Informe de Viabilidad y Plan de Implementación: Una Arquitectura para la Inteligencia Artificial Distribuida

Sección 1: Resumen Ejecutivo

Declaración de Tesis

El concepto de una Inteligencia Artificial (IA) distribuida, que aprovecha la computación inactiva de los usuarios, no solo es viable, sino que representa la próxima evolución lógica de la computación en la nube, fusionando el modelo de computación voluntaria con los incentivos económicos de las Redes de Infraestructura Física Descentralizada (DePIN). Este informe presenta una arquitectura integral para construir dicha red, enfocada en la **inferencia de modelos de Mezcla de Expertos (MoE)**, un enfoque que resuelve los desafíos clave de ejecutar modelos de IA a gran escala en hardware de consumo heterogéneo.

Síntesis de la Propuesta

Proponemos un sistema híbrido que combina la robusta arquitectura cliente-servidor de proyectos como BOINC para la orquestación y validación de tareas con una red peer-to-peer (P2P) basada en libp2p para una comunicación de baja latencia entre nodos.(1), (2), (3) El núcleo de la IA será un modelo de lenguaje grande (LLM) estructurado como una Mezcla de Expertos (MoE), donde un servidor central (o un núcleo de validadores de la DAO) actúa como la "red de enrutamiento" (*gating network*), y los nodos de los usuarios ejecutan los "expertos" individuales, que son subredes neuronales más pequeñas y manejables.(4), (5) Este diseño transforma el problema de ejecutar un modelo masivo en un solo dispositivo en la

tarea, mucho más factible, de ejecutar fragmentos más pequeños del modelo en muchos dispositivos y agregar los resultados.

Modelo Económico y de Gobernanza

La participación se incentivará a través de un token de doble propósito (utilidad y gobernanza), creando un volante económico sostenible donde el uso de la red por parte de los consumidores de IA financia las recompensas para los proveedores de cómputo.(6) Los usuarios pagarán por los servicios de inferencia, y una parte de estos ingresos se utilizará para recompensar a los proveedores de cómputo y para quemar tokens, creando una presión deflacionaria que vincula el valor del token al uso real de la red.(7), (8) La gobernanza se transferirá progresivamente a una Organización Autónoma Descentralizada (DAO), asegurando la alineación a largo plazo con los intereses de la comunidad y la resiliencia del protocolo.(9), (10)

Desafíos y Mitigaciones

Se abordarán los principales desafíos: lograr una baja latencia (a diferencia de los sistemas tradicionales de computación voluntaria de alto rendimiento), garantizar la privacidad y la seguridad en un entorno sin confianza (*trustless*), y navegar el complejo panorama legal y regulatorio. La baja latencia se logrará mediante la selección inteligente de nodos y la comunicación P2P directa.(11), (12) La privacidad se abordará mediante el Aprendizaje Federado para las actualizaciones del modelo y la oferta de servicios premium de Computación Segura Multipartita (SMPC) para inferencias sensibles.(13), (14) El riesgo legal, particularmente la responsabilidad personal de los miembros, se mitigará estableciendo una entidad legal formal, como una DAO LLC, en una jurisdicción favorable.(15), (16)

Conclusión Estratégica

Este modelo tiene el potencial de democratizar radicalmente el acceso a la computación de IA, ofreciendo una alternativa resistente a la censura y de costo significativamente menor a los proveedores de nube centralizados. Al alinear los incentivos económicos con la provisión de infraestructura, se puede construir una supercomputadora global, propiedad de la

comunidad, capaz de impulsar la próxima ola de innovación en inteligencia artificial.

Sección 2: Marco Conceptual: Un Análisis de la Inteligencia Artificial Distribuida

2.1. Validación de la Idea Central: Del Monopolio de la Nube a la Supercomputadora Comunitaria

La infraestructura de inteligencia artificial actual está predominantemente centralizada, dominada por un oligopolio de proveedores de servicios en la nube como Amazon Web Services, Google Cloud y Microsoft Azure.(17), (18) Esta centralización crea barreras significativas de costo y acceso, al tiempo que concentra el control sobre una tecnología transformadora. La creciente demanda de cómputo para el entrenamiento y la inferencia de modelos de IA, especialmente los LLM, está comenzando a superar la oferta de los centros de datos centralizados, lo que resulta en escasez de GPUs y precios elevados.(18), (19)

Esta tensión de mercado crea una oportunidad estratégica para modelos alternativos. La idea de una IA distribuida se basa en una observación fundamental: existe una vasta reserva de poder computacional latente en los miles de millones de dispositivos de consumo (PCs, consolas de juegos, e incluso teléfonos inteligentes) en todo el mundo.(20), (21) Estos dispositivos a menudo permanecen inactivos durante gran parte del día. Proyectos como Folding@home han demostrado que la potencia de procesamiento combinada de estas máquinas voluntarias puede superar con creces a las supercomputadoras más potentes del mundo.(20), (22) La IA distribuida busca transformar este pasivo global (tiempo de inactividad de la CPU/GPU) en un activo productivo, creando una supercomputadora comunitaria para la inteligencia artificial.

2.2. Precedentes Históricos y Lecciones Aprendidas: El Legado de la Computación Voluntaria

La idea de aprovechar el cómputo distribuido no es nueva. Durante más de dos décadas, proyectos de computación voluntaria han sentado las bases técnicas y sociales para este

modelo.

Análisis de BOINC, SETI@home y Folding@home

El **Berkeley Open Infrastructure for Network Computing (BOINC)** es una plataforma de software de código abierto que ha impulsado algunos de los proyectos de computación distribuida más grandes, incluyendo SETI@home (Búsqueda de Inteligencia Extraterrestre) y Folding@home (simulación de plegamiento de proteínas para la investigación de enfermedades).(1), (20), (21) La arquitectura de BOINC, un modelo cliente-servidor, es un precedente directo y fundamental para la red de IA propuesta.(1), (2)

En este modelo, un servidor central distribuye "unidades de trabajo" (*work units*)—fragmentos de datos y el código para procesarlos—a un software cliente que se ejecuta en los ordenadores de los voluntarios, conocidos como "nodos trabajadores" (*worker nodes*).(2), (23) Una de las fortalezas clave de BOINC es su capacidad para gestionar una red masiva de nodos que son:

- **Heterogéneos:** Con diferentes tipos de procesadores (CPU, GPU), arquitecturas (x86, ARM) y sistemas operativos (Windows, Mac, Linux, Android).(1), (24)
- **Esporádicamente disponibles:** Los nodos pueden conectarse y desconectarse de la red en cualquier momento.(1)
- **No confiables (*Untrusted*):** Los nodos pueden devolver resultados incorrectos, ya sea por fallos de hardware o por comportamiento malicioso.(1)

Mecanismos de Validación

Para abordar el problema de los nodos no confiables, BOINC implementa un robusto mecanismo de **validación por redundancia**. Cada unidad de trabajo se envía a múltiples clientes independientes. Cuando los resultados son devueltos, el demonio validator en el servidor los compara. Se requiere un "quórum" de resultados coincidentes (generalmente dos o tres) para que un resultado sea considerado válido. Si los resultados no coinciden o no se devuelven antes de la fecha límite, el servidor genera instancias adicionales de la tarea y las envía a otros nodos hasta que se alcanza el consenso.(2), (23) Este método, aunque no es adecuado para la baja latencia, establece un principio crucial: la confianza en un sistema distribuido se logra a través de la redundancia y la verificación, no asumiendo la honestidad de los nodos individuales.

Modelos de Incentivos No Económicos

Quizás la lección más importante de estos proyectos pioneros es su éxito en la movilización de millones de voluntarios sin ofrecer compensación financiera. Su modelo de incentivos se basó en factores intrínsecos y sociales:

- **Contribución a un bien mayor:** La "pequeña pero cautivadora posibilidad de que tu ordenador detecte la débil señal de una civilización más allá de la Tierra" fue un poderoso motivador para los participantes de SETI@home.(25) De manera similar, los voluntarios de Folding@home contribuyen directamente a la investigación de enfermedades como el Alzheimer, el cáncer y, más recientemente, el COVID-19.(21), (26)
- **Gamificación y Competencia:** Las plataformas incorporaron sistemas de "crédito" que cuantificaban la contribución computacional de cada usuario. Se publicaban tablas de clasificación para individuos y equipos, fomentando una sana competencia y un compromiso a largo plazo.(21), (25)

Estos proyectos demuestran que una misión convincente y una comunidad bien estructurada pueden ser incentivos tan poderosos como el dinero, una lección vital para la fase de arranque de cualquier proyecto descentralizado.

2.3. El Paradigma Moderno: Contextualización en el Ecosistema DePIN

Mientras que la computación voluntaria se basaba en el altruismo, el paradigma moderno de las **Redes de Infraestructura Física Descentralizada (DePIN)** se basa en incentivos económicos explícitos. DePIN se refiere a redes blockchain que utilizan tokens para incentivar a las personas a construir y operar infraestructura física en el mundo real, desde redes inalámbricas y de almacenamiento hasta redes de cómputo.(6), (27)

El modelo DePIN opera en un ciclo virtuoso conocido como el "volante económico" (*flywheel*). Inicialmente, el protocolo emite tokens para recompensar a los proveedores del lado de la oferta (por ejemplo, personas que instalan un punto de acceso inalámbrico o conectan su GPU a la red). Este incentivo arranca el crecimiento de la red. A medida que la red crece en capacidad y cobertura, se vuelve más útil y atrae a usuarios del lado de la demanda que pagan por el servicio. Los ingresos generados por la demanda se utilizan para recomprar y/o quemar tokens, o para financiar las recompensas de los proveedores, creando un modelo económico sostenible que depende cada vez menos de las emisiones inflacionarias.(6)

La IA distribuida propuesta encaja perfectamente en el sector de **cómputo DePIN**, un

espacio que ya cuenta con proyectos establecidos y una capitalización de mercado significativa.(28), (29) Un análisis de los actores clave revela diferentes enfoques:

- **Akash Network (AKT):** Funciona como un "supermercado en la nube" descentralizado y de código abierto, un mercado P2P donde los proveedores pueden ofrecer su capacidad de cómputo (CPU y GPU) y los usuarios pueden arrendarla. Utiliza un sistema de "subasta inversa" para reducir los costos.(30), (31), (32)
- **Render Network (RNDR):** Se especializa en el renderizado de gráficos 3D y efectos visuales, conectando a artistas y estudios que necesitan potencia de GPU con una red de operadores de nodos que ofrecen su hardware inactivo a cambio de tokens RNDR.(32), (33), (34)
- **Bittensor (TAO):** Va un paso más allá del simple cómputo. Es un mercado descentralizado para la inteligencia misma. Los "mineros" no solo aportan cómputo, sino que entrenan y sirven modelos de IA en varias "subredes". Los "validadores" evalúan la calidad de las respuestas de estos modelos, y el sistema recompensa a los que producen el valor informativo más alto, creando una competencia que impulsa la innovación en IA.(35), (36), (37)

La propuesta actual se diferencia al centrarse específicamente en la **inferencia de modelos MoE a gran escala y baja latencia**, un nicho que aprovecha las fortalezas del cómputo distribuido de una manera que los modelos de mercado más generalistas no abordan directamente.

2.4. Propuestas de Valor Clave

La arquitectura propuesta ofrece un conjunto de ventajas disruptivas sobre el modelo centralizado actual:

- **Reducción Radical de Costos:** Al eliminar intermediarios y utilizar recursos de hardware ya existentes y pagados, los costos de inferencia de IA pueden disminuir drásticamente. Proyectos como Akash afirman lograr ahorros de hasta un 85% en comparación con los proveedores de nube tradicionales, una cifra que sirve como un punto de referencia alcanzable.(31), (38)
- **Escalabilidad Masiva:** La red puede crecer orgánicamente a medida que más usuarios se unen. No está limitada por la capacidad de construcción de centros de datos de una sola empresa. Como demostraron BOINC y Folding@home, una red de voluntarios puede agregar una capacidad de cómputo que rivaliza y supera a las supercomputadoras más grandes del mundo.(20), (22), (25)
- **Resistencia a la Censura y Soberanía de Datos:** Una red descentralizada, gobernada por su comunidad global, es inherentemente resistente a la censura por parte de una sola entidad corporativa o gubernamental.(30), (31) Permite la ejecución de modelos de

IA de código abierto sin las restricciones o filtros impuestos por los proveedores centralizados, lo cual es crucial para la investigación sin restricciones y la libertad de expresión.(39), (40)

- **Acceso Democratizado:** Al reducir drásticamente los costos, la plataforma elimina las barreras económicas que actualmente impiden a muchos desarrolladores, investigadores y startups experimentar y construir con modelos de IA de última generación. Esto fomenta un ecosistema de innovación más equitativo y vibrante.(41), (42)

La convergencia de estos precedentes y paradigmas revela una verdad fundamental: la evolución de la computación distribuida ha pasado de un modelo impulsado por el altruismo y la gamificación a uno sostenido por incentivos económicos directos. Los proyectos pioneros como SETI@home demostraron la viabilidad técnica de agregar recursos voluntarios a una escala masiva, impulsados por la motivación intrínseca de contribuir a la ciencia.(23), (25) Sin embargo, este modelo voluntario a menudo resulta en una oferta de cómputo fluctuante, inadecuada para aplicaciones comerciales que requieren fiabilidad y baja latencia. La ola moderna de DePIN ha introducido un modelo económico explícito a través de tokens, transformando la contribución de un acto voluntario a una actividad económica.(6), (27) Esto atrae a un conjunto más amplio de participantes y estabiliza el suministro de infraestructura. Un proyecto exitoso en el panorama actual debe, por lo tanto, fusionar ambos enfoques. Debe comenzar con una misión fuerte y de código abierto para atraer a una comunidad inicial de entusiastas ideológicos, para luego cimentar su crecimiento a largo plazo con una tokenómica robusta que recompense de manera justa y sostenible a los proveedores de infraestructura.

Sección 3: Plan Arquitectónico para una Red de IA Distribuida

Para materializar la visión de una IA distribuida, se requiere una arquitectura robusta que equilibre la eficiencia, la escalabilidad y la seguridad. El diseño propuesto es un modelo híbrido que combina una capa de orquestación centralizada, inspirada en BOINC, con una capa de comunicación descentralizada P2P para lograr baja latencia.

3.1. Infraestructura Central: Un Modelo Híbrido

Capa de Orquestación (Basada en BOINC)

El núcleo de la gestión de la red residirá en un servidor central (o, en una fase posterior, en un conjunto de nodos validadores gobernados por la DAO). Este servidor no realizará el cómputo pesado, sino que actuará como el director de orquesta, coordinando a los miles de nodos trabajadores. Sus componentes principales, análogos a los demonios de BOINC, son (2):

- **Scheduler (Planificador):** Este es el punto de contacto principal para los clientes. Recibe las solicitudes de trabajo y asigna las unidades de trabajo de IA. A diferencia del planificador estándar de BOINC, que utiliza un exponential backoff para minimizar la carga del servidor, este componente será modificado para priorizar la baja latencia. Implementará directivas de reconexión rápida, como la etiqueta <next_rpc_delay>, que instruye al cliente a volver a contactar en un corto período de tiempo (por ejemplo, segundos en lugar de horas), asegurando un flujo constante de tareas para aplicaciones en tiempo real.(12)
- **Feeder (Alimentador):** Un proceso de backend que precarga las tareas desde la base de datos a un bloque de memoria compartida. Esto permite que el planificador acceda y asigne tareas con una latencia mínima, sin tener que consultar la base de datos en cada solicitud.(2)
- **Validator (Validador):** Este componente es crucial para la confianza en la red. Recibe los resultados de las tareas completadas por múltiples nodos y los compara. Si se alcanza un quórum de resultados idénticos, la tarea se marca como válida, y los nodos que proporcionaron la respuesta correcta ven su reputación aumentada. Los resultados no coincidentes son descartados.(2)
- **Transitioner (Transicionador):** Gestiona el ciclo de vida de las tareas. Si una tarea no es devuelta antes de su fecha límite o si sus resultados son invalidados, el transicionador la vuelve a poner en la cola para ser asignada a otros nodos.(2)

Software del Cliente (Nodo Trabajador): Un Agente Ligero y Multiplataforma

El software que los usuarios instalarán en sus dispositivos debe ser ligero, seguro y no intrusivo.

- **Núcleo del Cliente:** Un proceso en segundo plano (similar a boinc.exe o boincd) que se comunica con el servidor de orquestación. Gestiona la descarga de tareas, la ejecución de los motores de IA y la carga de resultados. Se configurará para utilizar recursos de CPU y GPU solo cuando el dispositivo esté inactivo o a una prioridad muy baja, para no interferir con el uso normal del ordenador por parte del usuario.(1), (23), (43)
- **Motor de Ejecución de IA:** El cliente no contendrá un modelo de IA monolítico. En su lugar, integrará motores de inferencia estandarizados y altamente optimizados, capaces

de ejecutar los fragmentos de modelo (los "expertos") que el servidor le envíe. Las dos tecnologías clave para esto son:

- **ONNX Runtime:** Un motor de inferencia multiplataforma de alto rendimiento de Microsoft. Su principal ventaja es el uso de "Execution Providers" (EPs), que son backends que optimizan la ejecución en hardware específico. Esto permite que el mismo modelo ONNX se ejecute de manera óptima en GPUs NVIDIA (a través del EP de CUDA), en hardware de Apple (a través de CoreML), en CPUs Intel (a través de OpenVINO), y en una amplia gama de otros dispositivos.(44), (45), (46)
- **llama.cpp:** Un proyecto de código abierto que ha revolucionado la inferencia de LLMs en hardware de consumo. Escrito en C/C++, ofrece un rendimiento excepcional en una amplia variedad de CPUs (x86, ARM) y GPUs (a través de backends como Metal, CUDA, Vulkan, OpenCL). Su formato de archivo nativo, GGUF, está diseñado para cargar y ejecutar eficientemente modelos cuantizados (modelos cuyos pesos se han reducido en precisión para disminuir el uso de memoria y acelerar el cálculo), lo cual es ideal para este caso de uso.(47), (48)

Capa de Comunicación (Basada en Web3 P2P)

Para las operaciones que requieren baja latencia y comunicación directa entre nodos, se superpondrá una red P2P a la arquitectura cliente-servidor.

- **Protocolo libp2p:** Es el estándar de oro para la comunicación P2P en el ecosistema Web3, utilizado por proyectos como IPFS y Ethereum. Es una pila de red modular que proporciona funcionalidades esenciales como el descubrimiento de pares (encontrar otros nodos en la red), el enrutamiento de mensajes y, de manera crucial, técnicas de **NAT traversal** (como Hole Punching y AutoRelay). Esto permite que dos nodos se conecten directamente incluso si ambos están detrás de firewalls o routers domésticos, un requisito indispensable para una red P2P verdaderamente descentralizada.(3), (49)
- **Enrutamiento Geográfico y de Baja Latencia:** Para minimizar el tiempo de viaje de los datos, el sistema implementará estrategias de enrutamiento basadas en la ubicación. Cuando se necesita una respuesta rápida, en lugar de seleccionar un nodo al azar, el planificador o los propios nodos darán prioridad a los pares que estén geográficamente más cerca.(11), (50), (51) La red también puede mantener un conjunto de "nodos de retransmisión" (*relay nodes*) de alto ancho de banda en puntos estratégicos de la red global de Internet para acelerar la propagación de mensajes críticos, una técnica probada en redes blockchain de alto rendimiento.(52)

3.2. Gestión de Tareas y Flujo de Trabajo

- **Definición de "Unidades de Trabajo de IA":** Una "unidad de trabajo" no será una tarea de inferencia completa, sino una micro-tarea atómica. En el contexto de un modelo MoE, una unidad de trabajo podría consistir en: la entrada (un lote de *tokens*), el identificador del "experto" a ejecutar y los metadatos necesarios. Esto descompone una gran tarea en miles de pequeñas tareas paralelas.
- **Manejo de la Heterogeneidad y Disponibilidad:** El servidor mantendrá un perfil detallado de cada nodo trabajador, incluyendo sus especificaciones de hardware (CPU, GPU, VRAM), benchmarks de rendimiento, velocidad de red y un puntaje de reputación. Las tareas se asignarán a los nodos que cumplan con los requisitos mínimos y que sean más adecuados para la tarea. Para gestionar la naturaleza intermitente de los nodos voluntarios, cada unidad de trabajo tendrá una fecha límite estricta (por ejemplo, unos pocos minutos). Si un resultado no se devuelve a tiempo, el Transitioner lo reasignará inmediatamente a otro nodo disponible.(21)
- **Sistema de Reputación y Confianza:** La confianza no se asume, se construye. Se implementará un sistema de reputación dinámico que califique a cada nodo basándose en métricas objetivas y verificables:
    - **Corrección:** El historial de envío de resultados que son validados positivamente por el sistema de quórum.
    - **Puntualidad:** La frecuencia con la que las tareas se completan antes de su fecha límite.(21), (43)
    - **Rendimiento:** La velocidad de cómputo medida, que permite al planificador estimar con precisión los tiempos de finalización.
    - Disponibilidad (Uptime): El tiempo que el nodo permanece conectado y activo en la red.
      El planificador utilizará este puntaje de reputación como un factor clave en la asignación de tareas, priorizando a los nodos más fiables y de mayor rendimiento.(53), (54)

La elección de una arquitectura híbrida no es un compromiso, sino una optimización estratégica. Un sistema puramente P2P para la asignación de tareas de IA a escala global enfrenta desafíos de coordinación casi insuperables. Cuestiones como el equilibrio de carga óptimo, la prevención de la colusión maliciosa y la selección eficiente de nodos son extremadamente difíciles de resolver sin una visión global del estado de la red, algo que una autoridad centralizada o un conjunto de validadores de confianza puede proporcionar de manera eficiente.(55) Por otro lado, un sistema puramente centralizado como el modelo BOINC clásico, aunque escalable, está intrínsecamente limitado por la latencia, ya que cada interacción debe pasar por el servidor.(12) Al combinar ambos modelos, se obtiene una sinergia poderosa: el servidor central se encarga de las tareas de "alta confianza" que requieren una visión global, como la validación de resultados, la gestión de la economía del token y el enrutamiento inicial de las consultas. Una vez que el servidor ha asignado una subtarea a un grupo de nodos, estos pueden utilizar la red P2P libp2p para comunicarse directamente entre sí, coordinando la ejecución y agregando resultados intermedios sin el

cuello de botella del servidor. Esta arquitectura híbrida se alinea perfectamente con el modelo de IA de Mezcla de Expertos: el servidor actúa como la red de enrutamiento global, y los nodos P2P funcionan como los expertos que realizan el trabajo computacional localizado de manera eficiente y con baja latencia.

Sección 4: Diseño de la IA: Modelos y Metodologías para un Mundo Descentralizado

La elección de la arquitectura del modelo de IA y la metodología de operación es tan crucial como el diseño de la infraestructura de red. El sistema debe estar diseñado desde cero para prosperar en un entorno de computación distribuida, heterogénea y de baja confianza.

4.1. El Foco Estratégico: Priorizando la Inferencia sobre el Entrenamiento

El primer y más importante principio de diseño es centrarse en la **inferencia** de IA, no en el entrenamiento. El entrenamiento distribuido de modelos de lenguaje grandes (LLMs) a través de Internet en hardware de consumo es, con la tecnología actual, prácticamente inviable por varias razones fundamentales:

- **Requisitos de Ancho de Banda y Latencia:** El entrenamiento distribuido, especialmente para modelos de transformadores, requiere una comunicación constante y de muy alta velocidad entre los nodos que procesan diferentes partes del modelo o de los datos. Técnicas como el paralelismo de datos y de tensores dependen de interconexiones de red de latencia ultra baja y ancho de banda masivo (cientos de GB/s), como NVLink de NVIDIA o redes RoCE especializadas en centros de datos.(19), (56), (57) Las conexiones a Internet domésticas, con su alta latencia y ancho de banda asimétrico, son completamente inadecuadas para esta tarea.
- **Dependencia del Estado y Sincronización:** El proceso de entrenamiento es secuencial y dependiente del estado. Cada paso de optimización (por ejemplo, descenso de gradiente) depende del resultado del paso anterior. La naturaleza esporádica de los nodos voluntarios, que pueden desconectarse en cualquier momento, haría que el proceso de entrenamiento se detuviera y reiniciara constantemente, haciéndolo ineficiente hasta el punto de ser inútil.

Por el contrario, la **inferencia** (el proceso de usar un modelo ya entrenado para hacer

predicciones) es un caso de uso ideal para la computación distribuida:

- **Altamente Paralelizable:** Cada solicitud de inferencia es independiente de las demás. Se pueden procesar miles de solicitudes en paralelo en miles de nodos diferentes sin necesidad de comunicación entre ellos. Es lo que se conoce como un problema "vergonzosamente paralelo" (*embarrassingly parallel*).(58), (59)
- **Sin Estado:** La inferencia no modifica los pesos del modelo. Si un nodo falla a mitad de una tarea, esta puede ser simplemente reasignada a otro nodo sin pérdida de progreso.
- **Requisitos de Comunicación Menores:** La comunicación se limita a enviar la entrada (el *prompt*) y recibir la salida (la respuesta generada).
- **Viabilidad en Hardware de Consumo:** Las GPUs de consumo modernas, aunque no son adecuadas para el entrenamiento a gran escala, son extremadamente capaces de realizar inferencia de manera eficiente, especialmente con modelos cuantizados.(17)

4.2. Arquitectura de Mezcla de Expertos (MoE): La Clave para Distribuir Modelos Gigantes

El principal obstáculo para la inferencia de LLMs en dispositivos de consumo es el tamaño del modelo, que a menudo excede la VRAM disponible. La arquitectura de **Mezcla de Expertos (MoE)** es la solución a este problema, y es el pilar central de la estrategia de IA propuesta.(4), (5), (60)

Un modelo MoE no es una red neuronal densa y monolítica. En su lugar, reemplaza algunas de sus capas (típicamente las capas de *feed-forward*) con un bloque MoE. Este bloque contiene dos componentes (5):

1. Un conjunto de **"expertos"**: Son redes neuronales más pequeñas e idénticas en arquitectura.
2. Una **"red de enrutamiento" (*gating network*)**: Una pequeña red neuronal que aprende a dirigir cada token de la secuencia de entrada al experto (o a un pequeño número de expertos, típicamente dos) que está mejor calificado para procesarlo.

Esto significa que para cada token, solo una pequeña fracción de los parámetros totales del modelo se activa y se utiliza para el cálculo. Por ejemplo, el modelo Mixtral 8x7B tiene un total de 46.7 mil millones de parámetros, pero para cada token, solo se activan dos de sus ocho expertos, utilizando aproximadamente 12.9 mil millones de parámetros por inferencia.(4) Este enfoque permite que el modelo tenga una capacidad y un conocimiento enormes (almacenados en el total de sus parámetros) mientras mantiene un costo computacional por inferencia relativamente bajo.

Esta arquitectura se mapea perfectamente a nuestra red distribuida:

1. El **servidor central** alojará las partes más ligeras y críticas del modelo: la red de enrutamiento y las capas compartidas (como las capas de atención).
2. Los **nodos de los usuarios** descargarán y alojarán uno o más de los "expertos". Dado que cada experto es un modelo mucho más pequeño, puede caber fácilmente en la VRAM de una GPU de consumo.
3. El flujo de inferencia sería el siguiente: un usuario envía una consulta al servidor. El servidor procesa la entrada a través de la red de enrutamiento, que determina la secuencia de expertos necesarios para generar la respuesta. El servidor luego descompone la tarea y envía micro-trabajos a los nodos de la red que alojan a los expertos requeridos. Los nodos ejecutan sus expertos y devuelven los resultados, que el servidor ensambla para formar la respuesta final.

Este enfoque es el habilitador fundamental que hace que la ejecución de un modelo de IA de vanguardia en una red de hardware de consumo sea factible. Transforma un problema intratable ("ejecutar un modelo de 100B de parámetros en un PC") en un problema logístico manejable ("ejecutar un modelo de 7B de parámetros en 16 PCs diferentes y coordinar los resultados").

4.3. Aprendizaje Federado (FL): Actualizaciones y Personalización con Preservación de la Privacidad

Para que el modelo de IA evolucione, mejore y se adapte con el tiempo, es necesario reentrenarlo con nuevos datos. Recopilar los datos de los usuarios en un servidor central plantearía enormes problemas de privacidad. El **Aprendizaje Federado (FL)** es la solución a este dilema.(13), (61)

En un sistema de FL, el modelo global se envía a los dispositivos de los usuarios. El modelo se entrena localmente en cada dispositivo, utilizando los datos locales del usuario, que nunca abandonan el dispositivo. Después del entrenamiento local, solo las actualizaciones de los pesos del modelo (los gradientes), no los datos en sí, se envían de vuelta al servidor. El servidor agrega estas actualizaciones de forma segura (a menudo utilizando técnicas criptográficas adicionales para garantizar el anonimato) para crear una nueva versión mejorada del modelo global.(13), (62)

En nuestro sistema, el FL se utilizará para:

- **Mejorar el Modelo Global:** Periódicamente, se pueden realizar rondas de FL para reentrenar tanto la red de enrutamiento como los expertos, mejorando el rendimiento general del sistema.
- **Crear Expertos Especializados:** El FL es particularmente efectivo para manejar datos heterogéneos y no independientemente e idénticamente distribuidos (no-IID), que es

exactamente la situación en una red de dispositivos de usuarios diversos.(61), (63) Esto podría permitir la creación de expertos especializados en ciertos dominios o tipos de datos que emergen orgánicamente de la comunidad de usuarios.

4.4. Optimización para Baja Latencia: Adaptando el Modelo de Alto Rendimiento

El objetivo principal de la inferencia de IA es una respuesta rápida. Por lo tanto, la arquitectura debe estar optimizada para la baja latencia, un objetivo a menudo en conflicto con la optimización para el alto rendimiento de los sistemas de computación voluntaria tradicionales.

- **Selección de Nodos Inteligente:** El planificador del servidor debe ser sofisticado. Al recibir una solicitud de inferencia, no solo debe considerar la capacidad computacional de los nodos, sino también su latencia de red con respecto al servidor y, potencialmente, al usuario final. Se dará prioridad a los nodos de alta reputación que estén geográficamente más cerca para minimizar el tiempo de ida y vuelta de los datos.(11), (50)
- **Ejecución Especulativa:** Para las tareas más críticas dentro de una cadena de inferencia, el planificador puede enviar la misma micro-tarea a varios nodos de alta calidad simultáneamente. El sistema utilizará el primer resultado correcto que llegue, cancelando las tareas duplicadas. Esto aumenta la redundancia computacional pero reduce la latencia al mitigar el riesgo de depender de un solo nodo que podría ser lento.
- **Gestión Agresiva de "Rezagados" (*Stragglers*):** En un sistema de baja latencia, los nodos lentos son un problema importante. Se establecerán tiempos de espera (*timeouts*) muy cortos para cada micro-tarea. Si un nodo no responde dentro de este umbral (por ejemplo, unos pocos cientos de milisegundos), la tarea se reasignará instantáneamente a otro nodo que esté en espera. El sistema de reputación penalizará fuertemente a los nodos que son consistentemente lentos o que no completan las tareas, asegurando que el grupo de nodos activos sea de alta calidad.

Sección 5: El Motor Económico: Un Modelo de Tokenomics Sostenible

Para pasar de un modelo de computación voluntaria basado en el altruismo a un ecosistema de infraestructura de grado comercial, es indispensable un motor económico robusto. Este motor se basará en un token nativo diseñado para incentivar la participación, facilitar las

transacciones y alinear los intereses de todos los participantes de la red a largo plazo.

5.1. Incentivando la Participación a Escala

El principio fundamental es que los recursos computacionales no son gratuitos. Los usuarios incurren en costos de electricidad y desgaste de hardware. Para atraer y retener un suministro de cómputo a gran escala, confiable y de alto rendimiento, la red debe compensar a los proveedores de manera justa y predecible.(64), (65), (66) Un sistema de recompensas cripto-económicas, pagado en el token nativo de la red, convierte la contribución de recursos de una actividad voluntaria a una oportunidad económica, atrayendo a un espectro más amplio de participantes, desde entusiastas individuales hasta operadores profesionales de hardware.

5.2. Diseño del Token: Un Activo de Doble Propósito (Utilidad y Gobernanza)

Se propone la creación de un token nativo, denominado provisionalmente NEURO, que cumplirá dos funciones críticas dentro del ecosistema.

Funciones de Utilidad

La utilidad del token es lo que genera una demanda orgánica y sostenible, más allá de la mera especulación.

- **Medio de Pago:** Los desarrolladores, empresas y usuarios finales que consuman los servicios de inferencia de la IA pagarán por ellos utilizando NEURO. Para simplificar la experiencia del usuario, la plataforma puede permitir pagos en stablecoins (como USDC) o incluso en moneda fiduciaria, que se convertirían automáticamente a NEURO en el backend para liquidar las transacciones en la red. Esto asegura que toda la actividad económica fluya a través del token nativo.(34), (65), (67)
- **Staking para Operadores de Nodos:** Para participar como proveedor de cómputo, un usuario deberá hacer *staking*, es decir, bloquear una cantidad mínima de NEURO como una forma de garantía o fianza. Este mecanismo cumple varias funciones: (1) actúa como un disuasivo contra el comportamiento malicioso (un nodo que intente engañar al

sistema puede ver su *stake* "recortado" o *slashed*); (2) alinea los incentivos de los proveedores con la salud a largo plazo de la red, ya que tienen un interés económico directo en su éxito; y (3) reduce la oferta circulante del token, lo que puede tener un impacto positivo en su valor.(35), (36)
- **Tasas de Transacción:** Pequeñas tasas en NEURO se aplicarán a las operaciones de la red para prevenir ataques de spam y financiar la seguridad de la red.

Funciones de Gobernanza

A medida que la red madure, su control se transferirá progresivamente a su comunidad de usuarios a través de una DAO.

- **Derechos de Voto:** La posesión de tokens NEURO, especialmente aquellos en *staking*, otorgará a los usuarios el derecho a proponer y votar sobre Propuestas de Mejora del Protocolo (PIPs).(36), (65), (67)
- **Ámbito de la Gobernanza:** Las decisiones clave que se someterán a la votación de la DAO incluirán: la incorporación de nuevos modelos de IA a la red, ajustes en la estructura de comisiones y recompensas, la asignación de fondos de la tesorería de la DAO para subvenciones de desarrollo, marketing y otras iniciativas del ecosistema, y la elección de los miembros de los comités técnicos.

5.3. Acumulación de Valor y Sostenibilidad: El Volante Económico

Un diseño de tokenomics exitoso debe crear un sistema de circuito cerrado donde el valor se acumule en el token a medida que aumenta el uso de la red.

- **Flujo de Ingresos y Tesorería de la DAO:** Un porcentaje de todos los pagos por servicios de inferencia (por ejemplo, el 5-10%) se dirigirá automáticamente a una tesorería controlada por la DAO. Estos fondos se utilizarán para financiar el desarrollo continuo del protocolo, subvenciones para desarrolladores, programas de marketing y otras iniciativas que beneficien al ecosistema.(38)
- **Mecanismo de Quema (*Burn*):** Para vincular directamente el valor del token a la actividad de la red, una porción de las tarifas recaudadas (por ejemplo, el 20% de los ingresos de la tesorería) se "quemará" programáticamente, es decir, se eliminará de forma permanente de la circulación. Este mecanismo deflacionario significa que a medida que la red procesa más inferencias y genera más ingresos, el suministro total de NEURO disminuye, aumentando la escasez y potencialmente el valor de los tokens restantes.(8)

- **Distribución de Recompensas Sostenible:** Las recompensas para los proveedores de cómputo se financiarán mediante un modelo dual. En la fase inicial, se utilizarán emisiones inflacionarias del token (una cantidad predefinida de nuevos NEURO creados por bloque o por época) para arrancar la red y atraer a los primeros proveedores. Sin embargo, el objetivo a largo plazo es hacer la transición a un modelo en el que la mayoría de las recompensas se financien con los ingresos reales generados por la red. Esto evita el modelo insostenible de muchos proyectos que dependen únicamente de una alta inflación para pagar las recompensas, lo que a menudo conduce a una presión de venta constante y a la devaluación del token.(7), (8)
- **Prevención de "Espirales de la Muerte":** El modelo económico debe diseñarse para ser robusto frente a condiciones de mercado extremas. Se evitarán los mecanismos de retroalimentación reflexiva que llevaron al colapso de proyectos como Terra/LUNA, donde el valor de un activo dependía algorítmicamente de otro, creando un riesgo sistémico.(68), (69) El valor de NEURO no se basará en una paridad algorítmica, sino en la demanda real de un servicio tangible: el cómputo de IA. Además, las recompensas por *staking* se mantendrán en niveles sostenibles y respaldados por los ingresos, para no repetir los errores de plataformas como Celsius, que ofrecían rendimientos insostenibles.(68)

5.4. Tabla Comparativa de Tokenomics DePIN

Para contextualizar el modelo propuesto, es útil compararlo con los enfoques de otros proyectos líderes en el sector de cómputo DePIN.

Característica	Akash Network (AKT)	Render Network (RNDR)	Bittensor (TAO)	Propuesta de IA Distribuida (NEURO)
Función Principal del Token	Utilidad (pago, staking de seguridad PoS), Gobernanza (30), (67)	Utilidad (pago por renderizado), Gobernanza (34)	Utilidad (acceso a subredes, staking), Incentivo (recompensa por inteligencia) (35), (36)	Doble Propósito: Utilidad (pago, staking de garantía) y Gobernanza (DAO)

Mecanismo de Acumulación de Valor	Tarifas de red, staking (participación en la inflación) (38)	Mecanismo de Quema y Acuñación (Burn-and-Mint Equilibrium) (37)	Emisión fija con halvings (similar a Bitcoin), recompensas basadas en el valor informativo (70)	Tarifas de uso de la red, quema de una porción de las tarifas, staking
Incentivo para Proveedores	Recompensas de bloque (inflación) y tarifas de arrendamiento pagadas por los usuarios (65), (67)	Tokens RNDR pagados por los creadores por cada trabajo de renderizado completado (33), (34)	Emisiones de TAO distribuidas a mineros y validadores según su rendimiento y consenso (35), (71)	Recompensas de bloque (fase inicial) y una porción de los ingresos por inferencia pagados por los usuarios
Gobernanza	On-chain a través de la votación de los stakers de AKT (67)	Gobernanza on-chain a través de la votación de los poseedores de RNDR (34)	Gobernanza a través de un "Triunvirato" y votación de los stakers de TAO (70)	DAO progresivamente descentralizada, controlada por los stakers de NEURO, utilizando un marco modular como Aragon

El mayor riesgo para un proyecto DePIN no es a menudo el fracaso técnico, sino el colapso económico debido a un diseño de tokenomics defectuoso. Un modelo exitoso debe ir más allá de la especulación y crear un ciclo de valor auto-reforzado. La utilidad del token debe ser indispensable para el funcionamiento de la red. En este caso, cada consulta de IA genera una tarifa tangible. Al canalizar una parte de esta tarifa hacia un mecanismo de quema o recompra, se crea un vínculo directo entre el uso de la plataforma y la escasez del token. Este mecanismo genera un bucle de