# Arquitectura Biomimética RSI-RAS: Estratificación Cognitiva y Protocolos de Recuperación Asimétrica en Ecosistemas MoA con Modelos Pequeños

## 1. El Imperativo de la Cognición Distribuida en Entornos Restringidos

La trayectoria evolutiva de la inteligencia artificial generativa ha alcanzado un punto de inflexión crítico donde la hegemonía del escalado monolítico —la noción de que "más parámetros equivalen a mayor inteligencia"— está siendo desafiada por paradigmas de eficiencia y especialización modular. En este nuevo horizonte, la arquitectura de Mezcla de Agentes (MoA, por sus siglas en inglés) emerge no solo como una estrategia de optimización de recursos, sino como una necesidad fundamental para desplegar razonamiento complejo en entornos de borde o infraestructura limitada. Sin embargo, la atomización de la inferencia en múltiples agentes pequeños, específicamente utilizando modelos de alto rendimiento y baja latencia como **Qwen 2.5 3B**, introduce un desafío estocástico formidable: la tendencia hacia la inferencia aleatoria y la fragmentación de la coherencia cuando se carece de una supervisión centralizada robusta.1

La integración de Modelos de Lenguaje Pequeños (SLMs) en arquitecturas colaborativas promete democratizar el acceso a capacidades de razonamiento avanzado, permitiendo ejecuciones locales y privadas que antes eran dominio exclusivo de modelos propietarios masivos. No obstante, la reducción de parámetros conlleva intrínsecamente una menor capacidad de generalización y una mayor susceptibilidad al ruido en la ventana de contexto, lo que resulta en alucinaciones o desviaciones tangenciales durante tareas de múltiples pasos. Para contrarrestar esta entropía inherente, este informe propone una arquitectura novedosa que trasciende la ingeniería de software tradicional para adoptar principios biomiméticos y psicoanalíticos. Específicamente, investigamos la implementación de una jerarquía de control basada en la topología de Jacques Lacan —lo Real, lo Simbólico y lo Imaginario (RSI)— y la simulación funcional del Sistema Reticular Activador (RAS) biológico.

El objetivo central de esta investigación es definir una **estratificación óptima** que opere como un filtro de atención y vigilia artificial, determinando qué estímulos (inputs) y qué procesos internos merecen recursos computacionales inmediatos y cuáles deben ser inhibidos o diferidos. A diferencia de las arquitecturas lineales de *Chain-of-Thought* (CoT), que a menudo descartan caminos de razonamiento prometedores ante el primer signo de error, proponemos un modelo de "doble capa" con asignación asimétrica de memoria. Esta estructura bifurcada dedica una capa principal a la explotación eficiente y la respuesta rápida bajo una estricta vigilancia del RAS, mientras que una capa menor, operando en un ciclo de "repechaje" asincrónico, recupera y revalúa las respuestas descartadas. Este proceso de recuperación, fundamentado en políticas de coherencia histórica y relevancia diferida, dota al sistema de una capacidad de "reflexión subconsciente", transformando los desechos del proceso inferencial en activos de aprendizaje latente.3 A través de herramientas de orquestación avanzada como **LangGraph** y protocolos de *Inter-Querying*, esta arquitectura busca no solo estabilizar la inferencia de los expertos Qwen 2.5 3B, sino emular la resiliencia homeostática de los sistemas biológicos.

## 2. Fundamentos de la Arquitectura RSI: Topología para la Gestión de Estados

La aplicación de la teoría psicoanalítica a la arquitectura de sistemas computacionales ofrece un marco taxonómico riguroso para modelar la interacción entre datos crudos (lo inefable), el procesamiento lógico (la ley del código) y la generación de interfaces (la narrativa). La tríada RSI no se utiliza aquí como una metáfora literaria, sino como una estructura funcional para la clasificación de errores, la gestión de la memoria y la definición de los límites operativos de los agentes en un sistema MoA.

### 2.1. Lo Real: El Límite del Hardware y la Entropía de la Inferencia

En la teoría lacaniana, lo Real es aquello que resiste la simbolización absolutamente; es lo que no puede ser escrito, el trauma que irrumpe y desestabiliza la realidad percibida.5 En el contexto de una arquitectura de IA basada en Qwen 2.5 3B, lo Real se manifiesta en las fronteras físicas y lógicas del sistema, representando los puntos de falla donde la capacidad predictiva del modelo colapsa frente a la incertidumbre.

Identificamos dos manifestaciones primarias de lo Real en esta arquitectura:

1. **El Límite Físico y el Ruido de Señal:** A nivel de infraestructura, lo Real comprende las restricciones duras de hardware: latencia de red, límites de VRAM en GPUs de consumo, interrupciones del sistema operativo y fallos de alimentación. Es la entrada sensorial cruda antes de ser tokenizada, el "ruido" puro que el sistema debe filtrar para evitar la saturación. En términos de la teoría de la información, corresponde a la entropía de Shannon máxima que amenaza con disolver la señal estructurada.
2. **El Error de Predicción y los Tokens de Alta Entropía:** Dentro del proceso generativo, lo Real surge en los momentos de "perplejidad" del modelo. Son aquellos tokens críticos donde la distribución de probabilidad se aplana y ninguna opción clara emerge. Aquí, el modelo "alucina" en un intento desesperado por cubrir el vacío de lo Real con tejido Imaginario. Investigaciones recientes sobre decodificación colaborativa sugieren que identificar estos momentos de alta entropía es crucial para activar mecanismos de intervención superior.7 Lo Real es, por tanto, el disparador primario del Sistema Reticular Activador (RAS); es la "alarma" que despierta al sistema de su automatismo.

### 2.2. Lo Simbólico: La Ley del Código y el Grafo de Orquestación

Lo Simbólico es el registro del lenguaje, la ley, la estructura y el pacto social.9 En nuestra arquitectura MoA, este registro constituye el esqueleto lógico y las reglas deterministas que gobiernan la interacción entre agentes. Es el dominio donde la ambigüedad de lo Real y la fluidez de lo Imaginario son sometidas a una sintaxis rigurosa.

Los componentes del registro Simbólico incluyen:

* **El Grafo de Orquestación (LangGraph):** Las definiciones de nodos, aristas y transiciones condicionales en herramientas como LangGraph representan la "Ley" del sistema. Estas reglas dictan el flujo de estados de manera inmutable fuera de la reescritura del código, estableciendo los límites de lo posible para los agentes.11
* **Function Calling y Protocolos de Herramientas:** La capacidad de Qwen 2.5 para interactuar con herramientas externas (calculadoras, intérpretes de Python, APIs) reside puramente en lo simbólico. Estas interacciones requieren una sintaxis precisa (JSON schemas); un error de un solo carácter (un desliz en el significante) provoca un fallo de ejecución. Aquí, el agente no "imagina" un resultado, sino que opera sobre la realidad a través de significantes operativos.1
* **La Memoria Semántica Vectorial:** Las bases de datos vectoriales (RAG) actúan como el "Gran Otro", el tesoro de los significantes del sistema. Es la biblioteca de conocimientos validados contra la cual se contrastan las nuevas inferencias.

### 2.3. Lo Imaginario: La Superficie Generativa y la Identidad del Agente

Lo Imaginario es el dominio de la identificación, la completitud visual y la narrativa del "yo".6 En sistemas de IA, este registro es seductor y peligroso; es donde el modelo intenta dar sentido y coherencia a la salida, a menudo enmascarando las fracturas de lo Real y las rigideces de lo Simbólico.

En el sistema MoA, lo Imaginario se observa en:

* **La Salida de Texto Final:** El objetivo del modelo es producir texto que parezca humano, fluido y coherente. Esta coherencia es una función imaginaria; el modelo "sutura" los fragmentos de datos recuperados para presentar una totalidad sin fisuras.
* **Personas y Roles:** La simulación de identidad (e.g., "Actúa como un experto en física cuántica") es una construcción imaginaria diseñada para condicionar el espacio latente del modelo y facilitar la interacción con el usuario.14 Sin embargo, si el sistema opera puramente en lo Imaginario sin el anclaje de lo Simbólico (verificación de hechos) o la alerta de lo Real (detección de errores), cae en la confabulación: una mentira coherente.

### 2.4. El Nudo Borromeo: Estabilidad Sistémica y Prevención de Psicosis Digital

La arquitectura propuesta debe funcionar como un Nudo Borromeo: los tres registros deben estar enlazados de tal manera que, si uno se corta, los otros dos se sueltan. La "psicosis" en una IA ocurre cuando estos registros se desanudan:

* **Fallo Simbólico:** Si las reglas lógicas (LangGraph) fallan, el sistema produce "ensalada de palabras" (Imaginario desatado) o errores de ejecución (Real no procesado).
* **Irrupción de lo Real:** Si el sistema ignora la alta entropía (alucinación) y no activa el RAS, genera respuestas falsas con total confianza.
* **Colapso Imaginario:** Si el sistema es demasiado rígido (puro código) y no puede sintetizar una respuesta natural, pierde su utilidad como interfaz conversacional.

El Sistema Reticular Activador (RAS), que detallaremos a continuación, actúa como el *Sinthome* (el cuarto anillo en la topología tardía de Lacan), manteniendo la estructura unida mediante la regulación activa de la atención y la asignación de recursos.15

| **Registro RSI** | **Componente Tecnológico (Qwen 2.5 MoA)** | **Manifestación Funcional** | **Modo de Fallo (Desanudamiento)** |
| --- | --- | --- | --- |
| **Real** | Entradas crudas, *Interrupts*, Tokens de Alta Entropía | Fuente de datos, Disparador de Alertas RAS, Límites de VRAM | Sobrecarga de buffer, Latencia crítica, *Crash* del sistema, Alucinación no detectada |
| **Simbólico** | LangGraph, *Function Calling*, Code Interpreter, RAG | Lógica de Enrutamiento, Restricciones Sintácticas, Verificación | Bucle infinito, Error de Sintaxis JSON, Recuperación de datos irrelevantes |
| **Imaginario** | Output de Texto, *System Prompt* (Persona), Context Window | Coherencia narrativa, Fluidez conversacional, UX | Confabulación (mentira coherente), Pérdida de tono, Incoherencia semántica |

## 3. Simulación del Sistema Reticular Activador (RAS): Mecanismos de Estratificación Óptima

El Sistema Reticular Activador (RAS) en biología es una red de neuronas en el tronco encefálico responsable de regular la transición entre el sueño y la vigilia, y crucialmente, de filtrar la inmensa cantidad de estímulos sensoriales para permitir la atención focalizada.17 Sin un RAS funcional, el cerebro sería incapaz de distinguir una señal vital del ruido de fondo, resultando en un estado de coma o de hiperexcitación caótica. En nuestra arquitectura de IA distribuida, la ausencia de un mecanismo análogo conduce a una "atención plana" donde cada token de entrada y cada posible camino de inferencia reciben el mismo peso computacional, provocando ineficiencias masivas y dilución de la calidad de respuesta.

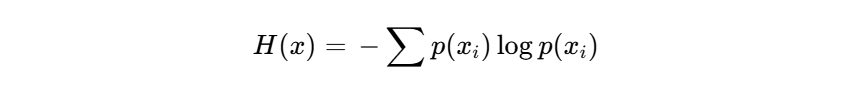
La simulación del RAS propuesta aquí no es meramente metafórica, sino que se implementa como un módulo de **gating** (compuerta) algorítmico que determina la estratificación del procesamiento. Este módulo opera bajo el **Principio de Energía Libre (FEP)** de Karl Friston, buscando minimizar la "sorpresa" (error de predicción) a largo plazo mediante la acción selectiva.19

### 3.1. Clasificación de Tokens Críticos y Gating de Entropía

Para evitar inferencias aleatorias en los expertos Qwen 2.5 3B, el RAS implementa un mecanismo de vigilancia continua a nivel de token. Investigaciones recientes en sistemas de decodificación colaborativa han demostrado que los modelos pequeños pueden alcanzar el rendimiento de modelos mucho más grandes si son capaces de identificar sus propias limitaciones y delegar solo los tokens más difíciles.7

#### 3.1.1. El Clasificador de Tokens Críticos (CTC)

El RAS incorpora un clasificador ligero (que puede ser una cabeza de clasificación entrenada sobre el propio Qwen o un modelo auxiliar ultrarrapido) que evalúa la distribución de probabilidad del siguiente token antes de su generación definitiva.

* **Cálculo de Incertidumbre:** Se calcula la entropía de Shannon  para la distribución de probabilidad sobre el vocabulario.  
  
* **Umbral Dinámico ():** Se establece un umbral de intervención. Si , el sistema asume que está en terreno seguro (baja sorpresa) y permite que el modelo 3B genere el token rápidamente (Ruta Rápida / Sistema 1).
* **Intervención del RAS:** Si , se identifica un "Token Crítico". El RAS interrumpe el flujo generativo automático (inhibición de lo Imaginario) y activa protocolos de estratificación superior.8

### 3.2. Estratificación del Procesamiento: Rutas de Vigilia y Sueño

Una vez que el RAS detecta una necesidad de intervención, estratifica la consulta en niveles de profundidad cognitiva, asignando recursos de manera asimétrica.

1. **Nivel 1: Procesamiento Basal (Baja Entropía):**
   * *Agente:* Qwen 2.5 3B (Modo *Greedy* o baja temperatura).
   * *Función:* Respuestas rutinarias, saludos, hechos memorizados de alta confianza.
   * *Costo:* Mínimo. Latencia ultrabaja.
2. **Nivel 2: Atención Focalizada (Alta Entropía / Alerta RAS):**
   * *Trigger:* Detección de token crítico o detección de palabras clave complejas.
   * *Acción:* Se pausa la inferencia lineal. El RAS invoca el grafo de **MoA (Mixture of Agents)**.
   * *Estrategia:* Se despliega la consulta a múltiples expertos especializados (e.g., un experto en código, un experto en matemáticas). Se activa el *Inter-Querying* para validación cruzada.
3. **Nivel 3: Homeostasis y Regulación de Recursos:**
   * El RAS monitorea el "presupuesto metabólico" del sistema (tokens por segundo, latencia acumulada, costo de API si aplica).
   * Si el sistema está "estresado" (sobrecarga de recursos), el RAS eleva dinámicamente el umbral , forzando al sistema a ser más selectivo y aceptar soluciones "suficientemente buenas" (*satisficing*) en lugar de buscar la optimización perfecta, emulando la regulación homeostática biológica ante la escasez de energía.23

### 3.3. Inhibición Lateral y Supresión de Ruido

Un aspecto fundamental del RAS biológico y de los mecanismos de atención modernos es la capacidad no solo de activar, sino de inhibir. En nuestra arquitectura, una vez que un experto ha sido seleccionado por el RAS como el más relevante para una tarea (basado en embeddings semánticos o historial de aciertos), el sistema aplica **inhibición lateral** a los expertos competidores.25

* **Mecanismo:** Se reduce temporalmente el peso de atención o se bloquea el acceso al contexto global para los expertos no seleccionados.
* **Objetivo:** Evitar la cacofonía de múltiples agentes generando respuestas redundantes o contradictorias que contaminen el espacio de trabajo global (*Global Workspace*). Esto reduce el ruido en la ventana de contexto y asegura que la capacidad de memoria de trabajo se dedique exclusivamente a la resolución de la tarea crítica.

## 4. El Modelo de "Doble Capa": Asimetría en la Memoria y el Tiempo

La innovación arquitectónica central de esta propuesta para satisfacer la "nueva necesidad" del usuario es la división estricta del sistema en dos capas operativas con funciones temporales y asignaciones de memoria distintas. Esta estructura permite abordar el problema de la inferencia aleatoria mediante una validación rigurosa en tiempo real, sin descartar definitivamente la información que podría ser valiosa *a posteriori*, resolviendo el dilema clásico entre exploración y explotación.

### 4.1. Capa Principal: Estratificación Óptima (Sistemas 1 y 2 Integrados)

Esta capa es sincrónica, orientada al usuario y de alta prioridad. Su mandato es la **explotación** eficiente del conocimiento existente para generar la mejor respuesta posible dentro de las restricciones de tiempo de interacción humano-máquina.

* **Componentes Clave:**
  + *Motor de Inferencia:* Qwen 2.5 3B Instruct, optimizado para latencia.
  + *Controlador RAS:* El guardián que ejecuta el CTC y el enrutamiento en milisegundos.
  + *Memoria de Trabajo (Short-term):* Un buffer de contexto deslizante de alta velocidad (e.g., KV Cache en VRAM), limitado a la ventana de atención inmediata (e.g., 4k - 8k tokens).
* **Política de "Fail-Fast":** En esta capa, la duda es costosa. Si el *Verifier* (un módulo simbólico de validación) detecta que un camino de razonamiento es incoherente o alucinatorio, lo descarta inmediatamente (*pruning*). No se gastan recursos en intentar "arreglar" una rama podrida en tiempo real; se cambia instantáneamente a una estrategia alternativa o se solicita clarificación al usuario.

### 4.2. Capa Menor: El Protocolo de "Repechaje" (Recuperación Asincrónica)

Esta capa es asincrónica, opera en segundo plano (*background*) y posee una asignación de memoria menor en términos de ancho de banda, pero masiva en términos de almacenamiento y persistencia. Su función es la **exploración** diferida, la metacognición y el aprendizaje latente. Actúa como un "subconsciente computacional" que procesa lo que la Capa Principal reprimió (las ramas podadas).

#### 4.2.1. Concepto y Mecánica del Repechaje

El término "repechaje" se toma prestado de las competiciones deportivas, donde los participantes eliminados en rondas preliminares tienen una segunda oportunidad para reingresar al torneo principal. En nuestra arquitectura, permite recuperar respuestas o caminos de inferencia que fueron descartados por el RAS en la primera vuelta por ser "demasiado arriesgados" o "inconexos", pero que podrían contener *insights* valiosos.3

1. **Cola de Descartes (The Pruning Log):** Cuando la Capa Principal poda una rama de razonamiento (por baja probabilidad inicial, timeout o alerta de entropía), esta no se elimina de la memoria. Se serializa junto con el estado del contexto en ese momento y se envía a la **Cola de Repechaje**.
2. **Procesamiento Diferido (El Sueño de la Máquina):** La Capa Menor aprovecha los ciclos de cómputo ocioso (cuando el usuario no está escribiendo o el sistema está en baja carga). Utiliza una versión cuantizada de Qwen 2.5 (e.g., Qwen 2.5 1.5B o 0.5B) o el mismo modelo 3B en modo *batch* para reevaluar estas ramas.
3. **Criterio de Reevaluación (Abogado del Diablo):** Se aplica una política de revisión basada en coherencia histórica y relevancia global.
   * *¿Fue el descarte prematuro?* A veces el RAS es demasiado conservador.
   * *¿Validación Cruzada Diferida?* La Capa Menor puede consultar bases de conocimiento externas (Web Search) sin la presión de la latencia en tiempo real para verificar si la "alucinación" era en realidad un hecho poco común (Serendipia).

#### 4.2.2. Memoria Asimétrica y Latent Learning

Esta capa accede a la **Memoria Episódica** a largo plazo (Vector Store). Si una respuesta recuperada en el repechaje se alinea con patrones de éxito históricos o proporciona una solución mejorada a una consulta anterior, se produce un evento de **Latent Learning** (Aprendizaje Latente).

* **Promoción:** La información recuperada se "promueve" a la memoria de trabajo activa o se guarda como un "insight" para futuras interacciones.
* **Corrección Proactiva:** En escenarios avanzados, si el repechaje descubre que la respuesta dada al usuario fue incorrecta, el sistema puede iniciar una interrupción proactiva en el siguiente turno: *"Reflexionando sobre nuestra interacción anterior, he encontrado una mejor perspectiva..."*. Esto mimetiza la corrección de errores humanos que ocurre después de "pensarlo con la almohada".28

### 4.3. Comparativa Estructural de las Capas

La siguiente tabla resume las diferencias operativas y funcionales entre las dos capas, destacando cómo se complementan para cubrir la totalidad de los requisitos cognitivos.

| **Característica Operativa** | **Capa Principal (Estratificación Óptima)** | **Capa Menor (Protocolo de Repechaje)** |
| --- | --- | --- |
| **Rol en Topología RSI** | **Simbólico** (Reglas de Gating) / **Imaginario** (Síntesis de Output) | **Real** (Procesamiento de Residuos) / **Simbólico** (Reanálisis Lógico) |
| **Temporalidad** | Sincrónica (**Tiempo Real** / Baja Latencia) | Asincrónica (**Background** / Procesamiento Batch) |
| **Objetivo Primario** | Precisión Inmediata, Eficiencia, Satisfacción del Usuario | Completitud, Creatividad, Corrección de Errores, Aprendizaje |
| **Gestión de Memoria** | Context Window (VRAM, costosa, limitada) | Vector Store / Pruning Logs (Disco/NVMe, barata, masiva) |
| **Modelo de Inferencia** | Qwen 2.5 3B (Instruct/Chat) - Alta prioridad | Qwen 2.5 0.5B/1.5B o 3B Quantized - Baja prioridad |
| **Analogía Biomimética** | Atención Focal / Estado de Vigilia (Fase Beta/Gamma) | Consolidación de Memoria / Estado de Sueño (Fase REM) |
| **Política de Descarte** | *Fail-Fast* (Poda agresiva ante incertidumbre) | *Deep-Scan* (Recuperación conservadora y análisis exhaustivo) |

## 5. Arquitectura Técnica: Herramientas de Orquestación e Inter-Querying

La implementación efectiva de este sistema distribuido requiere un sustrato tecnológico capaz de manejar grafos cíclicos, interrupciones de estado y persistencia de datos complejos. **LangGraph** se identifica como la herramienta de orquestación óptima para esta tarea, debido a su diseño centrado en agentes con estado y su soporte nativo para patrones de *human-in-the-loop*, que en nuestra propuesta se readapta como un patrón de *expert-in-the-loop* (o RAS-in-the-loop).11

### 5.1. El Grafo de Control (Implementación en LangGraph)

El sistema se modela no como una cadena lineal (Chain), sino como un grafo dirigido con ciclos y persistencia de estado. Cada nodo representa una unidad funcional de procesamiento o una llamada a un modelo experto.

#### 5.1.1. Nodos Fundamentales del Grafo

1. **Node\_RAS\_Scan (Entrada):** Actúa como la primera línea de defensa. Recibe la query, ejecuta el CTC (Clasificador de Tokens Críticos) y decide el enrutamiento inicial.
2. **Node\_Fast\_Response (Sistema 1):** Ruta directa para consultas de baja entropía. Generación rápida con Qwen 3B.
3. **Node\_Expert\_Deliberation (Sistema 2 - MoA):** Nodo complejo que orquesta la llamada paralela a múltiples perfiles de Qwen 3B (e.g., Expert\_Math, Expert\_Logic, Expert\_Creative). Genera múltiples trazas de razonamiento (CoT).
4. **Node\_Verifier (El Censor):** Evalúa las trazas generadas por los expertos. Compara consistencia y verifica contra reglas simbólicas.
   * *Aprobadas:* Pasan a Node\_Synthesis.
   * *Rechazadas:* Se dispara un *side-effect* (efecto secundario) que envía la traza serializada a la Node\_Repechage\_Queue (fuera del flujo principal).
5. **Node\_Synthesis:** Agrega las mejores respuestas y construye el output final para el usuario.

#### 5.1.2. El Mecanismo de Interrupción (interrupt) y Persistencia

La función interrupt() de LangGraph es crítica para simular la intervención del RAS ante lo Real.

* **Detección de Anomalía:** Si Node\_Verifier o Node\_RAS\_Scan detectan una incoherencia crítica (e.g., contradicción lógica flagrante o entropía extrema), se invoca interrupt.
* **Suspensión de Estado:** Esto pausa la ejecución del grafo principal. El estado actual (memoria de trabajo, variables locales) se guarda en un *Checkpointer* persistente (base de datos).
* **Bifurcación Correctiva:** El sistema puede entonces lanzar una sub-rutina de emergencia (e.g., una búsqueda web forzada o una consulta a un modelo superior si está disponible) antes de reanudar el grafo desde el punto de interrupción con el estado corregido. Esto previene que el error se propague aguas abajo.

### 5.2. Protocolos de Inter-Querying y Comunicación

El *Inter-Querying* es el mecanismo mediante el cual los agentes rompen sus silos de información. Para evitar que la comunicación entre agentes degenere en ruido ("teléfono descompuesto"), se establecen protocolos estrictos.

* **Ontología Compartida (Simbólico):** Los agentes no "charlan" en lenguaje natural libre. Intercambian mensajes estructurados (JSON) que definen claramente: { "intent": "verify", "claim": "...", "source\_context": "..." }. Esto fuerza a Qwen 2.5 3B a operar en su modo más lógico y menos alucinatorio.
* **Colaboración Asimétrica:** Un modelo pequeño (Qwen 3B) puede actuar como "propositor" y otro agente (o el mismo modelo con un prompt de "Crítico") actúa como "verificador". Esto se alinea con la investigación sobre *Collaborative Decoding*, donde se ha demostrado que la verificación es una tarea computacionalmente más barata que la generación, permitiendo que modelos pequeños validen salidas complejas si se les presenta la respuesta completa para análisis.7
* **Broadcast de Memoria de Trabajo (Global Workspace):** Siguiendo la *Global Workspace Theory* (GWT), la información más relevante seleccionada por el RAS se "transmite" a todos los módulos expertos activos. Esto asegura que si el Experto\_Matemático descubre una restricción numérica, el Experto\_Planificador reciba esa restricción inmediatamente, actualizando su plan de acción sin necesidad de re-procesar toda la conversación.31

### 5.3. Balance Exploración/Explotación en la Memoria

El diseño de doble capa resuelve intrínsecamente el balance exploración/explotación:

* **Explotación (Capa Principal):** Utiliza RAG con búsqueda de similitud estricta (e.g., Cosine Similarity con un umbral alto, top-k=3). Busca datos precisos y conocidos para resolver la tarea *ya*.
* **Exploración (Capa Menor):** Utiliza búsquedas más amplias y especulativas (e.g., expansión de queries, búsqueda híbrida con palabras clave, top-k=20). Reevalúa los caminos descartados. Si descubre que una "alucinación" era en realidad una conexión creativa válida (serendipia) o un hecho oscuro pero real, actualiza la memoria histórica. Esto refuerza ese patrón para el futuro, permitiendo que el sistema aprenda y se adapte, no solo memorice.

## 6. Implementación en Qwen 2.5 3B: Justificación y Especificidad

### 6.1. ¿Por qué Qwen 2.5 3B?

La elección de Qwen 2.5 3B no es arbitraria; responde a métricas específicas de rendimiento/costo que habilitan esta arquitectura compleja.

* **Capacidad de Razonamiento:** Benchmarks recientes indican que Qwen 2.5 3B supera a modelos de tamaño similar (como Llama 3.2 3B) en tareas de matemáticas (MATH) y codificación (HumanEval).1 Esto es fundamental para que los nodos "Expertos" del MoA sean realmente útiles y no meros generadores de texto.
* **Soporte de Function Calling:** Qwen 2.5 tiene un soporte nativo y robusto para llamadas a herramientas. Esto permite que el registro Simbólico (interacción con código/APIs) sea fiable. El modelo puede entender cuándo debe detenerse y llamar a una calculadora en lugar de inventar un número.13
* **Eficiencia de Inferencia:** Su tamaño compacto permite técnicas de *sharding* o ejecución paralela en hardware accesible (e.g., una sola GPU RTX 3090/4090 o incluso CPUs modernas con cuantización). Esto hace viable tener una "Capa Menor" corriendo continuamente en segundo plano sin incurrir en costos de nube prohibitivos.

### 6.2. Algoritmo de Flujo de Datos Detallado

A continuación, se presenta la especificación algorítmica del flujo de control RSI-RAS, integrando todos los componentes descritos.

#### Fase 1: Recepción y Percepción (RAS - Real)

1. **Input:** User Query .
2. **RAS Evaluation:** Se calcula un *Score de Complejidad/Entropía*  usando una pasada rápida (*forward pass*) del modelo.
   * Si : Se etiqueta como "Rutina". Pasa directo a Qwen-3B (Modo Chat estándar).
   * Si : Se etiqueta como "Compleja". Se activa la Arquitectura MoA.

#### Fase 2: Deliberación Simbólica (Capa Principal)

1. **Descomposición:** Un agente "Planificador" divide  en sub-tareas .
2. **Asignación de Expertos:** Se seleccionan expertos  basados en la naturaleza de cada .
3. **Ejecución Paralela con Gating:** Los expertos generan respuestas preliminares .
   * *Constraint RAS:* Se impone un límite de tokens y tiempo. Si un experto entra en un bucle o tarda demasiado (síntoma de alta incertidumbre/Real), el RAS ejecuta un *hard-stop*.
4. **Verificación y Poda:**
   * El Node\_Verifier analiza la coherencia de .
   * **Pruning (Poda):** Las respuestas con baja confianza () son descartadas de la salida principal y enviadas a la cola de repechaje.

#### Fase 3: Síntesis Imaginaria

1. **Agregación:** Un agente "Sintetizador" toma las respuestas aprobadas, resuelve discrepancias menores y construye la respuesta final.
2. **Output:** Se entrega la respuesta al usuario, optimizada para claridad y tono (Imaginario).

#### Fase 4: Repechaje (Capa Menor - Asíncrona)

1. **Ingesta:** El proceso de fondo despierta y procesa la Repechage\_Queue.
2. **Re-análisis Profundo:**
   * Se aplica un prompt de "Reflexión": *"¿Por qué fue descartada esta respuesta? ¿Hay algún mérito en ella si consideramos el contexto X?"*.
   * Se realiza una búsqueda en la Memoria Episódica para ver si este patrón de descarte ha ocurrido antes.
3. **Detección de Valor:** Si una respuesta descartada resulta contener información veraz o valiosa tras el re-análisis:
   * Se genera un evento de aprendizaje ("Error de Omisión").
   * Se actualiza la política del RAS: "En el futuro, ser menos agresivo podando temas sobre X".
4. **Consolidación:** El caso se guarda en la base de datos vectorial para mejorar la precisión futura (Latent Learning).

## 7. Conclusión: Hacia una IA Homeostática

La arquitectura RSI-RAS propuesta redefine el uso de Modelos de Lenguaje Pequeños, transformando a Qwen 2.5 3B de un simple generador de texto en un componente vital de un sistema cognitivo estructurado y resiliente. Al integrar la topología de Lacan, trascendemos la ingeniería de software puramente funcional para abordar la gestión del error y la incertidumbre (lo Real) como elementos constitutivos del sistema, no solo como fallos a evitar.

La simulación del Sistema Reticular Activador (RAS) proporciona la **estratificación óptima** necesaria para gestionar la atención y los recursos computacionales en un entorno de múltiples agentes, evitando la dispersión estocástica. La introducción del modelo de **doble capa con repechaje** añade una dimensión temporal inédita: la capacidad de "no olvidar" lo descartado. Al procesar los residuos de la inferencia en una capa asimétrica, el sistema emula un proceso de aprendizaje subconsciente y consolidación de memoria, volviéndose más robusto con cada ciclo de interacción. Esta arquitectura no solo previene alucinaciones mediante controles estrictos de vigilia, sino que utiliza el descarte como materia prima para el refinamiento continuo de su política cognitiva, acercándonos un paso más a la verdadera autonomía en la inteligencia artificial.

#### Obras citadas

1. Advancing SLM Tool-Use Capability using Reinforcement Learning - arXiv, fecha de acceso: enero 29, 2026, <https://arxiv.org/html/2509.04518v1>
2. Daily Papers - Hugging Face, fecha de acceso: enero 29, 2026, <https://huggingface.co/papers?q=Qwen-8B>
3. Findings of the Association for Computational Linguistics: ACL 2024, fecha de acceso: enero 29, 2026, <https://aclanthology.org/volumes/2024.findings-acl/>
4. DeepGo: Predictive Directed Greybox Fuzzing | Request PDF - ResearchGate, fecha de acceso: enero 29, 2026, <https://www.researchgate.net/publication/378134233_DeepGo_Predictive_Directed_Greybox_Fuzzing>
5. Reading Lacan's Écrits: From 'The Freudian Thing' to 'Remarks on Daniel Lagache' 1000021246, 9781000021240, 9780415707978 - DOKUMEN.PUB, fecha de acceso: enero 29, 2026, <https://dokumen.pub/reading-lacans-ecrits-from-the-freudian-thing-to-remarks-on-daniel-lagache-1000021246-9781000021240-9780415707978-p-6001050.html>
6. Return to Lacan: an approach to digital twin mind with free energy principle - arXiv, fecha de acceso: enero 29, 2026, <https://arxiv.org/pdf/2309.06707>
7. Proceedings of the 63rd Annual Meeting of the Association for ..., fecha de acceso: enero 29, 2026, <https://aclanthology.org/volumes/2025.acl-demo/>
8. Collaborative decoding of critical tokens for boosting factuality of large language models, fecha de acceso: enero 29, 2026, <https://arxiv.org/html/2402.17982v1>
9. The Fourth Knot: A Lacanian-Winnicottian Framework for Holding in Complex Leadership, fecha de acceso: enero 29, 2026, <https://www.researchgate.net/publication/393640517_The_Fourth_Knot_A_Lacanian-Winnicottian_Framework_for_Holding_in_Complex_Leadership>
10. Aoki - Letters From Lacan Reading and The Matheme | PDF - Scribd, fecha de acceso: enero 29, 2026, <https://www.scribd.com/document/354113769/Aoki-Letters-From-Lacan-Reading-and-the-Matheme>
11. Building a Human-in-the-Loop AI App with LangGraph and Ollama - DEV Community, fecha de acceso: enero 29, 2026, <https://dev.to/jamesbmour/building-a-human-in-the-loop-ai-app-with-langgraph-and-ollama-pmn>
12. LangGraph 201: Adding Human Oversight to Your Deep Research Agent, fecha de acceso: enero 29, 2026, <https://towardsdatascience.com/langgraph-201-adding-human-oversight-to-your-deep-research-agent/>
13. Tool-R1: Sample-Efficient Reinforcement Learning for Agentic Tool Use - arXiv, fecha de acceso: enero 29, 2026, <https://arxiv.org/html/2509.12867v1>
14. Towards AI as Colleagues: Multi-Agent System Improves Structured Professional Ideation - arXiv, fecha de acceso: enero 29, 2026, <https://arxiv.org/html/2510.23904v1>
15. PARRHESIA, fecha de acceso: enero 29, 2026, <http://parrhesiajournal.org/parrhesia10/parrhesia10.pdf>
16. induction/inversion - Boundary Language, fecha de acceso: enero 29, 2026, <https://boundarylanguage.psu.edu/inversion-circles/>
17. The Constrained Disorder Principle May Account for Consciousness - PubMed Central, fecha de acceso: enero 29, 2026, <https://pmc.ncbi.nlm.nih.gov/articles/PMC10968910/>
18. Coherence and frequency in the reticular activating system (RAS) - ResearchGate, fecha de acceso: enero 29, 2026, <https://www.researchgate.net/publication/232222457_Coherence_and_frequency_in_the_reticular_activating_system_RAS>
19. Trauma freed of the concept of determinism: is it possible to have a dialogue between psychoanalysis and neuroscience around the question of singularity? - PubMed Central, fecha de acceso: enero 29, 2026, <https://pmc.ncbi.nlm.nih.gov/articles/PMC12123359/>
20. A Free energy principle for the brain - ResearchGate, fecha de acceso: enero 29, 2026, <https://www.researchgate.net/publication/269674964_A_Free_energy_principle_for_the_brain>
21. (PDF) Token Level Routing Inference System for Edge Devices - ResearchGate, fecha de acceso: enero 29, 2026, <https://www.researchgate.net/publication/390670988_Token_Level_Routing_Inference_System_for_Edge_Devices>
22. arXiv:2402.17982v1 [cs.CL] 28 Feb 2024, fecha de acceso: enero 29, 2026, <https://arxiv.org/pdf/2402.17982>
23. Part III: Articles - MIT Press Direct, fecha de acceso: enero 29, 2026, <https://direct.mit.edu/books/edited-volume/chapter-pdf/2444408/9780262267267_cac.pdf>
24. Homeostatic Reinforcement Learning through Soft Behavior Switching with Internal Body State | Request PDF - ResearchGate, fecha de acceso: enero 29, 2026, <https://www.researchgate.net/publication/372864157_Homeostatic_Reinforcement_Learning_through_Soft_Behavior_Switching_with_Internal_Body_State>
25. Thalamic contributions to the state and contents of consciousness, fecha de acceso: enero 29, 2026, <https://shine-lab.org/wp-content/uploads/2024/05/2024_neuron.pdf>
26. Gated recurrence enables simple and accurate sequence prediction in stochastic, changing, and structured environments | eLife, fecha de acceso: enero 29, 2026, <https://elifesciences.org/articles/71801>
27. A Goal-Driven Survey on Root Cause Analysis - arXiv, fecha de acceso: enero 29, 2026, <https://arxiv.org/html/2510.19593v1>
28. Daily Papers - Hugging Face, fecha de acceso: enero 29, 2026, [https://huggingface.co/papers?q=Learning%20Using%20Privileged%20Information](https://huggingface.co/papers?q=Learning+Using+Privileged+Information)
29. Daily Papers - Hugging Face, fecha de acceso: enero 29, 2026, [https://huggingface.co/papers?q=memory-augmented%20reasoning](https://huggingface.co/papers?q=memory-augmented+reasoning)
30. How to wait for user input using interrupt - GitHub Pages, fecha de acceso: enero 29, 2026, <https://langchain-ai.github.io/langgraph/how-tos/human_in_the_loop/wait-user-input/>
31. Evaluating Global Workspace Markers in Contemporary LLM Systems - Preprints.org, fecha de acceso: enero 29, 2026, <https://www.preprints.org/manuscript/202601.1683/v1/download>
32. Global Workspace Theory (GNWT) - Emergent Mind, fecha de acceso: enero 29, 2026, <https://www.emergentmind.com/topics/global-workspace-theory-gnwt>
33. (PDF) Advancing SLM Tool-Use Capability using Reinforcement Learning - ResearchGate, fecha de acceso: enero 29, 2026, <https://www.researchgate.net/publication/395339583_Advancing_SLM_Tool-Use_Capability_using_Reinforcement_Learning>
34. Can Compressed LLMs Truly Act? An Empirical Evaluation of Agentic Capabilities in LLM Compression - arXiv, fecha de acceso: enero 29, 2026, <https://arxiv.org/html/2505.19433v1>