

**Ciencia y
Tecnología**

Secretaría de Ciencia, Humanidades,
Tecnología e Innovación



INAOE
Instituto Nacional de Astrofísica,
Óptica y Electrónica



Maestría en Ciencias y
Tecnologías de Seguridad

Sistema de Verificación Distribuida de Certificados Sanitarios mediante Blockchain

Alumno: Israel Jaudy Pérez Bermúdez

Materia: Sistemas Distribuidos y Cómputo en la Nube

Fecha: junio 27, 2025

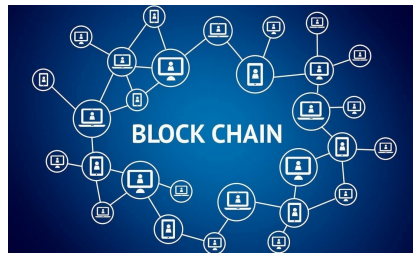
Maestría en: Ciencias y Tecnologías de Seguridad

Instituto: Instituto Nacional de Astrofísica, Óptica y Electrónica

Contenido

- 1 Introducción
- 2 Objetivo
- 3 Arquitectura del Sistema
- 4 Desarrollo
- 5 Resultados
- 6 Conclusiones
- 7 Trabajo Futuro
- 8 Repositorio
- 9 Referencias

- La pandemia global evidenció la necesidad de contar con sistemas digitales seguros para verificar el estado sanitario de las personas.
- Muchos certificados físicos o digitales carecen de mecanismos que garanticen su autenticidad e integridad.
- Blockchain ofrece una solución descentralizada, confiable y resistente a alteraciones, ideal para este tipo de sistemas.



Objetivo del Proyecto

- **Objetivo general:**

Diseñar e implementar un sistema de verificación distribuida de certificados sanitarios mediante tecnología blockchain.

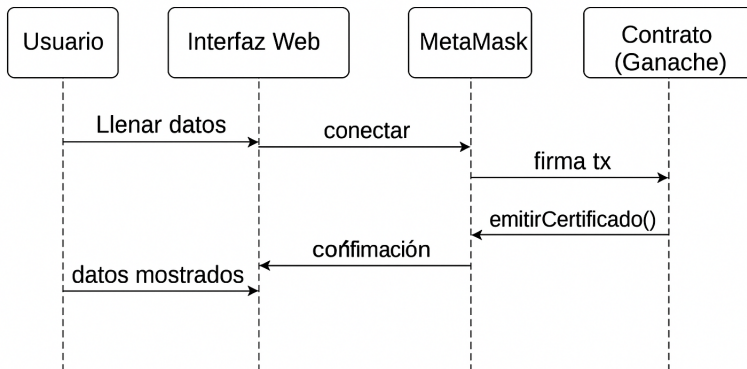
- **Objetivos específicos:**

- Emitir certificados sanitarios digitales que contengan información validada y autenticada mediante hash criptográfico.
- Permitir la verificación y revocación pública de certificados a través de una interfaz web conectada a la blockchain.



- Contrato inteligente en Solidity para registrar, verificar y revocar certificados.
- Frontend web en HTML + JavaScript con conexión mediante ethers.js y MetaMask.
- Red local simulada con Ganache y despliegue a través de Hardhat.
- Script en Python usando web3.py para verificación por línea de comandos.

Diagrama del Sistema



Arquitectura lógica del sistema distribuido con interacción entre contrato inteligente, interfaz web, MetaMask y consola Python.

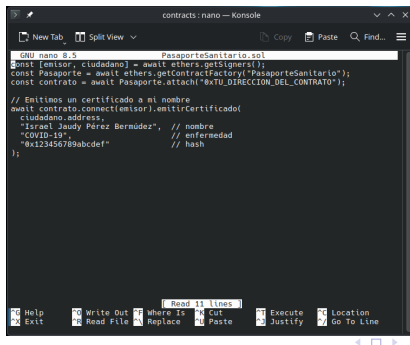
Resumen del Desarrollo

- Contrato inteligente en Solidity con funciones para emitir, verificar y revocar certificados.
- Red local simulada con Ganache y despliegue automatizado con Hardhat.
- Interfaz web conectada a MetaMask mediante la librería ethers.js.
- Script en Python para validación independiente desde consola usando web3.py.



Contrato Inteligente en Solidity

- Desarrollado con Solidity, permite registrar, verificar y revocar certificados sanitarios.
- Asocia el certificado a una dirección Ethereum y un hash de documento.
- Emisión controlada por una autoridad emisora mediante 'emitirCertificado()'.

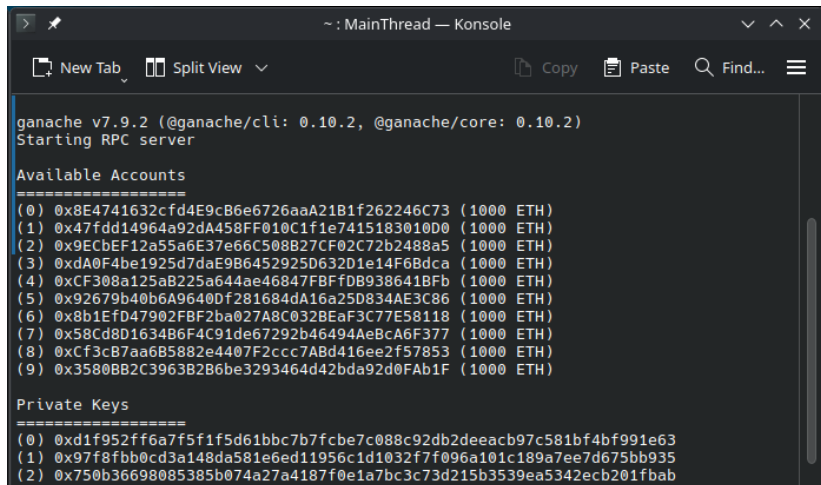


```
contracts: nano — Konsole
GNU nano 8.5 PasaaporteSanitario.sol
const [emisor, ciudadano] = await ethers.getSigners();
const Pasaaporte = await ethers.getContractFactory("PasaaporteSanitario");
const contrato = await Pasaaporte.attach("0xTU_DIRECCION_DEL_CONTRATO");

// Emitimos un certificado a mi nombre
await contrato.connect(emisor).emitirCertificado(
  ciudadano.address,
  "Israel Jaudy Pérez Bermúdez", // nombre
  "COVID-19", // enfermedad
  "0x123456789abcdef" // hash
);
```


Configuración de Red Local con Ganache

- Ganache simula nodos Ethereum con cuentas precargadas.
- Red local usada para pruebas y despliegue con Hardhat.
- Claves privadas y direcciones visibles para depuración.



```
> ~: MainThread — Konsole

New Tab Split View Copy Paste Find...

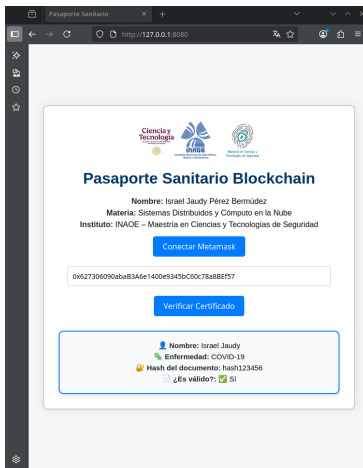
ganache v7.9.2 (@ganache/cli: 0.10.2, @ganache/core: 0.10.2)
Starting RPC server

Available Accounts
=====
(0) 0x8E4741632cfd4E9cB6e6726aaA21B1f262246C73 (1000 ETH)
(1) 0x47fdd14964a92dA458FF010C1f1e7415183010D0 (1000 ETH)
(2) 0x9ECbEF12a55a6E37e66C508B27CF02C72b2488a5 (1000 ETH)
(3) 0xdA0F4be1925d7daE986452925D632D1e14F6Bdca (1000 ETH)
(4) 0xCF308a125aB225a644ae46847FBFfDB938641BFb (1000 ETH)
(5) 0x92679b40b6A9640Df281684dA16a25D834AE3C86 (1000 ETH)
(6) 0x8b1EfD47902FBF2ba027A8C032BEaF3C77E58118 (1000 ETH)
(7) 0x58Cd8D1634B6F4C91de67292b46494AeBcA6F377 (1000 ETH)
(8) 0xCf3cB7aa6B5882e4407F2ccc7ABd416ee2f57853 (1000 ETH)
(9) 0x3580BB2C3963B2B6be3293464d42bda92d0FAB1F (1000 ETH)

Private Keys
=====
(0) 0xd1f952ff6a7f5d1f5d61bbc7b7fcbe7c088c92db2deeac97c581bf4bf991e63
(1) 0x97f8fbb0cd3a148da581e6ed11956c1d1032f7f096a101c189a7ee7d675bb935
(2) 0x750b36698085385b074a27a4187f0e1a7bc3c73d215b3539ea5342ecb201fbab
```

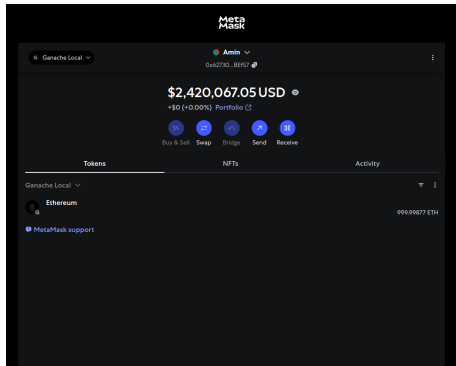
Interfaz Web y MetaMask

- Interfaz desarrollada con HTML, JavaScript y ethers.js.
- Permite conectar MetaMask, emitir y verificar certificados.
- Comunicación directa con el contrato inteligente desplegado.



Visualización de Fondos en MetaMask

- MetaMask conectado a red Ganache local.
- Muestra fondos simulados (1000 ETH).
- Permite firmar transacciones y verificar certificados.



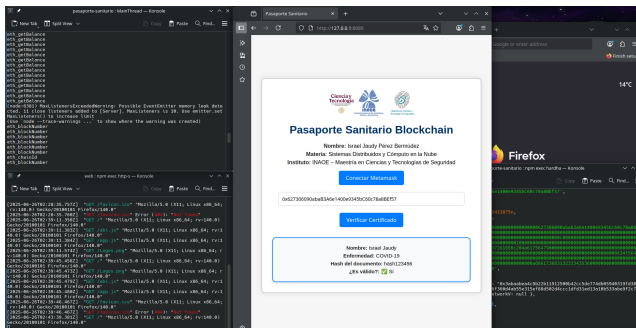
Verificación desde Consola

- Se ejecuta 'contrato.verificarUltimo(...)' para consultar certificado.
- Muestra datos hash, nombre, enfermedad y validez.
- Validación en línea de comandos usando web3.py.

[illegible]

Resultados del Sistema

- El sistema permite emitir certificados sanitarios con hash y validarlos correctamente.
- Se validó la interacción entre los componentes: interfaz web, MetaMask y contrato.
- Se logró la revocación de certificados y la consulta de su validez desde consola.



- Se demostró la viabilidad de un sistema distribuido para la emisión y verificación de certificados sanitarios digitales con tecnología blockchain.
- El uso de hashes asegura la integridad de los documentos sin comprometer la privacidad.
- La integración con herramientas como Ganache, MetaMask y ethers.js permitió simular un entorno seguro, confiable y funcional.
- El sistema es extensible y adaptable a contextos reales donde se requiere control de acceso sanitario.

- Incorporar múltiples perfiles de usuario con permisos diferenciados (ciudadano, autoridad, entidad de salud).
- Integrar almacenamiento descentralizado (IPFS) para enlazar los documentos originales con su hash.
- Agregar firma digital avanzada con certificados emitidos por autoridades de confianza.
- Implementar identidad autosoberana (SSI) mediante identificadores descentralizados (DIDs).
- Migrar el sistema a una testnet pública como Goerli o Sepolia para medir desempeño y costos reales.
- Aplicar revisiones formales de seguridad al contrato inteligente y al frontend.

- Código fuente, contrato inteligente, scripts y documentación disponibles en:
- <https://github.com/NullAstra404/Pasaporte-Sanitario>



Referencias Bibliográficas

- 1 Andreas M. Antonopoulos y Gavin Wood. *Mastering Ethereum: Building Smart Contracts and DApps*. O'Reilly Media, 2022.
- 2 Arvind Narayanan et al. *Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction*. Princeton University Press, 2016.
- 3 Gavin Wood. *Ethereum: A secure decentralised generalised transaction ledger*. Ethereum Foundation, 2014.
<https://ethereum.org/en/whitepaper/>
- 4 Repositorio GitHub del proyecto:
<https://github.com/NullAstra404/Pasaporte-Sanitario>