

Antonio Sánchez

**Embajador de Cardano en
España**

Algunos datos sobre mí

Soy Técnico Superior en Administración de Sistemas Informáticos, Perito Judicial Informático, experto en Hacking ético de sistemas y redes y especialista en seguridad informática.

**CEO Inforbyt | Presidente Blockchain Andalucía | Embajador Cardano | CISO
Blockchain OpenLab | CTO KeepSafe**

Cardano para Desarrolladores

Introducción
a Cardano

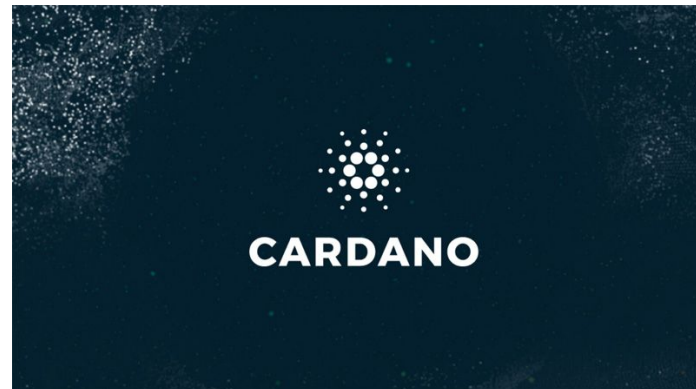
Máquinas
Virtuales

Contratos
Inteligentes

Introducción a Cardano

Cardano

- Es el primer proyecto BlockChain que está desarrollado partiendo de una filosofía **científica**.
- Cardano utiliza un algoritmo de consenso por Prueba de Participación con algunas particularidades que han sido integradas por ellos y que lleva el nombre de **Ouroboros**.
- Está basado en un **protocolo multicapa**, utiliza dos capas vinculadas entre sí, la capa de liquidación y la capa de control.
 - La CSL actúa como el Libro contable (Ledger) del balance y es la primera capa de la plataforma Cardano.
 - La CCL es la segunda capa que conforma Cardano, contienen toda la información sobre lo que pasa con las transacciones que se realizan en la plataforma.



INPUT | OUTPUT

EMURGO

Máquinas Virtuales



IELE es una máquina virtual dedicada que proporciona una base para el protocolo Cardano blockchain.

Ejecuta y verifica contratos inteligentes, además de proporcionar un lenguaje legible para los desarrolladores de blockchain.

El testnet IELE proporciona un mejor rendimiento y está basado en registros, lo que significa que puede hacer uso de una gama más amplia de análisis y optimizaciones que una máquina virtual basada en pila como KEVM.

La depuración de contratos es más fácil porque IELE tiene un código de error de estado diferente para cada una de las excepciones que pueden ocurrir al ejecutar las funciones dentro de un contrato.



KEVM

KEVM es una versión mejorada por construcción de la máquina virtual de Ethereum (EVM).

Permite a los desarrolladores experimentar con cualquier contrato inteligente que pueda ejecutarse en el EVM y ofrece una seguridad y un rendimiento mejorado.

Conjunto de habilidades y requisitos

Para usar las redes de prueba KEVM e IELE se requieren las siguientes habilidades:

1. Familiaridad con la escritura de contratos inteligentes en Solidity.
2. Necesitará un conjunto de contratos inteligentes bien definidos para usar en las redes de prueba.
3. Conocimiento avanzado del protocolo de llamada a procedimiento remoto JSON (RPC).
4. No es necesario registrarse para usar testnet.

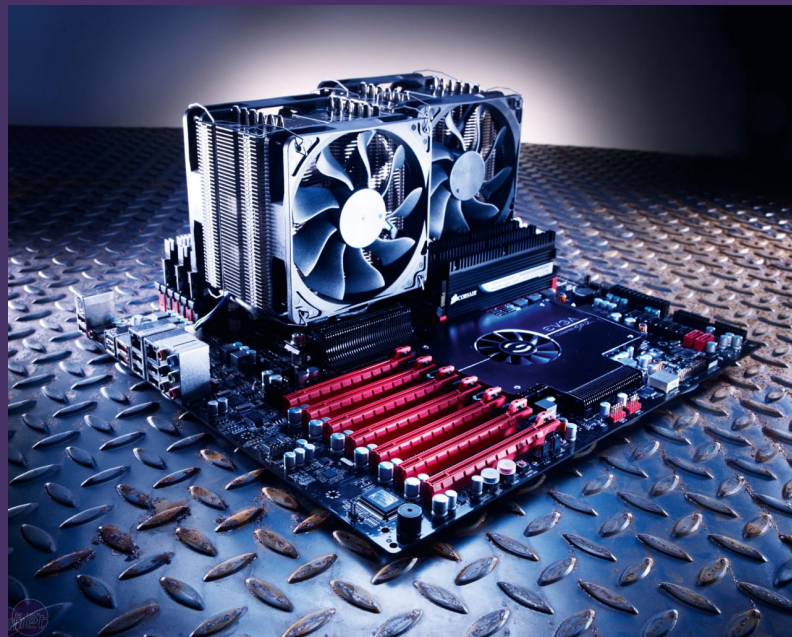


Para el testnet KEVM:

1. Se necesita una máquina virtual Java (JVM) para ejecutar el cliente. La versión debe ser 1.8.x. El cliente no ha sido probado con JVM 1.9.
2. Salida de 64 bits.
3. Espacio en disco adecuado: se necesitará suficiente espacio para que la cadena de bloques crezca. Se recomienda un SSD de al menos 35G.
4. El cliente Mantis se ha probado ampliamente en pequeñas instancias de Ethereum Classic con 2G de RAM. Esto es suficiente para ejecutar el cliente y el minero.

Para el testnet IELE:

1. Se necesita una máquina virtual Java (JVM) para construir el cliente, como con el testnet KEVM. La versión debe ser 1.8.x. El cliente no ha sido probado con JVM 1.9.
2. Se recomiendan 230 de espacio en disco y 10 MB de RAM.



Contratos Inteligentes



PLUTUS

Plutus: una plataforma de contrato funcional

Plutus proporciona una plataforma de ejecución y un lenguaje de programación funcional que se ejecuta en la capa de liquidación de Cardano y ofrece considerables ventajas de seguridad. Ofrece una manera más fácil y sólida de demostrar que sus contratos inteligentes son correctos y que no encontrarán los problemas encontrados en el diseño anterior del lenguaje de contratos inteligentes.

Al usar Plutus puede confiar en la ejecución correcta de sus contratos inteligentes. Ha sido diseñado con tres audiencias principales en mente: desarrolladores, expertos en negocios y academia.

Plutus está integrado en Haskell, es familiar para los programadores de Haskell, a la vez que es más fácil de analizar que el propio Haskell.



MARLOWE

Marlowe: un lenguaje contractual para el mundo financiero

Marlowe es un nuevo lenguaje para modelar instrumentos financieros como contratos inteligentes en una cadena de bloques. Ha sido diseñado para personas que son ingenieros de negocios o expertos en la materia en lugar de desarrolladores experimentados.

Es un lenguaje simple de dominio específico (DSL) que comprende una pequeña cantidad de poderosos bloques de construcción que pueden ensamblarse en contratos financieros expresivos.

Está incrustado en el lenguaje Haskell, que tiene su propio ecosistema establecido y marco de prueba. No necesita experiencia en programación para usar Marlowe y puede explorar sus construcciones financieras con un editor y simulador de contratos basado en navegador.

Muchas Gracias :)

Contacto

Antonio Sánchez

Antoniofsc87@gmail.com

Linkedin:

<https://www.linkedin.com/in/antonio-blockchain/>

Telegram: @Etherman

Grupo Cardano: @CardanoEsp

