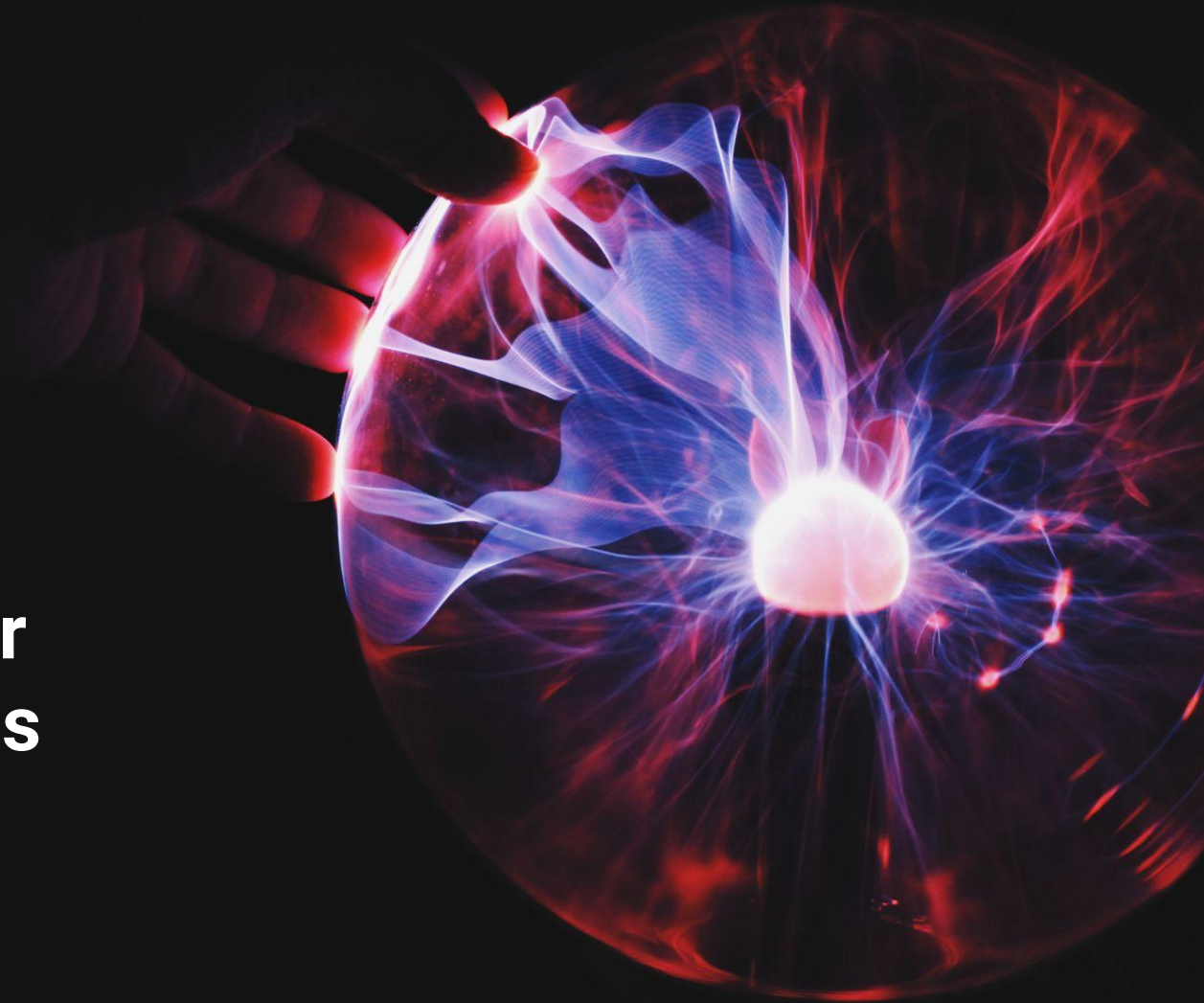


INPUT | OUTPUT

CIP-30 & Firmar Tx con billeteras de navegador



+ CIP-30 _

CIP-30: Qué es una Cardano Improvement Proposal (CIP)?

- Una [Cardano Improvement Proposal \(CIP\)](#) es un **documento de diseño formalizado** y el nombre del proceso mediante el cual se producen y enumeran dichos documentos.
- Una CIP proporciona información o describe un cambio en el ecosistema, los procesos o el medio ambiente de Cardano de forma concisa y con suficiente detalle técnico.
- Las CIP pueden ser en general tratadas como **fuentes de verdad* para implementaciones**. *Hay veces que detalles técnicos quedan sin especificar/evolución de la tecnología hace que cada desarrollador implemente ligeramente distinto o que se cree una nueva CIP.
- Como desarrollador de Cardano, deberías familiarizarte con las CIP aprobadas y propuestas.

CIP-30: Que cubre la CIP-30?

- La [CIP-30](#) describe un **punto de comunicación** que permite que las páginas web (es decir, dApps) interactúen con las billeteras Cardano.
- Para facilitar el desarrollo de DApps, fue necesario una forma estandarizada para que las DApps se comuniquen con la billetera del usuario.
- CIP-30 provee a detallada API con data types que permite tanto obtener UTxOs, collateral, dirección, y otra información sobre la billetera del usuario, cómo firmar y enviar transacciones.
- Cómo me ayuda todo esto a mí como desarrollador de DApps? Si usas una librería compatible con CIP-30, en lugar de integrar billeteras una por una, tu DApp automáticamente funciona para todas las billeteras de Cardano.

CIP-30: CIP relacionadas

- La [CIP-45](#) introduce un método de comunicación descentralizado entre dApps y billeteras basado en rastreadores WebTorrent y WebRTC.
- La [CIP-103](#) expande CIP-30 para proporcionar un endpoint adicional para que DApps firmen múltiples transacciones de forma masiva.
- La [CIP-95](#) expande CIP-30 para brindar soporte para CIP-1694 (“Un primer paso hacia la gobernanza descentralizada en cadena”).

+ FIRMAR TX_

Con billeteras de navegador

6

Firmar Tx con billeteras de navegador: Pasos

1. Crear proyecto con: `npx create-mesh-app starter-next-ts-template`
2. Mostrar ADA de usuario en DApp UI con `BrowserWallet.enable()` y `getLovelace()`
3. Manejar cualquier billetera CIP-30 con `<CardanoWallet/>` y `useWallet()`
4. Crear input numérico de cantidad de ADA a enviar y a que dirección on `<input/>`.
5. Crear código off-chain para firmar y enviar Tx con `Transaction()` y `wallet`

Fi

n Preguntas ?



INPUT | OUTPUT