Micro IA y el Desafío de la Escala: Análisis Arquitectónico, Operacional y Posicionamiento Competitivo del Modelo Tiny Recursive Model (TRM) de Samsung

I. Marco Conceptual de la Micro IA y Edge Computing

I.A. La Convergencia de Edge Computing e Inteligencia Artificial (Micro IA)

La Micro Inteligencia Artificial (Micro IA), también conocida como Edge AI, representa un cambio de paradigma en el procesamiento de datos, donde la computación se traslada del entorno centralizado de la nube a la proximidad inmediata de las fuentes de datos. Estas fuentes incluyen sensores, dispositivos de Internet de las Cosas (IoT) y usuarios finales.[1] Este modelo distribuido permite que los algoritmos de Inteligencia Artificial (IA) ejecuten procesos de Machine Learning de manera local en el *hardware* que genera los datos, incluso sin una conexión activa a internet.[3]

El principal impulsor de esta migración hacia el borde es la necesidad de una velocidad de procesamiento superior y la capacidad de ofrecer retroalimentación en tiempo real. Al eliminar la latencia inherente a la comunicación con servidores remotos, la Edge AI puede ofrecer respuestas en cuestión de milisegundos, un requisito crítico para aplicaciones que van desde sistemas de control robótico hasta recomendaciones dinámicas en puntos de venta.[1] Además de la velocidad, la implementación local refuerza significativamente la privacidad y seguridad de los datos. El procesamiento de información sensible en el dispositivo mitiga los riesgos asociados a la transferencia masiva de datos a entornos de nube, ayudando a garantizar el cumplimiento de normativas estrictas como el GDPR.[4] Estratégicamente, esta aproximación genera ahorros sustanciales en ancho de banda y costos operativos, ya que se evita la transferencia de volúmenes crecientes de datos multimodales.[4] La importancia de esta tendencia es subrayada por proyecciones de la industria que estiman que el porcentaje de datos empresariales procesados en el borde aumentará del 10% en 2018 a un estimado del 75% para 2025, lo que convierte a la Micro IA en una urgencia estratégica.[6]

I.B. Limitaciones Físicas y de Cómputo en Dispositivos de Borde

El despliegue de modelos de IA, especialmente los modelos de lenguaje (LLMs), en dispositivos de borde (móviles, tabletas, sensores industriales) se enfrenta a limitaciones físicas severas que no existen en los centros de datos en la nube. Estas restricciones definen el campo de juego para modelos como el *Tiny Recursive Model* (TRM).

En primer lugar, los dispositivos operan con un **presupuesto de cómputo limitado** impuesto por los System-on-Chip (SoC).[7] A diferencia de los servidores, los SoCs poseen recursos computacionales y un número de unidades de cómputo paralelas restringidos, lo que dificulta la ejecución de las billones de operaciones por segundo (TOPS) que típicamente demandan los modelos generativos. Esto exige que la carga computacional sea reducida a un nivel estructural.[7] En segundo lugar, existe una **limitación crítica en el ancho de banda de memoria I/O**. Los modelos generativos de alto rendimiento a menudo requieren manejar cientos de megabytes o incluso gigabytes de parámetros y activaciones intermedias. Sin embargo, la capacidad de DRAM y las velocidades de acceso a la memoria externa en dispositivos de borde son significativamente inferiores a las de los servidores, lo que crea un cuello de botella que degrada el rendimiento del sistema y aumenta el consumo energético.[7]

Finalmente, las limitaciones de **energía de batería y el sobrecalentamiento térmico** son determinantes. Los dispositivos móviles con batería tienen límites de potencia estrictos. El consumo excesivo de energía genera calor, lo que inevitablemente activa el estrangulamiento térmico (*thermal throttling*), reduciendo automáticamente el rendimiento sostenido. La ingeniería de modelos para el Edge, por lo tanto, debe priorizar la eficiencia energética.[7]

La necesidad de superar estas barreras físicas ha llevado a una **integración vertical hardware-software** por parte de fabricantes líderes. Si un modelo extremadamente pequeño como el TRM, con solo 7 millones de parámetros, es diseñado para ser viable en el dispositivo, requiere un entorno de ejecución (*runtime*) que aproveche al máximo el *hardware* local. Esto impulsa a compañías como Samsung a desarrollar Unidades de Procesamiento Neural (NPUs) optimizadas durante múltiples generaciones para mejorar la eficiencia y el rendimiento.[5] El desarrollo de arquitecturas de modelos de Micro IA (como el TRM) está, por lo tanto, intrínsecamente ligado a la infraestructura de *hardware* local para maximizar el desempeño en entornos propios.

II. Modelos de Lenguaje Pequeños (SLMs): Estrategias de Optimización y Contexto Competitivo

II.A. El Papel de los SLMs en la Cadena de Suministro de IA

Los Modelos de Lenguaje Pequeños (SLMs) son fundamentales para la viabilidad de la Micro IA, ya que están diseñados específicamente para operar dentro de las severas restricciones

del *edge computing*. Los SLMs ofrecen ventajas decisivas frente a sus contrapartes de gran escala (LLMs): son más rápidos, significativamente más rentables para entrenar y ejecutar, y requieren menos potencia de cómputo y datos.[8] Esto no solo facilita su despliegue en infraestructuras locales y privadas, sino que también garantiza una inferencia más rápida.[9]
No obstante, la reducción de la escala conlleva una limitación principal: la **generalización**. Los SLMs suelen tener una capacidad de generalización limitada fuera del dominio específico en el que fueron entrenados, a diferencia de la alta capacidad de los LLMs para abordar diversos temas.[9] La industria ha respondido a esta dualidad con una variedad de SLMs competitivos, incluyendo modelos de propósito general optimizados para el Edge, como Phi-3 Mini (3.8 mil millones de parámetros), Gemma 2B y Llama 3 8B, y Mistral 7B.[8]

II.B. Técnicas Fundamentales de Compresión de Modelos

Para hacer que los SLMs de propósito general sean aptos para la implementación en dispositivos, se emplean técnicas avanzadas de compresión que reducen el tamaño del modelo y la carga computacional mientras se conserva la máxima precisión posible.[13]

- **Cuantificación (*Quantization*):** Este es uno de los métodos más críticos, ya que reduce la precisión de los pesos del modelo, por ejemplo, de la precisión de punto flotante estándar (FP32) a formatos de 8 bits (Q8) o 4 bits (Q4). Esto aligera la carga de cómputo y acelera la inferencia.[13] La cuantificación se puede integrar durante el entrenamiento (QAT, *Quantization-Aware Training*) para mayor precisión, o realizarse después del entrenamiento (PTQ, *Post-Training Quantization*).[13]
- **Destilación de Conocimiento (*Knowledge Distillation*):** Esta técnica consiste en transferir el conocimiento y la capacidad de rendimiento de un modelo grande y complejo (el "maestro") a un modelo más pequeño y ágil (el "estudiante").[13]
- **Poda (*Pruning*) y Factorización de Bajo Rango (*Low-Rank Factorization*):** La poda elimina conexiones o neuronas redundantes, mientras que la factorización de bajo rango simplifica operaciones matriciales complejas mediante una aproximación más compacta.[13]

II.C. Aceleración de Hardware Dedicado: La NPU de Samsung

La capacidad de ejecutar SLMs de manera eficiente en el Edge depende intrínsecamente del *hardware* especializado. La **Unidad de Procesamiento Neural (NPU)** es un circuito diseñado específicamente para implementar la lógica y la aritmética necesarias para ejecutar algoritmos de aprendizaje automático de forma rápida y con bajo consumo de energía.[16]
Samsung ha invertido en el desarrollo continuo de sus NPUs, mejorando su rendimiento con cada generación.[5] Algunos de sus dispositivos ofrecen NPUs capaces de realizar hasta 4.8 billones de operaciones por segundo, asegurando el funcionamiento fluido de las funciones de IA.[18] Para apoyar a los desarrolladores, el Samsung Neural SDK proporciona una plataforma para la ejecución eficiente de redes neuronales preentrenadas. Este SDK permite

a los desarrolladores optimizar sus modelos (que admiten formatos populares como Caffe, TensorFlow y ONNX) y seleccionar el motor de cómputo más apropiado (CPU, GPU, DSP o NPU) según los requerimientos de la aplicación, garantizando el uso óptimo de recursos como la memoria y la potencia.[19]

Este panorama revela que la Micro IA se está polarizando en dos estrategias: los SLMs generalistas (1B-8B) que utilizan compresión estándar para tareas de conocimiento (evaluadas mediante *benchmarks* como MMLU) y los **Modelos Tiny Radicalmente Especializados (TSMs)**, como el TRM, que redefinen la arquitectura para lograr una eficiencia de parámetros extrema en tareas de razonamiento puro. Si bien los SLMs típicos de 1B a 8B son un avance, un modelo incluso más pequeño como el DeepSeek-R1-Distill-Qwen-1.5B sigue pesando 3.5 GB.[14] Para el razonamiento avanzado (como el exigido por ARC-AGI), el conocimiento general masivo se vuelve secundario.[20] Por lo tanto, el TRM capitaliza la necesidad de una huella de memoria mínima (7M de parámetros) al enfocarse exclusivamente en el *proceso* de razonamiento, logrando una implementación en el borde con una eficiencia inédita.

III. El Samsung Tiny Recursive Model (TRM): Una Ruptura en el Paradigma de la Escala

III.A. Origen y Desafío al Paradigma Dominante

Durante años, la industria de la IA ha estado dominada por el aforismo de que "cuanto más grande, mejor" en lo que respecta al rendimiento de los modelos de lenguaje.[21] Esto ha resultado en modelos fundacionales que consumen miles de millones de dólares en entrenamiento y poseen trillones de parámetros. El *Tiny Recursive Model* (TRM), desarrollado por el laboratorio Samsung SAIL Montréal, desafía radicalmente esta suposición.

El TRM demuestra que el rendimiento superior en tareas de razonamiento no siempre depende de la escala.[22] Con un total de solo **7 millones de parámetros** (0.007B) [23], el TRM es, en algunos casos, hasta 10,000 veces más pequeño que los LLMs prominentes.[21] Este tamaño minúsculo permite que el modelo se ejecute fácilmente en *hardware* de consumo limitado, como un ordenador portátil.[21]

III.B. Arquitectura Simplificada y Eficiencia Paramétrica

El TRM deriva y simplifica el enfoque de razonamiento recursivo introducido por el *Hierarchical Reasoning Model* (HRM), que utilizaba 27 millones de parámetros.[23] En contraste, la arquitectura del TRM es excepcionalmente minimalista, utilizando una única red neuronal diminuta con solo **2 capas**.[23]

La capacidad del TRM de lograr un alto rendimiento con una arquitectura tan poco profunda

se manifiesta en sus resultados en *benchmarks* de razonamiento complejo. El modelo está diseñado para sobresalir en tareas de lógica y patrones abstractos, como el *benchmark* ARC-AGI (Abstract Reasoning Corpus), una prueba desarrollada para evaluar la capacidad de razonamiento real.[21]

Los resultados de *state-of-the-art* (SOTA) del TRM en razonamiento son notables, superando a modelos que lo superan en escala por varios órdenes de magnitud:

- **ARC-AGI-1:** 44.6% de precisión (una mejora significativa respecto al 40% del HRM).[21]
- **ARC-AGI-2:** 7.8% de precisión (un *benchmark* donde la mayoría de los LLMs masivos obtienen menos del 5%).[21]
- **Sudoku-Extreme:** 87% de precisión (frente al 55% de los métodos anteriores).[21]

Esta eficiencia no se logra mediante una mayor cantidad de parámetros, sino a través de la **recurrencia**. La arquitectura del TRM compensa la falta de profundidad estructural (solo 2 capas) mediante un proceso de **intercambio de "parámetros por pases"**.[25] En lugar de acumular miles de millones de parámetros para ganar profundidad, el modelo realiza múltiples pasadas de inferencia recursivas para refinar su resultado. Esta aproximación logra una alta *profundidad efectiva* a través del tiempo y la iteración, optimizando el uso de la NPU local en ciclos repetitivos de cómputo en el dispositivo.[25]

IV. Mecanismos Operacionales del TRM: El Razonamiento Recursivo

IV.A. La Metodología del Razonamiento Iterativo

El mecanismo operativo central del TRM es su enfoque en el **razonamiento recursivo**. A diferencia de los modelos de lenguaje tradicionales que generan texto de forma auto-regresiva (token por token, donde un error inicial puede invalidar toda la respuesta [22]), el TRM opera mediante un proceso de refinamiento continuo. El concepto es simple: el modelo propone una respuesta, la evalúa internamente (implícitamente) y luego se retroalimenta para mejorarla a través de múltiples iteraciones.[21]

Este enfoque se diferencia de la popular técnica *Chain-of-Thought* (CoT) utilizada por LLMs masivos. CoT requiere que el modelo genere pasos de razonamiento explícitos, lo cual es costoso computacionalmente, exige datos de razonamiento de alta calidad y puede ser frágil si la lógica generada es incorrecta.[22] El TRM, en cambio, realiza este razonamiento de forma **latente e interna** mediante la recursión de su propia red, resultando en un método más elegante y eficiente para procesar la lógica.[25]

IV.B. El Modelo de Variables Centrales y el Bucle de Refinamiento

El TRM gestiona su razonamiento a través de un conjunto de variables de estado incrustadas

(embedded states) que se actualizan recursivamente. Estas variables son [24]:

1. **x (Entrada/Pregunta):** La representación incrustada de la pregunta o tarea inicial.
2. **y (Respuesta Actual):** La respuesta propuesta por el modelo en la iteración actual.
3. **z (Estado Latente):** Una variable clave que funciona como la **memoria de razonamiento** o "espacio de trabajo" interno del modelo.

El proceso de inferencia se ejecuta hasta K pasos de mejora, buscando optimizar la respuesta y.[24]

IV.C. El Ciclo de Doble Actualización (Mejora del Razonamiento y de la Respuesta)

El núcleo operacional del TRM consiste en un ciclo de doble actualización que permite al modelo profundizar en la lógica sin aumentar el número de parámetros:

1. **Paso I: Actualización Recursiva del Estado Latente (z):**
    - El modelo entra en un bucle interno donde actualiza recursivamente el estado latente z un número n de veces.[24] Esta actualización se basa en la combinación de la pregunta (x), la respuesta actual (y), y el estado latente previo (z).[24]
    - Este ciclo, que puede utilizar hasta $N_{sup}=16$ pasos de supervisión [29], es crucial, ya que permite al modelo **razonar sobre el problema** y refinar sus características latentes.[23] El estado latente z se reutiliza como la inicialización para el siguiente pase de procesamiento, permitiendo la iteración del proceso de pensamiento y emulando una profundidad de cómputo mucho mayor que la de sus 2 capas físicas.[24]
2. **Paso II: Actualización de la Respuesta (y):**
    - Una vez que el estado latente z ha sido refinado mediante la recursión interna (la fase de "planificación" o lógica), el modelo utiliza este estado mejorado para actualizar la respuesta predicha y.[24]

Este mecanismo de razonamiento recursivo ofrece una ventaja arquitectónica fundamental: la **separación entre la lógica y la generación de la respuesta**. Al refinar su lógica internamente dentro del espacio latente z antes de generar la salida y, el TRM evita la fragilidad inherente de los LLMs auto-regresivos, donde un error temprano en la secuencia de tokens puede arruinar la solución final.[23] Esta capacidad de gestionar la inteligencia de forma interna y generar solo el resultado final es la clave de su eficiencia extrema en tareas lógicas.

V. Análisis de Rendimiento y Posicionamiento Competitivo

V.A. Evaluación del Rendimiento del TRM en Benchmarks de

Razonamiento

Para comprender la eficiencia del TRM, es esencial evaluar su rendimiento en *benchmarks* que midan la capacidad de razonamiento y no solo el conocimiento general o la memorización. Mientras que métricas como el MMLU (*Massive Multitask Language Understanding*) evalúan el conocimiento interdisciplinario [30], tareas como ARC-AGI y Sudoku-Extreme se centran en la solución de problemas abstractos y la lógica, donde la memorización es inútil.[21]

Los resultados del TRM demuestran que, al priorizar una arquitectura eficiente para el razonamiento recursivo, se puede superar a modelos con miles de millones de parámetros. El TRM logra un 44.6% en ARC-AGI-1 y un 7.8% en ARC-AGI-2, superando a grandes competidores en esta métrica específica, a pesar de su minúsculo tamaño.[21]

V.B. Tabla Comparativa 1: TRM vs. Modelos de Lenguaje Pequeños (SLMs) de la Categoría Edge AI

La siguiente tabla compara el TRM, un modelo de razonamiento especializado, con los principales SLMs de propósito general optimizados para el Edge. La comparación destaca el *trade-off* entre especialización extrema (TRM) y generalización (SLMs típicos de 1B a 8B).

Tabla 1: Comparación del TRM con Modelos de Lenguaje Pequeños (SLMs) de la Categoría Edge AI

Modelo	Desarrollador	Parámetros	Foco Principal	MMLU (5-shot, Est.)	ARC-AGI-1 (Accuracy)
Tiny Recursive Model (TRM)	Samsung SAIL Montréal	7 Millones (0.007B) [23]	Razonamiento Lógico Recursivo	N/A (Bajo) [20]	~45% (SOTA) [21]
Gemma 3 1B	Google DeepMind	1B [33]	Propósito General, Multilingüe	38.8% - 49.3% [33]	N/A
Phi-3 Mini (Instruct)	Microsoft	3.8B [8]	Propósito General, Conocimiento	~69.0% [31]	N/A
Mistral 7B	Mistral AI	7B [34]	Propósito General, On-device AI	Alto (Generalista) [34]	N/A
Llama 3.1 8B	Meta	8B [11]	Propósito General, Rendimiento	Alto (Generalista) [11]	N/A

El análisis de la Tabla 1 ilustra una bifurcación estratégica en la Micro IA. Los SLMs más grandes (1B a 8B) compiten en métricas de conocimiento (MMLU) y capacidades de propósito

general, utilizando técnicas de compresión para caber en el Edge. Sin embargo, el TRM renuncia casi por completo a la huella de conocimiento a cambio de una arquitectura optimizada para la lógica (ARC-AGI), lo que resulta en una eficiencia paramétrica cientos de veces mayor para esa tarea específica.

V.C. Tabla Comparativa 2: Contraste de TRM con Modelos Fundacionales de Gran Escala (LLMs)

El contraste con los LLMs fundacionales resalta la desproporción de escala y la ineficiencia de los modelos generalistas cuando se enfrentan a tareas de razonamiento puramente lógico.
Tabla 2: Contraste de Escala y Rendimiento de Razonamiento: TRM vs. LLMs Líderes

Modelo	**Desarrollador**	**Parámetros (Estimación)**	**Paradigma Arquitectónico**	**ARC-AGI-2 (Accuracy)**	**Context Window (Tokens)**
Tiny Recursive Model (TRM)	Samsung SAIL Montréal	7 Millones [23]	Recursivo Iterativo (2 capas) [24]	8% (Alto SOTA) [23]	Bajo (Orientado a Puzzle)
DeepSeek R1	DeepSeek	671 Billones (MoE) [21]	Transformer MoE	< 8% [21]	Alto (N/A)
GPT-4o	OpenAI	$\sim$1.8 Billones (MoE) [35]	Transformer MoE (Multimodal)	< 8% [23]	128K [37]
Claude 3 Opus	Anthropic	$\sim$2 Billones (MoE) [38]	Transformer (Proprietary MoE)	< 8% [23]	200K–1M [37]

La Tabla 2 demuestra la **paradoja de la inteligencia general vs. profunda**. Modelos como GPT-4o y Claude 3 Opus están optimizados para manejar contextos extensos (128K a 1M de *tokens*) [37] y cubrir un vasto conocimiento (MMLU). Esta capacidad requiere miles de millones de parámetros. Sin embargo, en la métrica de razonamiento puro (ARC-AGI-2), el TRM, al ser miles de veces más pequeño, logra un rendimiento superior o comparable. Esto se debe a que el TRM ignora los requisitos de contexto extenso y generación masiva de texto, centrándose únicamente en el proceso de lógica a través de la recursión.[25]
Esto implica que la inteligencia artificial no puede ser dominada por una única arquitectura monolítica. Los LLMs son caros, lentos y están optimizados para el *contenido*, mientras que el TRM es extremadamente pequeño y está optimizado para la *ejecución lógica*. El futuro de los sistemas de IA de próxima generación requerirá la orquestación de **Modelos Tiny Especializados (TSMs)** como el TRM para tareas de lógica, combinados con SLMs generalistas (para conocimiento de dominio) y, solo si es estrictamente necesario, el uso de LLMs en la nube (*cloud fallback*).[40]

VI. Conclusiones Estratégicas y Perspectivas Futuras

VI.A. Implicaciones del TRM para la Estrategia de IA en Dispositivos

El *Tiny Recursive Model* de Samsung no es solo un logro académico; es una validación de la estrategia de **IA *On-Device***. Su éxito subraya la viabilidad de implementar capacidades de razonamiento complejo directamente en dispositivos móviles, lo cual es fundamental para el desarrollo de la línea Galaxy AI de Samsung.[7] El modelo proporciona una alternativa sostenible y altamente eficiente en términos de parámetros a la carrera armamentista de escala que ha dominado la industria, permitiendo que la IA avanzada sea ejecutada en *hardware* de bajo consumo.[21]

El logro del TRM valida la inversión continua de Samsung en su infraestructura de *hardware* local. El diseño recursivo del modelo está perfectamente adaptado para maximizar la eficiencia de los ciclos de cómputo repetitivos, explotando de manera óptima las Unidades de Procesamiento Neural (NPUs) que la compañía ha desarrollado.[5]

VI.B. La Dualidad Arquitectónica: Profundidad Física vs. Profundidad Efectiva

La principal conclusión arquitectónica del TRM es el establecimiento de una **dualidad entre la profundidad física del modelo y su profundidad efectiva de razonamiento**. Los *Transformers* tradicionales logran la profundidad apilando capas (millones o billones de parámetros). El TRM, en cambio, utiliza solo dos capas, pero logra una profundidad efectiva superior a través de la recurrencia dinámica.

La recursión permite al TRM ajustar dinámicamente la "profundidad" de su razonamiento (los K pasos de mejora) en función de la complejidad de la tarea [24], un contraste directo con la profundidad estática de una red neuronal profunda. Si bien este *trade-off* de "parámetros por pases" [25] podría introducir una latencia total ligeramente mayor en la inferencia debido a las repeticiones, resulta en una eficiencia de memoria óptima y una precisión final superior en tareas de alta lógica.

VI.C. Recomendaciones para el Desarrollo Futuro de Modelos de Micro IA Especializados

El rendimiento del TRM y otros modelos recurrentes sugiere un cambio fundamental en el desarrollo de la Micro IA. La comunidad de investigación y desarrollo debe considerar las siguientes estrategias:

1. **Fomento de TSMs (Tiny Specialized Models):** La industria no debe limitarse a la compresión de *Transformers* masivos. Es crucial priorizar la investigación y el desarrollo de arquitecturas no convencionales (recursivas o recurrentes) que optimicen la lógica

para tareas específicas, ya que estas pueden lograr una mayor capacidad de razonamiento que los modelos de propósito general, con una huella de memoria mínima.

2. **Orquestación Híbrida Cloud-Edge:** Los sistemas de IA deben diseñarse para aprovechar las fortalezas de cada plataforma, combinando la baja latencia y alta privacidad de los SLMs y TSMs locales con la capacidad de generalización y almacenamiento de contexto masivo de los LLMs en la nube, implementando un sistema de delegación de tareas (Cloud fallback).[40]
3. **Prioridad a la Recurrencia en la Inferencia:** El éxito del TRM de 7 millones de parámetros en lógica [23] indica que la capacidad de razonamiento complejo en dispositivos pequeños se basa en la **recurrencia** y la reutilización eficiente de pesos, más que en la simple expansión de la red. Esto sugiere que las futuras Unidades de Procesamiento Neural (NPUs) deberían ser diseñadas específicamente para acelerar los ciclos de recursión de alta tasa de repetición, maximizando la eficiencia de los modelos Tiny en entornos Edge.
4. **Estandarización de Benchmarks de Razonamiento:** La comunidad necesita expandir y estandarizar el uso de métricas que castiguen la memorización y recompensen la verdadera capacidad de solución de problemas, como el ARC-AGI, para impulsar la innovación en arquitecturas de modelos eficientes y especializadas sobre la simple escalabilidad.[22]

Obras citadas

1. Edge AI Solutions - Supermicro, fecha de acceso: octubre 17, 2025, https://www.supermicro.com/en/solutions/edge-ai
2. What Is Edge AI? Navigating Artificial Intelligence at the Edge - F5 Networks, fecha de acceso: octubre 17, 2025, https://www.f5.com/glossary/what-is-edge-ai
3. What is Edge AI, Its features, advantages & use cases, fecha de acceso: octubre 17, 2025, https://www.micro.ai/blog/edge-ai-what-is-it-and-how-does-it-work
4. Small Language Models (SLMs) for Efficient Edge Deployment - Prem AI, fecha de acceso: octubre 17, 2025, https://blog.premai.io/small-language-models-slms-for-efficient-edge-deployment/
5. [All About Exynos] ② An Upgraded Mobile Experience: The Important Role of CPU and NPU in Smartphones | Samsung Semiconductor Global, fecha de acceso: octubre 17, 2025, https://semiconductor.samsung.com/news-events/tech-blog/all-about-exynos-2-an-upgraded-mobile-experience-the-important-role-of-cpu-and-npu-in-smartphones/
6. Edge AI – A More Local and Secure Approach to AI, fecha de acceso: octubre 17, 2025, https://www.micro.ai/blog/edge-ai-a-more-local-and-secure-approach-to-ai
7. Beyond the Cloud: A Deep Dive Into On-Device Generative AI - Samsung Semiconductor, fecha de acceso: octubre 17, 2025,

https://semiconductor.samsung.com/news-events/tech-blog/beyond-the-cloud-a-deep-dive-into-on-device-generative-ai/
8. Conceptos: modelos de lenguaje pequeños y grandes - Azure Kubernetes Service, fecha de acceso: octubre 17, 2025, https://learn.microsoft.com/es-es/azure/aks/concepts-ai-ml-language-models
9. Los 5 mejores modelos de lenguaje pequeños y sus mejores casos de uso. - eesel AI, fecha de acceso: octubre 17, 2025, https://www.eesel.ai/es/blog/small-language-models
10. Ultimate Guide - The Best Small LLMs for On-Device Chatbots in 2025 - SiliconFlow, fecha de acceso: octubre 17, 2025, https://www.siliconflow.com/articles/en/best-small-LLMs-for-on-device-chatbots
11. Which is the best model out of these? : r/LocalLLaMA - Reddit, fecha de acceso: octubre 17, 2025, https://www.reddit.com/r/LocalLLaMA/comments/1g1vug8/which_is_the_best_model_out_of_these/
12. library - Ollama, fecha de acceso: octubre 17, 2025, https://ollama.com/library
13. ¿Qué son los modelos de lenguaje pequeños (SLM)? - IBM, fecha de acceso: octubre 17, 2025, https://www.ibm.com/es-es/think/topics/small-language-models
14. Are Local LLMs on Mobile a Gimmick? The Reality in 2025 - Callstack, fecha de acceso: octubre 17, 2025, https://www.callstack.com/blog/local-llms-on-mobile-are-a-gimmick
15. On-Device Language Models: A Comprehensive Review - arXiv, fecha de acceso: octubre 17, 2025, https://arxiv.org/html/2409.00088v2
16. NPU (Neural Processing Units) | Samsung Semiconductor Global, fecha de acceso: octubre 17, 2025, https://semiconductor.samsung.com/support/tools-resources/dictionary/the-neural-processing-unit-npu-a-brainy-next-generation-semiconductor/
17. Reversing and Exploiting Samsung's Neural Processing Unit - Longterm Security, fecha de acceso: octubre 17, 2025, https://blog.longterm.io/samsung_npu.html
18. Samsung Brings New AI Power to Its Interactive Display at Bett 2025, fecha de acceso: octubre 17, 2025, https://news.samsung.com/global/samsung-brings-new-ai-power-to-its-interactive-display-at-bett-2025
19. Samsung Neural SDK, fecha de acceso: octubre 17, 2025, https://developer.samsung.com/neural/overview.html
20. MMLU-Pro scores of small models (<5B) : r/LocalLLaMA - Reddit, fecha de acceso: octubre 17, 2025, https://www.reddit.com/r/LocalLLaMA/comments/1gii24g/mmlupro_scores_of_small_models_5b/
21. Samsung takes on OpenAI and Google with tiny, 7 million-parameter AI model, fecha de acceso: octubre 17, 2025, https://indianexpress.com/article/technology/artificial-intelligence/samsung-takes-on-openai-and-google-with-tiny-7-million-parameter-ai-model-10306145/
22. Samsung's tiny AI model beats giant reasoning LLMs - AI News, fecha de acceso:

octubre 17, 2025, https://www.artificialintelligence-news.com/news/samsung-tiny-ai-model-beats-giant-reasoning-llms/
23. Less is More: Recursive Reasoning with Tiny Networks - arXiv, fecha de acceso: octubre 17, 2025, https://arxiv.org/html/2510.04871v1
24. SamsungSAILMontreal/TinyRecursiveModels - GitHub, fecha de acceso: octubre 17, 2025, https://github.com/SamsungSAILMontreal/TinyRecursiveModels
25. Samsung's impressive tiny AI win - The Neuron, fecha de acceso: octubre 17, 2025, https://www.theneurondaily.com/p/samsung-s-impressive-tiny-ai-win
26. From HRM to TRM : r/singularity - Reddit, fecha de acceso: octubre 17, 2025, https://www.reddit.com/r/singularity/comments/1o0n1n1/from_hrm_to_trm/
27. Paper page - Less is More: Recursive Reasoning with Tiny Networks - Hugging Face, fecha de acceso: octubre 17, 2025, https://huggingface.co/papers/2510.04871
28. TRM - Radical improvement over Transformers ! - FiveTech Software tech support forums, fecha de acceso: octubre 17, 2025, https://fivetechsupport.com/forums/viewtopic.php?t=46014
29. Less is More: Recursive Reasoning with Tiny Networks - arXiv, fecha de acceso: octubre 17, 2025, https://arxiv.org/pdf/2510.04871
30. What Are LLM Benchmarks? - IBM, fecha de acceso: octubre 17, 2025, https://www.ibm.com/think/topics/llm-benchmarks
31. Llama 3.2 Benchmark Insights and Revolutionizing Edge AI and Vision - Medium, fecha de acceso: octubre 17, 2025, https://medium.com/towards-agi/llama-3-2-benchmark-insights-and-revolutionizing-edge-ai-and-vision-88542fe3dc0d
32. 30 LLM evaluation benchmarks and how they work - Evidently AI, fecha de acceso: octubre 17, 2025, https://www.evidentlyai.com/llm-guide/llm-benchmarks
33. Battle of the SLMs: Gemma vs LLama - Embedl, fecha de acceso: octubre 17, 2025, https://www.embedl.com/knowledge/battle-of-the-slms-gemma-vs-llama
34. The 11 best open-source LLMs for 2025 - n8n Blog, fecha de acceso: octubre 17, 2025, https://blog.n8n.io/open-source-llm/
35. Number of Parameters in GPT-4 (Latest Data) - Exploding Topics, fecha de acceso: octubre 17, 2025, https://explodingtopics.com/blog/gpt-parameters
36. Parameter Size of GPT-4o and Claude 3.5 Sonnet : r/singularity - Reddit, fecha de acceso: octubre 17, 2025, https://www.reddit.com/r/singularity/comments/1hdn2bs/parameter_size_of_gpt4o_and_claude_35_sonnet/
37. LLM Models Comparison: GPT-4o, Gemini, LLaMA | Deepchecks, fecha de acceso: octubre 17, 2025, https://www.deepchecks.com/llm-models-comparison/
38. AI Model Parameter Counts: A Comprehensive Analysis - Claude, fecha de acceso: octubre 17, 2025, https://claude.ai/public/artifacts/0ecdfb83-807b-4481-8456-8605d48a356c
39. Introducing the next generation of Claude - Anthropic, fecha de acceso: octubre 17, 2025, https://www.anthropic.com/news/claude-3-family
40. Edge-First Language Model Inference: Models, Metrics, and Tradeoffs - arXiv,

fecha de acceso: octubre 17, 2025, https://arxiv.org/html/2505.16508v1