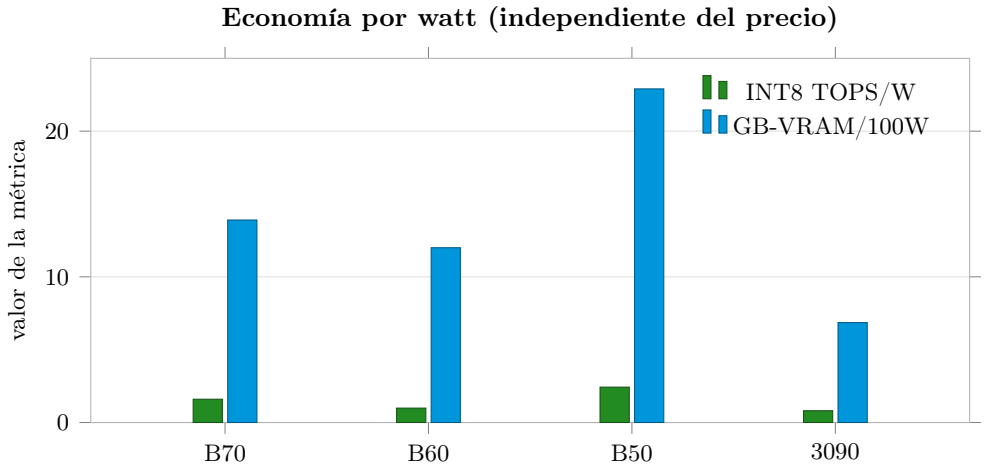

Figura 7: Eficiencia estructural que el precio del mercado de usados no puede erosionar: TOPS INT8 por watt y GB de VRAM por 100W.

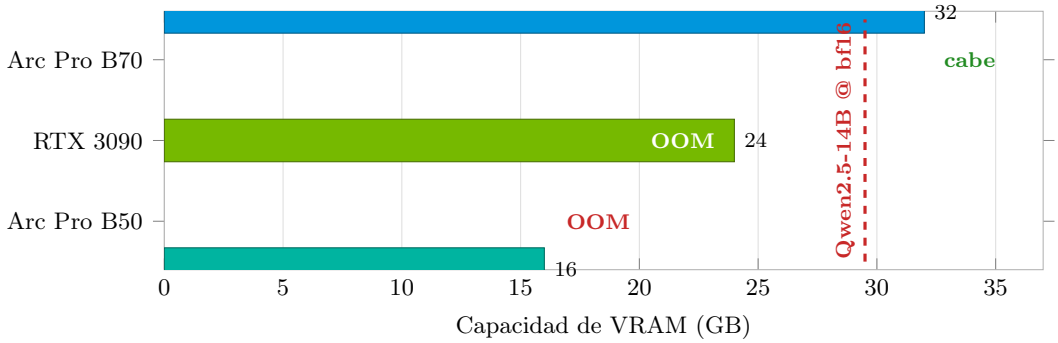

Figura 8: Un modelo 14B a precisión bf16 completa (∼29.5 GB) cabe en la B70 de 32GB y agota la memoria (OOM) tanto en la 3090 de 24GB como en la B50 de 16GB (la B70 usó 31.1 GB).

una 3090 usada, *mucho* más barata que una nueva, rendimiento por watt decisivamente mejor — y una pieza completamente nueva (garantía, drivers actuales, soporte) frente a una tarjeta de seis años en depreciación.

El precio de lista es la unidad equivocada. Lo que un operador realmente paga es el **costo por unidad de trabajo** a lo largo de la vida de la tarjeta — amortización del hardware *más* la electricidad para hacer el trabajo — y ahí es donde el rendimiento por watt medido en §4 se convierte directamente en dólares. El resto de §5 construye ese caso de costo total de propiedad (TCO) a partir de los datos medidos.

Supuestos de TCO (declarados una vez, usados a lo largo de §5). Amortización lineal del hardware a 3 años = **26,280 horas de operación**. Electricidad: **EE.UU. $0.15** / **Argentina $0.10** / **Alemania $0.30 por kWh**. Intensidad de carbono de la red ≈ **0.4 kg CO_2 / kWh**. Capital de hardware: **3090 usada $875**, **B70 $949**, **B50 $349** (la 3090 nueva de $1,499 se muestra donde agudiza el contraste). El throughput y los watts son las celdas medidas de §4. La fórmula completa y una nota de sensibilidad están en el **Apéndice A**.

5.1 Costo total por unidad de trabajo

$/unidad total = **hardware ÷ (unidades producidas durante la vida útil a utilización U)** + **energía/unidad × $/kWh**. El término de hardware se reduce con la utilización (una tarjeta amortizada sobre más trabajo es más barata por unidad); el término de energía no. Las dos cargas de trabajo principales de abajo se calculan a partir del throughput y la potencia de §4.

Serving de LLM (Qwen-7B, concurrencia C=16) — $ / millón de tokens @ US $0.15, 100 % de utilización:

Tarjeta	HW $/M-tok	Energía $/M-tok	Total $/M-tok
Arc B70	$0.0181	$0.0166	**$0.0347**
3090 (usada)	—	—	**$0.0401**
3090 (nueva)	—	—	**$0.0514**

Tabla 13: $/millón de tokens todo incluido, serving de LLM a C=16, US $0.15/kWh, 100 % de utilización.

La victoria de la B70 en serving de LLM es **puramente energética** — está ligeramente *detrás* en throughput bruto (§4.2), así que no gana en el término de hardware; gana porque cada token cuesta ∼la mitad de los joules. Eso hace que la ventaja en serving de LLM esté **condicionada por la utilización**: a utilización muy baja domina el término de hardware y la 3090 usada, de precio de lista más barato, puede tomar la delantera; a medida que sube la utilización domina el término de energía y la B70 se despega. El punto de cruce (crossover) es exacto:

> **Utilización de cruce U^* = $0.0429 / price_kWh.** La B70 supera a una 3090 usada en $/token por encima de **U* = 28.6 % de utilización a US $0.15/kWh**, **14.3 % en Alemania $0.30**, y **43 % en Argentina $0.10**. Cuanto más cara sea su energía y más ocupada esté su flota, más decisivamente gana Arc en serving de LLM — y una flota de inferencia en producción opera muy por encima del 28.6 %.

Generación de imágenes SDXL — $ / 1,000 imágenes @ US $0.15, 100 % de utilización:

Tarjeta	Total $/1,000 img	vs 3090 usada	vs 3090 nueva
Arc B50	**$0.123**	—	—
Arc B70	**$0.141**	**33 % más barata**	**47 % más barata**
3090 (usada)	**$0.211**	—	—
3090 (nueva)	**$0.265**	—	—

Tabla 14: $/1,000 imágenes SDXL todo incluido, US $0.15/kWh, 100 % de utilización.

SDXL es diferente en esencia: la B70 gana **tanto** en el término de hardware por imagen (es *más rápida* por imagen, §4.3) **como** en el término de energía (la mitad de los joules). Cuando una tarjeta es más barata en cada componente del costo, **no hay punto de cruce (crossover)** —

Bajo nuestro throughput/potencia medidos y los supuestos de TCO declarados, SDXL es más barato en Arc a toda utilización y a todo precio de electricidad. En nuestro modelo no existe punto de operación — ninguna flota, ningún país, ningún ciclo de trabajo — en el que una 3090 genere imágenes SDXL más barato que una B70 (mucho menos que una B50). La B50 es la fábrica de imágenes más barata de las cuatro piezas.

Conclusión: el precio de lista más alto de una B70 se repaga con la factura de energía, no con marketing. En serving de LLM se recupera por encima de ∼29 % de utilización (EE.UU.) — es decir, en cualquier flota real — y en SDXL es más barata que una 3090 en todo punto de operación de nuestro modelo de TCO.

5.2 Energía y carbono a throughput fijo

Invirtamos la pregunta: mantengamos la *salida* constante y preguntemos cuánto cuesta en potencia y carbono. Este es el número que le importa a un operador de centro de datos o a un inversionista con enfoque ESG.

Throughput fijo	**RTX 3090**	**Arc B70**	**Arc B50**
1 M img SDXL / día	900 kWh/día, 328.5 MWh/año	**450 kWh/día, 164.4 MWh/año (−50 %, −66 t CO_2/año)**	**359 kWh/día (−60 %)**
1 B tokens / día (LLM)	161 kWh/día	**111 kWh/día (−31 %)**	OOM (16 GB)

Tabla 15: Energía y carbono a salida fija. Arc recorta la factura de energía entre 31 % (LLM) y 60 % (SDXL).

El número de difusión es el dramático: **un segmento con predominio de difusión de una flota de IA reduce a la mitad su electricidad y elimina ∼66 t CO_2/año por cada millón de imágenes diarias de capacidad** simplemente por ser de clase Arc en lugar de clase 3090. Conecte esto con la visión de escalado complementaria de ColabHive de **6.1 TWh / 10 millones de GPU**: un segmento con predominio de difusión de esa flota se reduce a la mitad con eficiencia de clase Arc. Como *techo ilustrativo* — no un pronóstico — si la totalidad de los 6.1 TWh fuera trabajo equivalente a SDXL corriendo en 3090s, el mismo trabajo en B70s consumiría **3.05 TWh/año**, un ahorro de **−3.05 TWh/año ≈ la producción continua de una central eléctrica de ∼350 MW**.

Conclusión: a cualquier salida fija, Arc recorta la factura de energía entre 31 % (LLM) y 60 % (SDXL), y el carbono con ella — y a escala de flota, los ahorros en difusión por sí solos se miden en centrales eléctricas, no en porcentajes.

5.3 Densidad de rack / chasis

La potencia, no los slots, es la restricción determinante en un rack moderno. Dimensionar por **potencia total de placa (TBP — 3090 350 W / B70 230 W / B50 70 W)** dentro de un **rack fijo de 10 kW** muestra lo que realmente se puede desplegar por kilowatt:

Por watt, la B50 empaca **3.3× la VRAM y 2.2× el throughput de SDXL** de una 3090; la B70 empaca **2× la VRAM y 1.5× el throughput de LLM**. Dicho al revés — **dimensionar en el**

Por rack de 10 kW	RTX 3090	Arc B70	Arc B50
Tarjetas	28	**43**	**142**
VRAM agregada (GB)	672	**1,376 (2.05×)**	**2,272 (3.38×)**
Throughput SDXL (img/min)	204	**353 (1.73×)**	**454 (2.23×)**
Throughput LLM (tok/s)	16,296	**23,878 (1.47×)**	OOM (16 GB)
VRAM por kW (GB/kW)	68.6	139.1	**228.6**

Tabla 16: Densidad de rack / chasis dentro de un presupuesto de potencia fijo de 10 kW, dimensionado por potencia total de placa.

nivel de valor para igualar el throughput SDXL de una 3090 usada requiere solo **0.89 de una B70 ($845, 205 W)** o **2.28 B50 ($796, 160 W, 36.5 GB)**. La ruta de la B50 iguala el throughput de imágenes de la 3090 por **menos capital, la mitad de la potencia y más VRAM** — y en una tarjeta **sin ningún conector de alimentación externa**.

Conclusión: la B50 de 70 W sin alimentación externa es la jugada de densidad — en un rack limitado por potencia entrega 3.3× la VRAM y 2.2× el throughput de SDXL por kilowatt de una 3090, e iguala la salida de imágenes de una 3090 usada por menos dinero y la mitad de la potencia.

Nota conservadora. Las tres subsecciones escalan las tarjetas *linealmente* e ignoran el sobrecosto de ∼10–20 % de PSU/enfriamiento que un rack real paga por encima del TBP de las tarjetas. Como ese sobrecosto escala con el wattaje de la tarjeta, penaliza *más* a la flota de 3090 de mayor wattaje que a la flota Arc de menor wattaje — así que cada cifra de densidad y energía aquí **subestima** la ventaja de Arc en lugar de inflarla.

6 Evaluación honesta de la madurez del ecosistema

El caso del hardware es sólido; la honestidad intelectual sobre el stack de software es lo que lo hace creíble.

Dónde Intel XPU está rezagado hoy:

- **QLoRA de 4 bits** inicialmente pareció fallar, pero la causa fue un **mismatch del selector de dispositivo SYCL** en el op custom de 4 bits de bitsandbytes bajo `ONEAPI_DEVICE_SELECTOR=level_zero:0`, *no* un kernel Triton-XPU faltante; con `ONEAPI_DEVICE_SELECTOR=*:gpu` entrena de extremo a extremo y supera a la 3090 (§4.7). Tanto **LoRA-fp16 como QLoRA de 4 bits** son hoy rutas de fine-tuning funcionales en Intel (§4.4, §4.7). El punto pendiente es cablear el selector en la ruta de lanzamiento de entrenamiento Intel de la plataforma.
- **Sin FlashAttention / sin xformers** en XPU (solo CUDA en upstream); la atención recurre a las rutas `torch_sdpa` / oneDNN.
- **Overhead de operadores pequeños** (TabNet, entrenamiento raw-eager): la XPU se subsatura y el costo por kernel/eager domina (§4.6).

- **El gradient boosting clásico (XGBoost, CatBoost) no tiene backend para GPU Intel** — estos permanecen en CPU sobre Intel (solo CUDA / SYCL experimental). No es una deficiencia de la B70; es una realidad de las bibliotecas.
- vLLM-XPU tiene brechas de cuantización (sin kernel INT8 scaled-mm en XPU, §4.10) y sin equivalente de CUDA-graph; Triton-XPU sigue fuera del árbol (out-of-tree).
- **El tensor-parallel multi-tarjeta (TP>1) no está validado en Arc**, y observamos una **falla real de TP en dual-B70 en el vLLM-XPU actual** (TP>1 no corre limpio a través de dos tarjetas Arc en nuestro entorno). Esta es la brecha individual más importante para la historia de 70 B / Battlematrix en el nodo (§2.3) — la eficiencia de una sola tarjeta está probada, la composición multi-tarjeta no.

La trayectoria es rápida y favorable:

- `torch.xpu` **nativo** está en PyTorch desde 2.5 (Arc A-Series + Data Center Max nombrados primero), con **el soporte de Arc B-Series (Battlemage) madurando a lo largo de 2.6–2.7**; la pieza que evaluamos corre `torch 2.10.0+xpu`.
- **bitsandbytes 0.48** agregó soporte oficial para Arc B-Series (la ruta de 4 bits es la pieza inmadura, no bnb-sobre-Arc per se).
- **IPEX se está integrando al PyTorch upstream** (el paquete standalone alcanza su EOL ∼marzo de 2026) — es decir, la era de "se necesita una extensión especial" está terminando.
- **Triton-XPU está en proceso de integración upstream** (relevante para otros kernels de cuantización, aunque — según §4.7 — *no* es la dependencia bloqueante para la ruta QLoRA de 4 bits de bitsandbytes, que ya funciona en Battlemage con el selector de dispositivo SYCL correcto).

Un hallazgo concreto de integración surgido de este trabajo: una imagen de entrenamiento construida ingenuamente sobre `ubuntu:24.04` + un compute-runtime instalado a mano **producía un segfault** en la enumeración de dispositivos de `torch.xpu`, a pesar de tener una versión correcta de torch. La solución fue construir la imagen de entrenamiento sobre la **base runtime** `llm-scaler-vllm` **de la propia Intel** (el mismo stack coherente de oneAPI/runtime usado para inferencia). La lección — *usar el runtime curado de Intel, no ensamblar uno propio* — es en sí misma un dato útil para cualquiera que ponga en marcha entrenamiento en Arc, y confirmó la hipótesis del operador de que "la clave está en el llm-scaler".

7 Conclusión y hoja de ruta

Medida en hardware de producción con energía atribuida por dispositivo, **Intel Arc Pro (Battlemage) es una alternativa práctica y eficiente en potencia y costo frente a la RTX 3090 para las cargas de trabajo de IA moderna del nivel asequible** — serving de LLM, fine-tuning LoRA/QLoRA y Stable Diffusion — entregando **95–112 % del throughput a 56–64 % de la potencia (∼1.45–2.0× perf/watt)**, a **∼0.4× el $/GB de VRAM frente a una 3090 nueva (∼0.6–0.8× frente a una usada)**. Es genuinamente *mejor* en difusión y en QLoRA, ∼paridad en LoRA y (ligeramente detrás, 95–97 %) en serving de LLM, y decisivamente mejor en rendimiento por watt en todos los frentes. Es honestamente más débil en cargas de trabajo de operadores pequeños

(TabNet — una derrota que, según la nota de §4.7, está *sin auditar*) y tiene un conjunto definido y en vías de cierre de brechas del ecosistema de software (FlashAttention, ML clásico en GPU). Este es un **caso de segunda fuente en el nivel de valor** — no un reclamo contra el silicio de frontera H100/B200.

Patrones de despliegue recomendados (hoy):

- **Serving de LLM e inferencia SDXL en Arc Pro** — el mejor perf/watt y la mejor economía; desplegar primero.
- **Fine-tuning LoRA-fp16 y QLoRA de 4 bits en Arc Pro** — ambos más rápidos y más eficientes energéticamente que la 3090; las rutas de personalización de LLM en producción (QLoRA necesita el selector `*:gpu` cableado en el lanzador de entrenamiento — una tarea de integración pequeña).
- **Mantener deep-tabular/TabNet en NVIDIA** por ahora; el GBDT clásico permanece en CPU en cualquiera de las dos plataformas.

Ruta de expansión (conforme a la propuesta de Intel de ampliar la colaboración): escalar la capacidad Arc Pro para el nivel de inferencia + LoRA, agregar **Xeon** para el nivel CPU/clásico (donde de todos modos es neutral respecto del proveedor), y evaluar **Crescent Island** para la flota de inferencia de próxima generación a medida que el stack de software XPU cierra sus brechas restantes. La combinación sirve directamente a la misión original: **cómputo de IA asequible, escalable y eficiente en potencia para la comunidad de desarrolladores e investigadores de LATAM.**

8 Apéndices — modelo TCO, datos crudos, reproducción y reporte upstream, metodología

Apéndice A — El modelo TCO (fórmula, supuestos, sensibilidad)

Todo lo de §5.1–5.3 se calcula a partir de este único modelo; un revisor puede recomputar cada celda desde él.

Costo total (all-in) por unidad de trabajo. Para una tarjeta que produce un throughput T unidades/hora a una potencia promedio P watts:

```
all_in_$/unit(U) = capital / (T x hours_life x U)        <- hardware term
                 + (P / 1000 / T) x price_kWh            <- energy term

where:
  units_over_life     = T x hours_life x U
  energy_per_unit_kWh = (P watts / 1000) / (T units/hour)
```

Supuestos (idénticos al recuadro de §5): `hours_life = 26,280 h` (depreciación lineal a 3 años). `price_kWh` ∈ {US 0.15, AR 0.10, DE 0.30}. Carbono de la red `= 0.4 kg CO2/kWh`. Capital: `3090-used $875`, `3090-new $1,499`, `B70 $949`, `B50 $349`. T y P son las celdas medidas de §4 (serving de LLM C=16: 3090 582 tok/s @ 343 W, B70 555.3 tok/s @ 219 W; SDXL: 3090 7.3 img/min @ 394 W, B70 8.2 img/min @ 221 W, B50 3.2 img/min @ 69 W).

Utilización de cruce. Igualando `all_in_B70(U) = all_in_3090used(U)` y despejando U en la celda LLM (donde la B70 gana en energía pero no en hardware) se obtiene la forma cerrada usada en §5.1:

```
U* = dHW_constant / price_kWh = 0.0429 / price_kWh
  -> US 0.15 -> 28.6 %   |   DE 0.30 -> 14.3 %   |   AR 0.10 -> 43 %
```

Para SDXL la B70 gana en **ambos** términos, por lo que la ecuación no tiene cruce con U positivo → victoria incondicional (§5.1).

Sensibilidad (las dos perillas que mueven la respuesta):

- **$/kWh.** El término de energía — y la *totalidad* de la ventaja de la B70 en serving de LLM — escala linealmente con el precio de la electricidad. La energía más barata (Argentina) empuja el cruce de LLM hasta 43 % de utilización y achica el margen; la energía cara (Alemania) lo baja a 14.3 % y lo amplía. SDXL es insensible en el signo (siempre gana) y solo varía en *magnitud*.
- **Utilización *U*.** U solo escala el término de hardware, nunca el término de energía. Por eso cada resultado aquí es **más favorable a la tarjeta de energía más barata (Arc) a alta utilización** y más favorable a la tarjeta de menor precio de lista (3090 usada) a baja utilización. Una flota de producción vive a U alto, que es el régimen donde Arc gana.

Apéndice B — Tabla consolidada de datos crudos (cada celda medida)

Para que un revisor pueda recomputar J/unidad, perf/watt y la economía de §5 de forma independiente. Todas las celdas fueron medidas el 2026-06-21, bf16/eager, GPU única dedicada. "—" = no ejecutado / no aplica; "OOM" = no entró en la VRAM.

Método de energía por celda: Intel = contador exacto sysfs de Xe (todas las celdas); NVIDIA = contador exacto NVML para QLoRA, integral de `nvidia-smi power.draw` @200 ms para LLM/SDXL (asimetría señalada en el Apéndice D).

Apéndice C — Reproducción y reporte upstream

C.1 Reproducción de las celdas principales. Fijar una GPU única dedicada por lado, drenar de ella los demás modelos, bf16/eager en todo, prompts/entradas/longitudes idénticos. El throughput de LLM sale del delta de `vllm:generation_tokens_total` de vLLM sobre una ventana delimitada por `MEASURE_START/END` (`ignore_eos` para fijar la longitud de salida). La energía, de los contadores acumulativos exactos donde fue posible (Xe `energy1_input`, NVML `nvmlDeviceGetTotalEnergyConsumption`). Versiones de software en el Apéndice D. La celda QLoRA requiere la corrección del selector descrito abajo.

C.2 Reporte de bug listo para presentar — QLoRA de 4 bits con bitsandbytes en Intel Arc B-series (trampa del selector de dispositivo).

Título: QLoRA de 4 bits falla en Intel Arc B-series bajo `ONEAPI_DEVICE_SELECTOR=level_zero:N` — SYCL `No device of requested type available` (el op de 4 bits de bnb rechaza

el selector Level-Zero que la propia documentación multi-GPU de Intel recomienda)
Componentes: `bitsandbytes` (op custom de 4 bits en XPU) · documentación multi-GPU de Intel IPEX / `llm-scaler`
Entorno: Arc Pro B70 (`0xe223`, BMG-G31, 32 GB) y Arc Pro B50 (`0xe212`, 16 GB); `torch 2.10.0+xpu`; `bitsandbytes 0.49.2`; compute-runtime 26.05+; driver `xe`; Ubuntu 24.04 / kernel 6.17; imagen construida sobre la base `intel/llm-scaler-vllm:0.14.0-b8.3.1`.
Firma de la falla: Con `ONEAPI_DEVICE_SELECTOR=level_zero:0` (según la guía de pinning multi-GPU de Intel), cargar un modelo cuantizado en NF4 e iniciar un paso de QLoRA hace que el op nativo de 4 bits de bitsandbytes (`torch.ops.bitsandbytes.quantize_4bit`) lance un SYCL `No device of requested type available`. Nótese que `torch.xpu.is_available()` es `True` y `torch.xpu.get_device_properties()` enumera la GPU correctamente — solo la cola del kernel SYCL compilado por separado de bitsandbytes rechaza el dispositivo.
Causa raíz: la cola SYCL de 4 bits de bitsandbytes no resuelve un dispositivo cuando el proceso está acotado al selector de backend **Level-Zero**. `torch.xpu` y el runtime SYCL de bnb resuelven dispositivos por rutas distintas; el acotamiento solo-Level-Zero que satisface a torch deja sin dispositivo a la cola de bnb.
Fix (una línea de configuración): Usar el selector de GPU de cualquier backend y fijar la tarjeta específica por máscara de afinidad en lugar de por índice Level-Zero:

```
# FAILS for bnb 4-bit:  ONEAPI_DEVICE_SELECTOR=level_zero:1
# WORKS:
export ONEAPI_DEVICE_SELECTOR='*:gpu'
export ZE_AFFINITY_MASK=1        # pin to the desired card (0-based)
```

Con esto, QLoRA de 4 bits entrena de extremo a extremo (verificado como genuinamente de 4 bits: NF4 7B = 5.45 GiB).
Alcance afectado: la **serie Arc B completa** (reproducido en **la B70 y la B50** con la misma corrección). Críticamente, afecta a **Project Battlematrix (8×B60)** y a cualquier despliegue Arc multi-tarjeta, porque allí el pinning por tarjeta es obligatorio y la guía estándar `level_zero:N` es exactamente lo que dispara la falla — de modo que seguir la documentación multi-GPU estándar es exactamente lo que hace aflorar esta falla.
Cambio de documentación solicitado (Intel): en la guía de pinning multi-GPU de IPEX / llm-scaler, indicar que las cargas de trabajo que usan bitsandbytes de 4 bits deben fijarse con `ONEAPI_DEVICE_SELECTOR='*:gpu'` + `ZE_AFFINITY_MASK=<idx>`, no con `level_zero:<idx>`.
Fix solicitado (bitsandbytes): hacer que la cola del kernel SYCL de 4 bits resuelva un dispositivo bajo el selector de backend Level-Zero (o emitir un error accionable que apunte al selector, en lugar del opaco `No device of requested type available`).

Apéndice D — Metodología, versiones, reproducibilidad (procedencia)

Versiones de software (tal como se evaluaron):

- Inferencia Intel: `inference-ipex:v0.7.24` ← `intel/llm-scaler-vllm:0.14.0-b8.3.1` (`torch 2.10.0+xpu`, vLLM-XPU, compute-runtime 26.09).
- Entrenamiento Intel: `training-ipex:v0.2.0` (base llm-scaler + `peft`, `trl`, `bitsandbytes 0.49.2`, `pytorch-tabnet 4.1.0`; PIP_CONSTRAINT fija `torch==2.10.0+xpu`).

- NVIDIA: `inference-vllm:v2.0.8`, `inference-generative:v3.0.0`, `training-transformers-cu121:v1.0.5`, `training-pytorch-cu121:v1.0.1` (la línea base de TabNet requirió en runtime `numpy<2` + `pytorch-tabnet==4.1.0` debido a una incompatibilidad numpy-2/torch-2.1.2 en la imagen más antigua).
- Host: Ubuntu 24.04, kernel 6.17, driver `xe`, compute-runtime 26.05+, Resizable BAR activado.

Método de energía (y su asimetría, declarada sin rodeos): Intel = sysfs `energy1_input` de `xe` (μJ, contador acumulativo exacto, por tarjeta según id PCI). NVIDIA = el contador exacto NVML `nvmlDeviceGetTotalEnergyConsumption` donde se usó (QLoRA); en el resto, la integral de `nvidia-smi power.draw` @200 ms (LLM, SDXL). **Los dos métodos de NVIDIA no son el mismo instrumento** — el muestreo a 200 ms puede perder transitorios sub-muestra — por lo que los márgenes de perf/watt de LLM/SDXL cargan una pequeña incertidumbre extra que la celda QLoRA (contador exacto, ambos lados) no tiene. Ventana delimitada por `MEASURE_START/END` emitidos por la carga de trabajo. Ahora se recolecta a nivel de toda la flota mediante node-runtime 0.10.119 → `node_power_samples`.

Cargas de trabajo: Qwen2.5-7B-Instruct (LLM, vLLM y LoRA/QLoRA); Stable Diffusion XL base 1.0 (inferencia + entrenamiento UNet-LoRA); tabular sintético (TabNet). bf16/eager en todo; prompts/entradas/longitudes idénticos entre fabricantes.

Procedencia de los datos / puntos abiertos:

- **K=3 (±1 desviación estándar muestral, n=3)** cubre las dos celdas de inferencia — serving de LLM (§4.2) y SDXL (§4.3). LoRA (§4.4) y QLoRA (§4.7) son corridas medidas apareadas reportadas como **estimaciones puntuales** (repetidas, sin afirmación de IC ajustado). SDXL-UNet-LoRA (§4.5) y TabNet (§4.6) son corridas únicas. **Salvedad de muestreo:** todo "K=3" es repetición *en el mismo dispositivo* — **N=1 de hardware por proveedor** — así que la variación por lotería de silicio / board-partner / térmica no está controlada; un segundo par físico y un segundo tamaño de modelo (∼1.5 B) son los próximos pasos. El caso de margen de VRAM de 14 B (§4.8) está medido.
- **Confusores conocidos (aún no eliminados):** (a) *desfase de versión de motor* — Intel corre vLLM-XPU 0.14.x sobre `torch 2.10+xpu`; NVIDIA corre vLLM stock (`inference-vllm:v2.0.8`, linaje CUDA cu121). bf16/eager está fijado para fast-paths justos, pero son forks/versiones distintos de vLLM, así que la cuasi-paridad refleja silicio *más* motor, no silicio solo. (b) *desajuste de CPU del host* — host Intel i7-10700 (8c) vs NVIDIA Xeon E5-2680 v4 (28c); irrelevante para kernels residentes en GPU, pero un confusor vigente específicamente para la celda **TabNet**, ligada a sincronización con el host (su pérdida de 2.2× puede deberse en parte a la CPU del host, y está sin auditar).
- **QLoRA de 4 bits ahora corre en la B70** (§4.7) — 0.949 pasos/s · 242.3 J/paso, ∼7 % más rápido y 1.89× más eficiente que la 3090, verificado como genuinamente de 4 bits (5.45 GiB en ambos fabricantes). El resultado anterior de "falla" fue un **mismatch del selector de dispositivo SYCL** (`level_zero:0` → el op de 4 bits de bnb lanza "No device"; `*:gpu` lo arregla), **no** una brecha de Triton-XPU como se hipotetizó al principio.
- Intel **no publica TFLOPS BF16 densos** para Arc Pro B-series; la tabla de economía de cómputo usa únicamente los TOPS INT8 publicados.
- Se identificó una mejora de plataforma aparte (aún no desplegada): el runtime del nodo lee la VRAM de Intel vía `xpu-smi` (que reporta "No device" en la B70) y recurre a contabilidad; leer

`torch.xpu.mem_get_info()` daría una lectura viva y precisa de la VRAM Intel y mejoraría el placement.

- Las cifras de participación de mercado, precios, CUDA-EULA y especificaciones en §2 y §5 provienen de fuentes públicas (TechInsights/HPCwire, NVIDIA CUDA EULA, datasheets/newsroom de Intel, páginas de especificaciones de los proveedores); los precios de calle son volátiles y están señalados como tales.

Apéndice E — Fuentes (públicas)

Las cifras externas de mercado, precios, licenciamiento y especificaciones de hardware en §2 y §5 provienen de fuentes públicas; los precios de calle y las especificaciones en distribución son volátiles y estaban vigentes a la fecha de medición 2026-06-21.

1. Participación de mercado de NVIDIA en GPU de centro de datos (∼98%, 3.76M de 3.85M unidades, 2023) — TechInsights, reportado vía HPCwire: https://www.hpcwire.com/2024/06/10/nvidia-shipped-3-76-million-data-center-gpus-in-2023-according-to-study/ (ver también Tom's Hardware: https://www.tomshardware.com/tech-industry/nvidia-shipped-376m-data-center-gpus-in-2023-dominates-business-with-98-revenue-share).
2. NVIDIA CUDA End User License Agreement, §1.2 "Limitations" (ítem 8, sobre traducir salida de CUDA a plataformas no NVIDIA) — NVIDIA: https://docs.nvidia.com/cuda/eula/index.html.
3. Baja (takedown) del proyecto ZLUDA (CUDA sobre no-NVIDIA) a pedido de AMD, agosto de 2024 — repositorio del proyecto (https://github.com/vosen/ZLUDA) y cobertura de prensa (Phoronix: https://www.phoronix.com/news/AMD-ZLUDA-CUDA-Taken-Down).
4. Suministro de Blackwell ("sold out ∼12 months ahead," oct 2024) — Tom's Hardware: https://www.tomshardware.com/pc-components/gpus/nvidias-blackwell-gpus-are-sold-out-for-the-next-12-months-chipmaker-to-gain-market-share- precio por GPU ($30,000–40,000, Jensen Huang) — CNBC: https://www.cnbc.com/2024/03/19/nvidias-blackwell-ai-chip-will-cost-more-than-30000-ceo-says.html.
5. Intel Arc Pro B-series (Battlemage) y Project Battlematrix (8×B60 → 192 GB) — Intel Newsroom, Computex 2025: https://newsroom.intel.com/client-computing/computex-intel-unveils-new-gpus-ai-workstations; stack de software llm-scaler: https://github.com/intel/llm-scaler.
6. Soporte de 4 bits Arc / XPU en bitsandbytes — notas de la versión bitsandbytes 0.48.0: https://github.com/bitsandbytes-foundation/bitsandbytes/releases/tag/0.48.0.
7. Línea de tiempo del soporte nativo torch.xpu (PyTorch 2.5 en adelante; Battlemage madurando en 2.6–2.7) — PyTorch: https://pytorch.org/blog/intel-gpu-support-pytorch-2-5/; EOL del IPEX standalone (∼marzo de 2026) — Intel: https://github.com/intel/intel-extension-for-pytorch/issues/867.
8. Especificaciones de GPU (VRAM, ancho de banda de memoria, TBP, TOPS INT8) — NVIDIA RTX 3090: https://www.nvidia.com/en-us/geforce/graphics-cards/30-series/rtx-3090-3090ti/; Intel Arc Pro B70: https://www.intel.com/content/

www/us/en/products/sku/245797/intel-arc-pro-b70-graphics/specifications.html; B60: https://www.intel.com/content/www/us/en/products/sku/243916/intel-arc-pro-b60-graphics/specifications.html; B50: https://www.intel.com/content/www/us/en/products/sku/242615/intel-arc-pro-b50-graphics/specifications.html.

9. Precios de calle de GPU workstation en LATAM (§2.1): PNY RTX 4000 Ada 20 GB a R$ 19,229.99 de lista / R$ 17,306.99 al contado, KaBuM (Brasil), consultado el 2026-07-08: https://www.kabum.com.br/produto/594205/placa-de-video-pny-nvidia-quadro-rtx-4000-ada-20gb-gddr6-160-bit-vcnrtx4000ada-pb; US$ 1,250 de MSRP de lanzamiento en EE.UU. — CG Channel: https://www.cgchannel.com/2023/08/nvidia-unveils-rtx-4000-4500-and-5000-workstation-gpus/.

10. Tributación de importación en Brasil sobre compras internacionales (60 % sobre US$50 más 17–20 % de ICMS estadual, régimen Remessa Conforme) — Receita Federal (oficial): https://www.gov.br/receitafederal/pt-br/assuntos/aduana-e-comercio-exterior/manuais/remessas-postal-e-expressa/programa-remessa-conforme-o-que-e-como-funciona.

11. Brecha de precios de electrónica en Argentina (∼2.5× EE.UU.: teléfono insignia US$ 2,566 vs US$ 1,011, mayo de 2025) — Argentina.gob.ar (oficial): https://www.argentina.gob.ar/noticias/el-gobierno-nacional-eliminara-impuestos-productos-electronicos-para-alentar-la-baja-de.

12. Disponibilidad y precios de GPU en la nube en LATAM (§2.1): Lehdonvirta, Wu & Hawkins, "Compute North vs. Compute South: The Uneven Possibilities of Compute-based AI Governance Around the Globe," AAAI/ACM AIES 2024: https://ojs.aaai.org/index.php/AIES/article/view/31683 (preprint: https://osf.io/preprints/socarxiv/8yp7z); Google Cloud "GPU regions and zones" (consultado el 2026-07-08): https://docs.cloud.google.com/compute/docs/regions-zones/gpu-regions-zones; precios on-demand de AWS EC2, sa-east-1 vs us-east-1 (consultado el 2026-07-08): https://aws.amazon.com/ec2/pricing/on-demand/.

Carga de trabajo (unidad)	Métrica	RTX 3090	Arc B70	Arc B50
Serving de LLM C=1 (tok/s)	throughput	37.8 ± 3.4	36.6 ± 0.1	—
Serving de LLM C=8 (tok/s)	throughput	296.7 ± 6.9	287.0 ± 0.5	—
Serving de LLM C=16 (tok/s)	throughput	582.0 ± 19.6	555.3 ± 2.1	OOM
Serving de LLM C=16	tok/J	1.723	2.511	—
Serving de LLM (carga)	potencia prom. (W)	∼343	∼219	—
SDXL (por imagen)	latencia (s)	8.21 ± 0.09	7.33 ± 0.03	18.74
SDXL (por imagen)	throughput (img/min)	7.3	8.2	3.2
SDXL (por imagen)	energía/imagen (J)	3,240 ± 37	1,621 ± 6	1,291
SDXL	potencia prom. (W)	∼394	∼221	∼69
LoRA-fp16 7B (paso)	thr. (st/s, tok/s)	1.90, 1,944	1.96, 2,007	OOM
LoRA-fp16 7B	energía/paso (J)	180.8	110.1	OOM
LoRA-fp16 7B	potencia prom. (W)	∼347	∼222	OOM
QLoRA-NF4 7B (paso)	thr. (st/s, tok/s)	0.889, 911	0.949, 972	0.373
QLoRA-NF4 7B	energía/paso (J)	459.0	242.3	187
QLoRA-NF4 7B	potencia prom. (W)	∼408	∼230	∼70
QLoRA-NF4 7B	VRAM 4 bits (GiB)	5.45	5.45	5.45
SDXL UNet-LoRA (paso)	throughput (st/s)	1.625	0.930	—
SDXL UNet-LoRA	energía/paso (J)	208	161	—
SDXL UNet-LoRA	potencia prom. (W)	∼346	∼152	—
TabNet 16k×64 (rows/s)	throughput	16,852	7,739	3,700
TabNet	potencia prom. (W)	∼108	∼87	∼39
TabNet	rows/J	159	89	—
Qwen2.5-14B bf16	carga de pesos	OOM (24 GB)	carga (31.1 GiB)	—
AWQ-4bit C=1 (tok/s)	throughput	40.8	48.9	—
AWQ-4bit C=8 (tok/s)	throughput	302.9	381.8	—
AWQ-4bit C=16 (tok/s)	throughput	598.0	709.5	—

Tabla 17: Datos crudos consolidados, cada celda medida.