

Arquitectura de un Sistema de Inteligencia de Palabras Clave Nativo de Web3: Un Enfoque Basado en Datos On-Chain y Análisis Semántico con IA

Parte I: Un Nuevo Paradigma para la Inteligencia Digital: Más Allá de las Métricas de Web2

1.1. Introducción: El Valor de la Autenticidad

El panorama de la inteligencia de mercado digital ha estado dominado durante mucho tiempo por herramientas que analizan los rastros indirectos de la intención del usuario. Las herramientas de palabras clave de la Web2, en su núcleo, interpretan aproximaciones: consultas de búsqueda, tasas de clics, tiempo de permanencia en la página y perfiles de enlaces. Si bien son poderosos, estos métodos analizan las sombras en la pared, no los objetos que las proyectan. El proyecto aquí delineado se basa en una premisa fundamentalmente diferente y estratégicamente superior: la oportunidad de analizar acciones directas, con respaldo económico y discursos públicos registrados en un libro mayor inmutable.

Este informe presenta un plan arquitectónico para la creación de una herramienta de investigación de palabras clave y análisis de intenciones que opera directamente sobre los datos de la blockchain. El objetivo es trascender las métricas tradicionales para capturar una señal de mayor fidelidad: la "Intención On-Chain". Una transacción en la blockchain, ya sea una compra de un NFT, un intercambio en un exchange descentralizado (DEX) o una interacción con un protocolo de gobernanza, representa un compromiso y una expresión de interés mucho más definitivos que una simple búsqueda en la web. No es una consulta sobre

una acción potencial; es la evidencia registrada de la acción misma.

La distinción entre los navegadores de Web2 y Web3 es fundamental para este nuevo paradigma. Los navegadores de Web2 operan sobre la base de historiales privados y basados en sesiones, cuyos datos agregados y anonimizados son el producto. En contraste, las interacciones a través de navegadores habilitados para Web3, como las billeteras de navegador, generan un rastro de actividad público, persistente y seudónimo en la blockchain. Este cambio transforma la naturaleza del análisis del comportamiento del usuario de la inferencia a la observación directa. En lugar de adivinar lo que un usuario podría querer basándose en sus búsquedas, es posible observar lo que valora basándose en dónde asigna sus activos digitales, a qué comunidades se une y en qué conversaciones participa. El sistema propuesto está diseñado para aprovechar esta transparencia radical, construyendo una comprensión del mercado basada en la verdad fundamental registrada en la cadena de bloques.

1.2. El Imperativo Estratégico de la Soberanía de los Datos

La decisión explícita de evitar las interfaces de programación de aplicaciones (API) de terceros, como las ofrecidas por Alchemy o Moralis, no es una mera restricción técnica, sino una ventaja estratégica central. Estos servicios de infraestructura han sido fundamentales para el crecimiento del ecosistema Web3, proporcionando un acceso conveniente a datos de blockchain preprocesados e indexados.¹ Ofrecen abstracciones que aceleran el desarrollo al simplificar la compleja tarea de consultar datos de nodos.³ Sin embargo, esta conveniencia tiene un costo oculto: la dependencia de una capa de interpretación.

Los agregadores de datos, por su propia naturaleza, toman decisiones editoriales. Filtran, estructuran y priorizan los datos según sus modelos de negocio y limitaciones técnicas. Esto puede llevar a la abstracción de datos, la limitación de velocidad o el filtrado que pueden oscurecer la "verdad fundamental" de la blockchain. Al construir una canalización de datos propietaria, el sistema propuesto obtiene un acceso sin filtros al libro mayor. Esto permite el desarrollo de modelos analíticos únicos que los competidores que dependen de estos servicios no pueden replicar.

La verdadera ventaja competitiva de este enfoque reside en la creación de un "foso de datos" defendible. Los métodos propietarios desarrollados para analizar, estructurar y analizar datos brutos on-chain se convierten en la propiedad intelectual central de la herramienta. Mientras que los competidores ven el mercado a través de la lente estandarizada proporcionada por los agregadores, este sistema desarrolla su propia lente única y de mayor resolución. Puede detectar patrones, anomalías o correlaciones que son invisibles a través de las API estandarizadas. Por lo tanto, la independencia técnica se traduce directamente en una

ventaja informativa. La capacidad de analizar el flujo de datos sin procesar, en su totalidad, es la base para generar una inteligencia de mercado verdaderamente diferenciada y de alto valor, un "alfa" que no puede obtenerse de fuentes pre-digeridas.

Parte II: El Crisol de Datos On-Chain: Arquitectura de la Capa de Extracción

2.1. La Base: Interacción Directa con Nodos e Indexadores

La base de la soberanía de los datos de este sistema es la interacción directa con la infraestructura de la blockchain. Utilizando los indexadores existentes, la capa de extracción se conectará directamente a los nodos de la blockchain para obtener un flujo de datos sin procesar y en tiempo real. Este proceso implica varias operaciones fundamentales:

1. **Obtención de Bloques:** El sistema consultará continuamente a los nodos para obtener los bloques más recientes. Cada bloque sirve como un contenedor de transacciones y cambios de estado.
2. **Análisis de Transacciones:** Dentro de cada bloque, el sistema analizará la lista de transacciones. Cada transacción contiene información crucial como las direcciones de origen y destino, el valor transferido y un campo de datos de calldata.
3. **Decodificación de Registros de Eventos (Logs):** La fuente de datos más rica para el análisis de la actividad de los contratos inteligentes son los registros de eventos. Los contratos inteligentes emiten eventos cuando ocurren acciones significativas (por ejemplo, Transfer para un token, Mint para un NFT). Estos eventos son datos estructurados que se almacenan de manera más eficiente que el estado del contrato.

Para que estos datos brutos sean útiles, el sistema debe ser capaz de decodificarlos. Esto requiere el mantenimiento de un **Registro de Contratos Inteligentes**. Este registro es una base de datos interna que mapea las direcciones de los contratos inteligentes de interés (por ejemplo, el contrato de un popular juego de NFT, un protocolo DeFi o una plataforma de identidad social) a sus Interfaces Binarias de Aplicación (ABI). El ABI es una especificación en formato JSON que define las funciones y eventos de un contrato. Con el ABI, el sistema puede decodificar el calldata binario de una transacción en una llamada a una función legible por humanos (por ejemplo, `swap(address tokenIn, address tokenOut, uint256 amountIn)`) y analizar los temas y datos de los registros de eventos en información estructurada y significativa. Este registro es el mapa esencial que permite al sistema traducir los bytes

crípticos de la blockchain en las acciones y comportamientos concretos de los usuarios.

2.2. Decodificación de Activos e Identidad del Usuario: NFTs y ENS

Para construir un perfil de los intereses y afiliaciones de un usuario, el sistema debe ir más allá de las transacciones financieras y analizar los activos que señalan identidad y capital cultural. Los Tokens No Fungibles (NFTs) y el Ethereum Name Service (ENS) son las dos fuentes de datos primordiales para este propósito.

Extracción de Metadatos de NFT

Los NFTs son más que simples tokens; son certificados de propiedad de activos digitales únicos, a menudo vinculados a comunidades, arte o funcionalidades específicas.⁵ Sus metadatos (el título, la descripción, los atributos y la imagen) son una ventana directa a los intereses del propietario. Si bien muchos NFTs almacenan estos metadatos fuera de la cadena en sistemas como IPFS o Amazon S3 para reducir costos ⁶, el puntero a estos metadatos siempre se encuentra on-chain.

El sistema implementará un proceso robusto para la extracción directa de estos metadatos:

1. **Llamada a tokenURI:** Para un NFT dado (identificado por la dirección del contrato y el tokenId), el sistema realizará una llamada RPC directa a la función tokenURI del contrato estándar ERC-721 o ERC-1155. Esta función devuelve una URI que apunta a los metadatos.
2. **Resolución de URI:** El sistema debe ser capaz de manejar varios esquemas de URI:
 - **HTTP/HTTPS:** Para metadatos alojados en servidores centralizados, se realizará una simple solicitud GET.
 - **IPFS/Arweave:** Para metadatos en almacenamiento descentralizado, el sistema utilizará una puerta de enlace IPFS/Arweave o se conectará a un nodo local para resolver el hash y recuperar el archivo JSON de metadatos.
 - **On-Chain (Data URIs):** Algunos NFTs, especialmente en el ámbito del arte generativo, codifican sus metadatos directamente en la URI devuelta, a menudo como un JSON codificado en Base64. El sistema debe detectar y decodificar estos formatos.
3. **Análisis de Metadatos:** Una vez obtenido el archivo JSON, se analizará para extraer campos clave como name, description, image, y el array de attributes. Estos atributos son particularmente valiosos, ya que a menudo describen rasgos específicos que

pueden ser agregados para análisis de tendencias (por ejemplo, "¿cuántos poseedores de CryptoPunks tienen el atributo 'gorro de lana'?").

En casos donde los metadatos se almacenan completamente on-chain dentro del propio almacenamiento del contrato, se requerirán técnicas más avanzadas para leer directamente el estado del contrato, conceptualmente similares a las funciones de carga de datos de bajo nivel observadas en otras arquitecturas de blockchain.⁷

Resolución de Dominios ENS

El ENS es la capa de identidad de facto en Ethereum y las redes compatibles con EVM.⁸ Un nombre ENS como

vitalik.eth es un identificador legible por humanos que un usuario elige para asociar con su dirección. La posesión y configuración de un nombre ENS es un acto deliberado de construcción de identidad. El sistema interactuará directamente con los contratos inteligentes del Registro ENS para extraer esta información.⁹

Se pueden utilizar bibliotecas como web3.py o ethers.js, que tienen módulos incorporados para la resolución de ENS¹⁰, o se pueden realizar llamadas RPC directas a los métodos del contrato:

- **Resolución Directa (forward resolution):** Dado un nombre (por ejemplo, ens.eth), el sistema llamará al resolver en el contrato de registro para encontrar la dirección del contrato resolutor específico de ese nombre. Luego, en ese resolutor, llamará a la función `addr()` para obtener la dirección de Ethereum asociada.
- **Resolución Inversa (reverse resolution):** Dado una dirección de Ethereum, el sistema construirá un nombre especial (por ejemplo, `0x...addr.reverse`) y consultará al registro para encontrar su resolutor. Luego, en ese resolutor, llamará a la función `name()` para obtener el nombre ENS principal que el propietario ha configurado para esa dirección.

La combinación de los holdings de NFT con un nombre ENS para una sola dirección crea un "personaje de usuario" de alta fidelidad. Una dirección que posee NFTs de colecciones de arte generativo, avatares de proyectos de redes sociales descentralizadas y un nombre .eth que coincide con un perfil en otras plataformas sociales, probablemente pertenece a un nativo de Web3 culturalmente comprometido. Sus acciones y discurso on-chain tienen un contexto y un peso diferentes a los de un especulador puramente financiero. La síntesis de los datos de activos (NFTs) e identidad (ENS) proporciona el contexto necesario para interpretar correctamente las conversaciones y transacciones posteriores del usuario.

2.3. El Libro Mayor Social: Minería de Lens y Farcaster en Busca de Oro Conversacional

Para una herramienta de palabras clave, la fuente de datos más directa es la conversación pública. En Web3, Lens Protocol y Farcaster han surgido como las principales plataformas de redes sociales descentralizadas (DeSoc). La extracción de datos de estas plataformas sin depender de agregadores de terceros es fundamental para el éxito del proyecto.

Lens Protocol

Lens es un grafo social componible construido sobre la blockchain de Polygon. Su arquitectura expone una API de GraphQL completa y bien documentada, que es la interfaz principal para la extracción de datos.¹¹ El sistema se conectará directamente a un punto final de la API de Lens (público o auto-alojado) para construir un flujo de datos en tiempo real. Es crucial diferenciar esta API del protocolo social de otras tecnologías con nombres similares pero no relacionadas.¹²

La estrategia de extracción se centrará en la consulta `explorePublications`. Esta consulta permite recuperar una lista paginada de publicaciones (posts, comentarios, espejos) basada en varios criterios de filtrado y ordenación. Para construir un "firehose" completo de conversaciones, el sistema implementará una lógica para paginar continuamente a través de las publicaciones más recientes (`orderBy: LATEST`). Cada publicación recuperada contendrá el texto del contenido, metadatos, y el ID del perfil del autor. A partir del ID del perfil, se pueden realizar consultas adicionales para obtener detalles del perfil, como el nombre de usuario, la biografía y la dirección de la billetera propietaria, creando así un enlace directo entre la conversación y la identidad on-chain.

Farcaster Protocol

La arquitectura de Farcaster es más descentralizada que la de Lens. Los datos no residen en un único conjunto de contratos inteligentes, sino que se distribuyen a través de una red de servidores peer-to-peer llamados "Hubs".¹⁶ Cada Hub almacena una copia completa de todos

los datos de Farcaster y los sincroniza ("gossips") con otros Hubs.

El método principal y más robusto para la extracción de datos es operar un Hub de Farcaster local o conectarse a un Hub público de confianza. Los Hubs exponen una API gRPC y una API HTTP para consultar datos.¹⁸ La estrategia más efectiva para la captura de datos en tiempo real es suscribirse al flujo de eventos del Hub a través de gRPC. Esto proporcionará un flujo continuo de "mensajes" a medida que se propagan por la red. Estos mensajes incluyen:

- **CastAdd:** La creación de una nueva publicación ("cast").
- **ReactionAdd:** Un "like" o "recast".
- **LinkAdd:** Un seguimiento de otro usuario.
- **UserDataAdd:** Actualizaciones del perfil de un usuario (por ejemplo, PFP, nombre de usuario, biografía).

Para cada mensaje, el sistema extraerá el contenido relevante y el Farcaster ID (FID) del autor. El FID se puede utilizar para realizar consultas adicionales a través de la API HTTP del Hub (por ejemplo, `/v1/castsByFid`) para obtener el historial de un usuario o datos de perfil.¹⁸ Aunque existen servicios de terceros como Neynar que simplifican el acceso a los datos de Farcaster¹⁹, el requisito fundamental del proyecto de soberanía de los datos exige la interacción directa con la red de Hubs. Este enfoque garantiza un acceso completo, sin censura y en tiempo real a todo el grafo social de Farcaster.

Parte III: De Transacciones en Bruto a Perspectivas Accionables: La Canalización de Procesamiento de Datos

3.1. La Capa Semántica: Normalización y Estructuración

Los datos extraídos directamente de las diversas fuentes on-chain son heterogéneos, a menudo crípticos y no están listos para el análisis de IA. Antes de que cualquier modelo de lenguaje pueda derivar significado, los datos deben pasar por una rigurosa canalización de Extracción, Transformación y Carga (ETL) para ser normalizados y estructurados. Esta capa semántica es el puente entre el caos de los datos brutos y la coherencia requerida para el análisis inteligente.

El primer paso es definir un conjunto de esquemas de datos estandarizados. Estos esquemas,

probablemente en formato JSON, actuarán como un lenguaje común para todos los tipos de eventos on-chain que el sistema ingiere. La creación de modelos de datos bien definidos es crucial para la consistencia y la escalabilidad. Se proponen los siguientes esquemas iniciales:

- **Transaction:** tx_hash, timestamp, from_address, to_address (contrato), value, function_signature, decoded_params.
- **TokenTransfer:** tx_hash, timestamp, contract_address, token_standard (ERC-20, etc.), from_address, to_address, amount.
- **NftActivity:** tx_hash, timestamp, contract_address, token_id, activity_type (Mint, Transfer, Burn), from_address, to_address.
- **SocialPost:** post_id, protocol (Lens/Farcaster), author_address, author_ens, author_fid, content_text, timestamp, parent_post_id, mentioned_users.
- **ProfileUpdate:** protocol, user_address, update_type (SetENS, SetPFP), new_value, timestamp.

La transformación de los datos brutos a estos esquemas es un paso computacionalmente intensivo. Para las transacciones, implica buscar la firma de la función de 4 bytes en bases de datos públicas (como 4byte.directory) para identificar la función que se llamó. Luego, utilizando el ABI del contrato (obtenido de nuestro registro de contratos), el calldata se decodifica en parámetros con nombre y tipo. De manera similar, los registros de eventos se decodifican utilizando los temas del evento y el ABI. El objetivo es transformar un registro de log de bajo nivel en un evento estructurado y legible por humanos como: { "event": "Swap", "contract": "UniswapV3", "params": { "fromToken": "WETH", "toToken": "USDC", "amountIn": "1.0" } }. Este proceso de enriquecimiento y estructuración es lo que convierte los datos de la blockchain de un simple registro de auditoría en una rica fuente de comportamiento del usuario.

3.2. Construcción del Grafo de Actividad de Usuario Unificado

Con los datos ahora limpios y estructurados, el siguiente paso es organizarlos de una manera que refleje la naturaleza interconectada y componible de Web3. Almacenar estos eventos en tablas relacionales separadas crearía silos de datos, haciendo que las consultas complejas que abarcan diferentes tipos de actividad sean ineficientes y difíciles de escribir. Una estructura de datos mucho más poderosa y adecuada es un **grafo de conocimiento**.

Este modelo de grafo representa las entidades como nodos y las relaciones entre ellas como aristas. Esta estructura mapea de forma nativa cómo funcionan las interacciones en la blockchain.

- **Nodos (Entidades):**
 - UserAddress: El nodo central que representa a un participante seudónimo.

- SmartContract: Representa un protocolo o dApp.
- NftCollection: Una colección específica de NFTs (por ejemplo, Bored Ape Yacht Club).
- Token: Un token fungible específico (por ejemplo, UNI, WETH).
- SocialPost: Una publicación individual en Lens o Farcaster.
- ENSName: Un nombre de dominio ENS.
- **Aristas (Relaciones/Acciones):**
 - OWNS: Conecta un UserAddress a un NftCollection o Token.
 - INTERACTED_WITH: Conecta un UserAddress a un SmartContract.
 - POSTED: Conecta un UserAddress a un SocialPost.
 - COMMENTED_ON: Conecta un SocialPost a otro SocialPost.
 - FOLLOWS: Conecta un UserAddress a otro UserAddress.
 - TRANSFERRED_TO: Conecta dos UserAddress a través de una transferencia de token.
 - MENTIONED: Conecta un SocialPost a un UserAddress o ENSName.

La implementación de este modelo, ya sea en una base de datos de grafos dedicada (como Neo4j o TigerGraph) o como un modelo conceptual sobre una base de datos relacional, es la clave para desbloquear conocimientos de segundo orden. Permite la correlación de identidad y actividad a través de cadenas y protocolos. El nodo UserAddress actúa como el pivote que conecta una publicación en Farcaster (una red de Capa 2) con una acuñación de NFT en la red principal de Ethereum y una transferencia de tokens en Polygon. Esto crea una vista holística y multicadena del viaje de un solo usuario.

El verdadero poder de este enfoque se hace evidente al realizar consultas relacionales complejas. Preguntas que son casi imposibles de responder con registros de eventos aislados se vuelven triviales de ejecutar con una consulta de grafo. Por ejemplo: "Mostrar todos los UserAddress que POSTED sobre una NftCollection específica en Farcaster y luego INTERACTED_WITH el SmartContract de esa colección en un plazo de 24 horas". Este tipo de consulta de comportamiento secuencial es precisamente lo que se necesita para pasar de la simple extracción de palabras clave al análisis predictivo de tendencias. El grafo de actividad unificado no es solo un repositorio de datos; es el "cerebro" analítico del sistema, proporcionando el contexto rico e interconectado que los modelos de IA necesitarán en la siguiente etapa para hacer inferencias sofisticadas sobre la intención y el comportamiento del usuario.

Parte IV: El Núcleo Sensible: Aprovechando la IA Generativa para el Análisis Semántico

4.1. El Analista de IA: Una Arquitectura de Prompting de Múltiples Pasos

Intentar extraer palabras clave, temas e intenciones de datos complejos on-chain con un único y monolítico prompt a un Modelo de Lenguaje Grande (LLM) es una estrategia destinada a producir resultados genéricos y poco fiables. La tarea es demasiado compleja y matizada para un solo paso. En su lugar, este sistema adoptará una arquitectura mucho más robusta y sofisticada: una "cadena de prompts" o un "flujo de trabajo agéntico". Este enfoque imita el proceso cognitivo de un analista humano, descomponiendo una tarea compleja en una secuencia de sub-tareas más simples y manejables. Cada paso en la cadena enriquece los datos y proporciona un contexto más sólido para el siguiente, lo que mejora drásticamente la fiabilidad, la transparencia y la calidad del resultado final.²¹

El flujo de trabajo de análisis de IA se estructurará de la siguiente manera:

1. **Ingesta y Estructuración de Datos:** El LLM recibe un objeto JSON bien definido que representa la actividad reciente de un usuario o un conjunto de publicaciones, extraído del grafo de conocimiento. Este no es texto en bruto, sino datos ya estructurados.
2. **Reconocimiento de Entidades Nombradas (NER):** La primera tarea de la IA es leer el contenido textual (por ejemplo, de las publicaciones sociales) e identificar y etiquetar entidades clave específicas del dominio Web3.
3. **Modelado de Temas:** La segunda tarea es clasificar la actividad y el discurso del usuario en categorías temáticas amplias, proporcionando una visión general de los intereses del usuario.
4. **Inferencia de Intención:** La tercera tarea es sintetizar las acciones, entidades y temas para inferir la motivación subyacente o el objetivo del usuario.
5. **Síntesis de Palabras Clave:** El paso final es generar y puntuar un conjunto de palabras clave relevantes basadas en la totalidad del análisis previo.

Este enfoque secuencial asegura que cada inferencia se base en una base de datos estructurada y verificada, guiando al LLM hacia conclusiones más precisas y contextualmente relevantes.

4.2. Paso 1: Reconocimiento de Entidades Nombradas (NER) para el Léxico de Web3

El Reconocimiento de Entidades Nombradas (NER) es una tarea fundamental del

Procesamiento del Lenguaje Natural (PLN) que implica identificar y clasificar entidades en texto en categorías predefinidas.²³ Los LLMs modernos han demostrado ser excepcionalmente buenos en esta tarea, incluso en entornos de pocos ejemplos ("few-shot").²⁴ Para este sistema, el objetivo no es utilizar categorías genéricas como "Persona" u "Organización", sino definir un conjunto personalizado de entidades que capturen el léxico único de Web3.

El diseño del prompt es crucial. El prompt debe definir claramente las categorías de entidades personalizadas. Una taxonomía inicial podría incluir:

- PROTOCOL: Nombres de blockchains, protocolos DeFi, plataformas de juegos, etc. (Ej: "Uniswap", "Farcaster", "Arbitrum").
- TOKEN_TICKER: Símbolos de tokens fungibles (Ej: "\$ETH", "\$OP", "\$DEGEN").
- PERSONA: Nombres de usuario, dominios ENS o identidades conocidas en el espacio (Ej: "@vitalik.eth", "dwr", "cobie").
- EVENT: Nombres de conferencias, hackathons o eventos comunitarios (Ej: "ETHDenver", "Devcon").
- CONCEPT: Términos técnicos o jerga de la industria (Ej: "MEV", "ZK-proofs", "airdrop", "frames").

La técnica más efectiva para esta tarea es el **prompting de pocos ejemplos (few-shot prompting)**.²⁶ En lugar de solo dar instrucciones, el prompt incluirá 3-5 ejemplos de alta calidad de texto ya anotado. Esto permite que el LLM aprenda por analogía, mejorando drásticamente su precisión en la clasificación de entidades específicas del dominio.

A continuación se muestra una estructura de prompt ejemplar, diseñada siguiendo las mejores prácticas para modelos como Claude de Anthropic, que incluye un rol de sistema, ejemplos de pocos disparos y el uso de etiquetas XML para estructurar la entrada y pre-llenar la salida ²⁷:

XML

<system>

Usted es un analista experto en Web3. Su tarea es realizar el Reconocimiento de Entidades Nombradas en el texto proporcionado. Extraiga y clasifique las entidades en las siguientes categorías:. La salida debe ser un único array JSON de objetos.

</system>

<user>

Aquí hay algunos ejemplos:

<example>

```
<text>Acabo de escuchar que @dwr es optimista sobre la nueva función de frames de Farcaster. Creo que podría ser grande para el ecosistema de $DEGEN.</text>  
<entities>
```

```
</entities>  
</example>
```

```
<example>  
<text>¿Alguien va a ETHDenver este año? Estoy buscando aprender más sobre las pruebas ZK y la abstracción de cuentas.</text>  
<entities>
```

```
</entities>  
</example>
```

Ahora, analice el siguiente texto:

```
<text_to_analyze>  
{{INPUT_TEXT_FROM_SOCIAL_POST}}  
</text_to_analyze>  
</user>  
<assistant>  
<entities>
```

La salida de este paso será un objeto JSON estructurado que etiqueta todas las entidades relevantes en el texto, proporcionando la primera capa de enriquecimiento semántico.

4.3. Paso 2: Modelado Dinámico de Temas con LLMs

Los modelos de temas tradicionales (como LDA) a menudo tienen dificultades con textos cortos y ruidosos como las publicaciones en redes sociales, ya que dependen de la co-ocurrencia de palabras que puede no estar presente. Los LLMs, sin embargo, pueden realizar un modelado de temas de cero disparos ("zero-shot") con una eficacia notable, aprovechando su profunda comprensión semántica del lenguaje. Este sistema adaptará los conceptos del enfoque PromptTopic, que aprovecha los LLMs para la identificación de temas sin necesidad de un ajuste fino extenso.³⁰

El prompt para esta etapa le pedirá al LLM que asigne uno o más temas a una pieza de contenido o a un conjunto de actividades del usuario. Se proporcionará una lista predefinida pero flexible de temas de Web3. Una taxonomía inicial podría ser:

- DeFi (Finanzas Descentralizadas)
- NFTs (Arte, Coleccionables, PFP)

- Infraestructura (Escalado, Nodos, Oráculos)
- Gobernanza (DAOs, Votación)
- SocialFi (Redes Sociales Descentralizadas)
- Gaming (Juegos Web3, Metaverso)
- Regulación y Política

Una característica clave de este prompt es su dinamismo. Se le puede indicar al LLM que si ninguna de las categorías existentes encaja bien, puede sugerir una `new_topic` con una breve justificación. Este mecanismo de retroalimentación permite que la taxonomía de temas del sistema evolucione orgánicamente con el tiempo, adaptándose a las narrativas emergentes en el ecosistema.

Los temas asignados a publicaciones individuales se agregarán a nivel de usuario, de colección de NFT y a nivel global. Esto permitirá identificar no solo los intereses de un usuario individual, sino también los temas que están ganando tracción en toda la red, una señal crucial para la detección de tendencias.

4.4. Pasos 3 y 4: Inferencia de la Intención del Usuario y Síntesis de Palabras Clave

Esta es la culminación de la canalización de IA, donde el sistema pasa de la descripción (qué se dijo) a la inferencia (por qué se dijo y qué significa).

Inferencia de Intención

La entrada a este prompt será la salida estructurada de los pasos anteriores: las acciones on-chain originales, las entidades extraídas y los temas asignados. Esta rica entrada contextual permite al LLM hacer una inferencia mucho más informada sobre la intención del usuario. El prompt le pedirá al LLM que clasifique el comportamiento del usuario en una categoría de una taxonomía de intenciones nativa de Web3. Comprender la intención es fundamental para contextualizar las palabras clave.³²

Una taxonomía de intenciones inicial podría incluir:

- **Búsqueda de Información:** El usuario está haciendo preguntas, buscando explicaciones o tratando de entender un concepto.
- **Inversión/Especulación:** El discurso se centra en precios, análisis de mercado, potencial de crecimiento o "alpha".

- Participación Comunitaria: El usuario está interactuando con otros miembros, participando en la cultura de un proyecto o construyendo una red social.
- Feedback de Producto: El usuario está informando de un error, sugiriendo una característica o criticando una dApp.
- Desarrollo Técnico: La conversación es sobre código, arquitectura de contratos inteligentes o construcción de nuevas aplicaciones.
- Airdrop Farming: Las acciones y el discurso indican un patrón de comportamiento diseñado para calificar para futuras recompensas de tokens.

Síntesis de Palabras Clave

El prompt final de la cadena es donde se genera el producto principal. Este prompt recibirá toda la información estructurada acumulada: las acciones, las entidades, los temas y la intención inferida. Se le pedirá al LLM que asuma el rol de un experto en optimización de motores de búsqueda (SEO) y marketing digital. Con base en el perfil completo del usuario y su intención, el LLM deberá:

1. **Generar un conjunto de palabras clave:** Términos y frases que un usuario con este perfil e intención podría buscar en Google, preguntar en un chatbot o usar para encontrar información relevante.
2. **Clasificar las palabras clave:** Agruparlas en categorías como "cola larga" (específicas), "de marca" (relacionadas con el protocolo), "informativas" y "transaccionales".
3. **Puntuar las palabras clave:** Asignar una puntuación de relevancia y especificidad a cada palabra clave, basándose en qué tan directamente se relaciona con los datos de entrada.

La arquitectura de múltiples pasos crea un "proceso de pensamiento estructurado" para el LLM. Cada etapa refina y enriquece los datos, proporcionando un andamiaje contextual para la siguiente. Este enfoque de "Cadena de Pensamiento" (Chain-of-Thought - CoT) evita que el LLM haga conexiones superficiales y lo obliga a construir sus conclusiones sobre una base de entidades reconocidas y temas clasificados, lo que aumenta drásticamente la calidad, la relevancia y la accionabilidad de la producción final de palabras clave.²¹

A continuación se presenta una tabla que resume la estrategia de prompts para el núcleo de IA, sirviendo como un plan de implementación para el equipo de ingeniería de IA.

Tabla 1: Matriz de Estrategia de Prompts para el Análisis de Web3

Objetivo Analítico	Etapas del Pipeline	LLM Sugerido	Técnica de Prompting	Fragmento del Prompt Central
--------------------	---------------------	--------------	----------------------	------------------------------

				(en inglés para claridad técnica)
Identificar conceptos clave de Web3 en texto	1. Reconocimiento de Entidades (NER)	Anthropic Claude 3 Opus	Rol de Sistema, Pocos Ejemplos (Few-shot), Etiquetas XML	<system>You are a Web3 NER expert. Extract entities:.</system><user><example>...</example><text_to_analyze>...</text_to_analyze></user><assistant><entities>
Clasificar el tema de una conversación	2. Modelado de Temas	DeepSeek-V2	Cero Ejemplos (Zero-shot), Instrucciones Claras	Classify the following text into one of these topics:. If none fit, output a new topic and a justification.
Determinar la motivación del usuario	3. Inferencia de Intención	Anthropic Claude 3 Opus	Cadena de Pensamiento (CoT), Entrada Estructurada	Given the user's actions;, entities mentioned;, and topics discussed;, infer their primary intent from this list:. Explain your reasoning step-by-step.
Generar términos de	4. Síntesis de	DeepSeek-V2	Rol de Sistema,	<system>You are a

búsqueda accionables	Palabras Clave		Entrada Estructurada	world-class SEO strategist.</system><user>Based on this complete user profile (intent, topics, entities), generate a scored list of relevant long-tail, informational, and transactional keywords.</user>
----------------------	----------------	--	----------------------	---

Parte V: Plan de Implementación y Operaciones del Sistema

5.1. Integración de las Canalizaciones de Datos e IA

La implementación exitosa de este sistema requiere una orquestación robusta que conecte la capa de extracción de datos, la canalización de procesamiento y el núcleo de análisis de IA. Un gestor de flujos de trabajo como Apache Airflow o Prefect es la herramienta ideal para programar, ejecutar y monitorear este complejo proceso de extremo a extremo.

El flujo de trabajo orquestado se puede describir en los siguientes pasos:

1. **Disparo de la Extracción de Datos:** Se programarán trabajos (DAGs en Airflow) para que se ejecuten a intervalos frecuentes (por ejemplo, cada minuto) para obtener los datos más recientes. Trabajos separados se encargarán de:
 - Consultar los nodos de la blockchain para obtener nuevos bloques y transacciones.

- Conectarse a los puntos finales de la API de Lens y Farcaster para obtener nuevas publicaciones sociales.
 - Monitorear los contratos de ENS y NFT para detectar nueva actividad.
2. **Procesamiento y Carga en el Grafo:** A medida que se ingieren los datos brutos, se activará la canalización de procesamiento de datos. Esta etapa normalizará los datos en los esquemas definidos y los cargará en la base de datos de grafos, creando o actualizando nodos y aristas.
 3. **Activación de la Cadena de Análisis de IA:** El sistema de flujo de trabajo monitoreará el grafo de conocimiento. La creación de un nuevo nodo de SocialPost o un patrón significativo de actividad de un UserAddress activará la cadena de análisis de IA de múltiples pasos.
 4. **Interacción con las API de LLM:** Los datos estructurados relevantes se pasarán a las API de LLM (DeepSeek, Anthropic) utilizando sus respectivos SDKs. El código de integración debe ser resiliente, implementando un manejo de errores adecuado, lógica de reintentos con retroceso exponencial y validación de las respuestas de la API para asegurar que se adhieran al formato esperado (por ejemplo, JSON válido).
 5. **Almacenamiento de los Resultados de IA:** La salida estructurada de la cadena de IA (entidades, temas, intenciones, palabras clave) se almacenará en una base de datos optimizada para la consulta y el análisis, como PostgreSQL (para datos relacionales) o Elasticsearch (para búsqueda de texto completo y análisis). Este almacenamiento secundario servirá como el backend para la interfaz de usuario.

5.2. Puntuación de Palabras Clave y Análisis de Tendencias

La simple generación de palabras clave no es suficiente; deben ser puntuadas y clasificadas para ser accionables. Se propone un algoritmo de puntuación multifactorial que vaya más allá de la simple frecuencia para capturar el verdadero impulso de una palabra clave o tema.

El algoritmo de puntuación podría basarse en los siguientes factores, ponderados según la estrategia del producto:

- **Frecuencia:** ¿Con qué frecuencia aparecen la palabra clave o sus entidades y temas relacionados en el conjunto de datos durante un período de tiempo determinado?
- **Velocidad (Velocity):** ¿Está aumentando la frecuencia con el tiempo? El cálculo de la primera y segunda derivada de la frecuencia puede identificar temas que están no solo creciendo, sino acelerando su crecimiento, una señal clave de una tendencia emergente.
- **Conectividad (Connectedness):** ¿Cuántos usuarios de alto valor están asociados con esta palabra clave? Se puede definir "alto valor" basándose en métricas del grafo, como el número de seguidores, el valor de los activos de NFT o el volumen de transacciones. Una palabra clave discutida por "ballenas" o desarrolladores influyentes es más

significativa.

- **Fuerza de la Intención (Intent Strength):** Las palabras clave asociadas con intenciones de alto compromiso (como Inversión/Especulación o Desarrollo Técnico) podrían recibir una puntuación más alta que las asociadas con intenciones de bajo compromiso (como Búsqueda de Información).
- **Novedad (Novelty):** Se puede asignar una puntuación más alta a las palabras clave y entidades que han aparecido recientemente por primera vez, para destacar conceptos completamente nuevos.

Estos factores se combinarán en una puntuación de "Tendencia" normalizada que se puede utilizar para clasificar las palabras clave y los temas en el panel de control del usuario.

5.3. La Interfaz de Usuario: Un Panel de Análisis Nativo de Web3

La interfaz de usuario final debe reflejar la naturaleza única de los datos que impulsa el sistema. Un simple listado de palabras clave y volúmenes de búsqueda sería un flaco favor a la riqueza del conjunto de datos subyacente. El panel de control debe ser una herramienta de exploración que permita a los usuarios sumergirse en el contexto on-chain.

Los diseños conceptuales para el front-end deberían incluir las siguientes características:

- **Panel de Tendencias Emergentes:** Una vista principal que no solo muestra las palabras clave de mayor puntuación, sino también los **temas y conceptos** que están ganando tracción, visualizados con gráficos de velocidad.
- **Explorador de Palabras Clave Contextual:** Al hacer clic en una palabra clave, el usuario no solo ve métricas. En su lugar, se le presenta un panel de control detallado para esa palabra clave, que muestra:
 - **Conversaciones On-Chain:** Un feed en vivo de las publicaciones de Lens y Farcaster que mencionan la palabra clave.
 - **Perfiles Clave:** Una lista de los usuarios más influyentes (basado en la puntuación de conectividad) que discuten el tema.
 - **Actividad de Contratos Relacionados:** Un resumen de las interacciones con los contratos inteligentes asociados a las entidades de la palabra clave (por ejemplo, si la palabra clave es "Friend.tech", mostraría el volumen de transacciones en el contrato de Friend.tech).
- **Visualizador de Grafo:** Una herramienta interactiva que permite a los usuarios explorar una subsección del grafo de conocimiento. Por ejemplo, podrían visualizar la relación entre una colección de NFT, los principales poseedores que también son activos en Farcaster y los temas de los que discuten.

Este tipo de interfaz transforma la herramienta de un simple generador de palabras clave a

una plataforma de inteligencia de mercado de Web3, proporcionando a los usuarios no solo el "qué" (las palabras clave populares), sino también el "quién" (los influenciadores), el "dónde" (los protocolos) y el "porqué" (la intención del usuario).

A continuación se presenta una tabla que detalla las fuentes de datos y las señales analíticas que proporcionan, sirviendo como un plan estratégico para el equipo de ingeniería de datos.

Tabla 2: Matriz de Fuentes de Datos y Señales Analíticas

Fuente de Datos	Tipo de Dato Bruto	Método de Extracción	Información Clave Extraída	Señal Analítica Derivada	Relevancia para Palabras Clave
Nodos de Blockchain (EVM)	Bloques, Transacciones, Logs	Conexión RPC directa	Llamadas a funciones, eventos emitidos, transferencias de valor	Comportamiento transaccional, interacción con dApps, flujos económicos	Media: Contextualiza la actividad económica relacionada con temas
Farcaster Hub Stream	Mensajes CastAdd, ReactionAdd	Suscripción gRPC	Contenido de texto, autor (FID), URL padre, reacciones	Sentimiento social, narrativas emergentes, interés del usuario, viralidad	Alta: Fuente principal de texto para la minería de palabras clave
API GraphQL de Lens Protocol	Objetos de Publication	Consultas GraphQL paginadas	Contenido de texto, autor (ProfileID), espejos, comentarios	Grafo de influencia, compromiso de la comunidad, temas de discusión	Alta: Fuente principal de texto y métricas de compromiso
Contrato de	Eventos	Escucha de	Nombre de	Construcci	Media:

Registro ENS	NameRegistered, AddrChanged	eventos de contrato	dominio, dirección del propietario, registros de texto	ón de identidad, afiliación de marca personal/proyecto	Proporciona contexto de identidad para los autores de contenido
Contratos de NFT (ERC-721/1155)	Eventos Transfer, llamada a tokenURI	Escucha de eventos, llamadas RPC	Propietario del token, metadatos (atributos, descripción)	Pertenencia a la comunidad, intereses culturales, capital social	Media: Contextualiza los intereses y afiliaciones de los usuarios

Conclusión

La arquitectura delineada en este informe representa un cambio fundamental en el enfoque de la inteligencia de mercado, alejándose de los modelos basados en inferencias de Web2 hacia un paradigma de observación directa habilitado por la transparencia de la blockchain. Al construir una canalización de datos propietaria y soberana, el sistema propuesto obtiene una ventaja estratégica inherente: la capacidad de analizar la totalidad de la actividad on-chain sin la abstracción o el filtrado de intermediarios. Esta independencia no es solo una elección técnica, sino la base para desarrollar una propiedad intelectual única en la forma de modelos analíticos que pueden descubrir patrones y tendencias invisibles para aquellos que dependen de infraestructuras de datos estandarizadas.

El núcleo del sistema, un flujo de trabajo de IA de múltiples pasos, está diseñado para la precisión y la profundidad contextual. En lugar de tratar al LLM como una caja negra, la arquitectura de la cadena de prompts lo guía a través de un proceso lógico que imita el análisis humano: primero identifica entidades concretas, luego clasifica temas amplios, infiere la intención subyacente y, finalmente, sintetiza palabras clave accionables. Este enfoque estructurado mitiga el riesgo de respuestas genéricas y asegura que la salida final esté firmemente anclada en la evidencia extraída de la cadena de bloques.

La implementación exitosa de este plan dará como resultado más que una simple herramienta de palabras clave. Creará una plataforma de inteligencia de mercado nativa de Web3 capaz

de proporcionar una visión de alta fidelidad de las narrativas emergentes, el comportamiento del consumidor y las intenciones del usuario. Al conectar el discurso social descentralizado con la identidad seudónima y la actividad económica verificable, el sistema puede responder no solo "qué" está en tendencia, sino "quién" está impulsando la tendencia y "por qué". En un mercado digital cada vez más definido por la autenticidad y la participación verificable, la capacidad de extraer inteligencia directamente de la fuente de verdad on-chain será un diferenciador crítico y duradero.

Obras citadas

1. NFT API | Build on 30+ chains - Alchemy, fecha de acceso: septiembre 7, 2025, <https://www.alchemy.com/nft-api>
2. Moralis NFT API - NFT Tools - Alchemy, fecha de acceso: septiembre 7, 2025, <https://www.alchemy.com/dapps/moralis-nft-api>
3. Exploring the Alchemy NFT API - Hackernoon, fecha de acceso: septiembre 7, 2025, <https://hackernoon.com/exploring-the-alchemy-nft-api>
4. Overview | Moralis API Documentation, fecha de acceso: septiembre 7, 2025, <https://docs.moralis.com/web3-data-api/evm/nft-api>
5. Exploring the Alchemy NFT API - DEV Community, fecha de acceso: septiembre 7, 2025, https://dev.to/envoy_/the-ultimate-nft-api-24d5
6. Build NFT metadata access control with Ethereum signatures and AWS Lambda authorizers, fecha de acceso: septiembre 7, 2025, <https://aws.amazon.com/blogs/database/build-nft-metadata-access-control-with-ethereum-signatures-and-aws-lambda-authorizers/>
7. NFTs with On-chain Metadata on TON: Tutorial | by Vladislav Lenskii | MiKi Blockchain, fecha de acceso: septiembre 7, 2025, <https://medium.com/miki-dev/nfts-with-on-chain-metadata-on-ton-tutorial-55ac0cbb17d5>
8. Address Lookup | ENS Docs, fecha de acceso: septiembre 7, 2025, <https://docs.ens.domains/web/resolution/>
9. Where are ENS names stored? - Support - ENS Domains, fecha de acceso: septiembre 7, 2025, <https://support.ens.domains/en/articles/7900423-where-are-ens-names-stored>
10. Ethereum Name Service (ENS) — web3.py 7.13.0 documentation - Read the Docs, fecha de acceso: septiembre 7, 2025, https://web3py.readthedocs.io/en/stable/ens_overview.html
11. Lens API GraphQL API Reference, fecha de acceso: septiembre 7, 2025, <https://api-v2-docs.lens.xyz/>
12. Lenses API docs, fecha de acceso: septiembre 7, 2025, <https://api.lenses.io/>
13. Lens API Documentation, fecha de acceso: septiembre 7, 2025, <https://docs.api.lens.org/>
14. Lens Extension Development, fecha de acceso: septiembre 7, 2025, <https://api-docs.k8slens.dev/>
15. LENS API Guide v2.0, fecha de acceso: septiembre 7, 2025, <https://docs.lightcast.dev/guides/lens-api-guide-v2-0>

16. Message API - Farcaster Docs, fecha de acceso: septiembre 7, 2025, <https://docs.farcaster.xyz/reference/hubble/httpapi/message>
17. farcasterxyz/protocol: Specification of the Farcaster Protocol - GitHub, fecha de acceso: septiembre 7, 2025, <https://github.com/farcasterxyz/protocol>
18. HTTP API - Farcaster Docs, fecha de acceso: septiembre 7, 2025, <https://docs.farcaster.xyz/reference/hubble/httpapi/httpapi>
19. Overview - Farcaster Docs, fecha de acceso: septiembre 7, 2025, <https://docs.farcaster.xyz/reference/>
20. Farcaster Client API Reference / Farcaster Docs, fecha de acceso: septiembre 7, 2025, <https://docs.farcaster.xyz/reference/warpcast/api>
21. Prompt Chaining - Prompt Engineering Guide, fecha de acceso: septiembre 7, 2025, https://www.promptingguide.ai/techniques/prompt_chaining
22. GPT-5 prompting guide | OpenAI Cookbook, fecha de acceso: septiembre 7, 2025, https://cookbook.openai.com/examples/gpt-5/gpt-5_prompting_guide
23. Prompt Engineering for Named Entity Recognition (NER) - DS With Mac, fecha de acceso: septiembre 7, 2025, <https://dswithmac.com/posts/prompt-eng-ner/>
24. FsPONER: Few-shot Prompt Optimization for Named Entity Recognition in Domain-specific Scenarios - arXiv, fecha de acceso: septiembre 7, 2025, <https://arxiv.org/html/2407.08035v2>
25. Improving large language models for clinical named entity recognition via prompt engineering | Journal of the American Medical Informatics Association | Oxford Academic, fecha de acceso: septiembre 7, 2025, <https://academic.oup.com/jamia/article/31/9/1812/7590607>
26. PromptNER : Prompting For FewShot Named Entity Recognition - OpenReview, fecha de acceso: septiembre 7, 2025, <https://openreview.net/forum?id=WDQ9ZzsgDL>
27. Claude 4 prompt engineering best practices - Anthropic API, fecha de acceso: septiembre 7, 2025, <https://docs.anthropic.com/en/docs/build-with-claude/prompt-engineering/claude-4-best-practices>
28. Prompt engineering overview - Anthropic API, fecha de acceso: septiembre 7, 2025, <https://docs.anthropic.com/en/docs/build-with-claude/prompt-engineering/overview>
29. Giving Claude a role with a system prompt - Anthropic API, fecha de acceso: septiembre 7, 2025, <https://docs.anthropic.com/en/docs/build-with-claude/prompt-engineering/system-prompts>
30. Prompting Large Language Models for Topic Modeling - arXiv, fecha de acceso: septiembre 7, 2025, <https://arxiv.org/html/2312.09693v1>
31. Prompting Large Language Models for Topic Modeling - arXiv, fecha de acceso: septiembre 7, 2025, <https://arxiv.org/pdf/2312.09693>
32. A Beginner's Guide to LLM Intent Classification for Chatbots - Vellum AI, fecha de acceso: septiembre 7, 2025, <https://www.vellum.ai/blog/how-to-build-intent-detection-for-your-chatbot>

33. Using Large Language Models to Generate, Validate, and Apply User Intent Taxonomies - arXiv, fecha de acceso: septiembre 7, 2025,
<https://arxiv.org/pdf/2309.13063>
34. Complexity-Based Prompting, fecha de acceso: septiembre 7, 2025,
https://learnprompting.org/docs/advanced/thought_generation/complexity_based_prompting