### **Evaluación Crítica de Idea: Plataforma de Denuncias de Violencia con Evidencia Inmutable**

**1. VEREDICTO DE VIABILIDAD (36H):** VIABLE CON SIMPLIFICACIÓN EXTREMA

**2. Análisis del Problema y Relevancia Social**

* **Problema Real (Bolivia/LATAM):** El problema es de máxima relevancia. En Bolivia, como en gran parte de la región, la violencia de género y doméstica es una crisis persistente. Las víctimas a menudo no denuncian por miedo a represalias, desconfianza en el sistema judicial, o falta de pruebas contundentes. La idea ataca directamente la necesidad de un registro de evidencia seguro, anónimo y a prueba de manipulaciones.
* **Beneficiarios Claros:**
  1. **Víctimas:** Obtienen una herramienta para crear un registro de evidencia con fecha y hora inalterables (timestamping), sin tener que exponer su identidad inmediatamente. Esto puede ser un primer paso crucial para construir un caso.
  2. **ONGs y Asesores Legales:** Podrían recibir evidencia verificable en su origen y fecha, lo que fortalece los casos de apoyo a las víctimas.
* **Impacto Potencial:** Muy alto. Si se implementa correctamente, podría convertirse en una herramienta estándar para la recopilación de pruebas preliminares, empoderando a las víctimas y creando una nueva forma de presión social y legal sobre las instituciones.

**3. Propuesta de MVP y Arquitectura para 36 Horas**

* **MVP Mínimo y Realista:** El alcance original es imposible de completar en 36 horas. El componente de IA es una trampa mortal en un buildathon. **Deben eliminarlo por completo.** El MVP funcional para el domingo debe ser:
  1. **Frontend Simple:** Una única página web con un formulario para escribir un testimonio y adjuntar **un solo archivo** (imagen o PDF). No más.
  2. **Conexión de Wallet:** El usuario se conecta con su wallet (ej. MetaMask) para mantener el seudoanonimato. No se solicita ningún otro dato personal.
  3. **Flujo de Evidencia:**
     + El usuario sube el archivo.
     + El frontend lo envía a un servicio de IPFS (como Pinata) para obtener un hash (CID).
     + El frontend invoca una función en un Smart Contract para registrar ese hash de IPFS junto a la dirección del denunciante y un timestamp.
  4. **Página de Confirmación:** Se muestra al usuario el hash de la transacción y el hash de IPFS, explicándole que esa es su prueba inmutable.
* **Arquitectura Técnica Sugerida:**
  + **Frontend:** Vite + React + Tailwind CSS (para máxima velocidad de desarrollo). Usar librerías como ethers.js o viem para la interacción con la blockchain.
  + **Almacenamiento:** **IPFS** a través de una API de un servicio de "pinning" como **Pinata**. Intentar correr un nodo propio en 36h es un riesgo innecesario.
  + **Smart Contract:** Solidity, desarrollado con **Foundry** o Hardhat. Debe ser extremadamente simple: una estructura Denuncia {bytes32 ipfsHash; uint256 timestamp; address denunciante;} y una única función registrarDenuncia(bytes32 \_ipfsHash).
  + **Blockchain:** Desplegar en una red de prueba (Testnet) de una L2 como **Base Sepolia** u **Optimism Sepolia**. Son rápidas, baratas y el tooling es excelente.
  + **UI/UX:** Un diseño simple en Figma que se enfoque únicamente en el flujo del MVP descrito.
* **Uso de Tecnología Descentralizada:** La justificación es sólida y no es un adorno. La blockchain se usa para lo que es mejor: **crear un registro público, inmutable y resistente a la censura.** El hash en la cadena es una prueba irrefutable de que "X datos existían en el momento Y". Esto es imposible de lograr con una base de datos centralizada tradicional, que puede ser alterada o eliminada por sus administradores.

**4. Cumplimiento de Reglas y Entregables**

* **Desarrollo desde Cero:** La idea simplificada es perfecta para ser construida desde cero. [Inferencia] La idea original, con su componente de IA, habría forzado al equipo a buscar librerías o APIs externas complejas, lo cual, aunque permitido, es una trampa que consume tiempo valioso de integración y pruebas, yendo en contra del espíritu del buildathon.
* **Plan de Entregables (Estimación de Tiempo):**
  + Smart Contract (diseño, codificación, pruebas y despliegue): [Estimación] 6-8 horas
  + Desarrollo MVP (Frontend, integración con wallet e IPFS): [Estimación] 16-20 horas
  + UI/UX, Pitch Deck y Video: [Estimación] 8-10 horas
  + **Total Estimado:** [Estimación] 30-38 horas. El cronograma es muy ajustado y no deja margen para errores significativos.
* **Riesgo Principal:** El mayor riesgo es el **"scope creep" (ampliación del alcance)**. La tentación de añadir "solo una cosa más" (como la encriptación compleja, múltiples tipos de archivo, o un dashboard para ONGs) destruirá sus posibilidades de entregar un demo funcional. El segundo riesgo es subestimar el tiempo de integración entre el frontend, IPFS y el contrato.

**5. Conclusión y Recomendaciones Estratégicas**

* **Valoración Final:** La idea tiene un potencial enorme y un encaje perfecto con los valores del ecosistema Web3. Es una excelente candidata para el buildathon **SI Y SOLO SI** el equipo adopta la versión ultra-simplificada del MVP. La propuesta original es una receta para el fracaso en este evento. Un demo funcional, aunque mínimo, de este concepto es mucho más poderoso que un proyecto ambicioso a medio terminar.
* **Sugerencia Crítica para el Equipo:** **ABANDONEN INMEDIATAMENTE Y POR COMPLETO CUALQUIER COMPONENTE DE INTELIGENCIA ARTIFICIAL.** Olvídense de verificar la "veracidad" o detectar contenido falso; eso es un problema de investigación, no un feature de fin de semana. Su única misión para el domingo es demostrar un flujo impecable: **un usuario anónimo sube una prueba, esta se guarda en IPFS y su hash queda grabado para siempre en la blockchain.** Si logran demostrar eso de forma clara y concisa, tienen un proyecto ganador en sus manos.