Sistema Blockchain de Almacenamiento Distribuido para Dispositivos IoT

José Ignacio Bravo Vicente

unir LA UNIVERSIDAD EN INTERNET

ÍNDICE

01 - Introducción
02 - Diseño del Sistema
03 - Implementación
04 - Validación y Pruebas
05 - Conclusiones

Introducción
Antecedentes

Motivación y contexto inicial – Surge de una experiencia laboral: industria farmacéutica con fábricas que utilizan sondas OT / IoT para registrar la telemetría de sus procesos.

- Múltiples sondas capturando datos de telemetría.
- Envío de registros a un servidor central.
- Muy poco volumen de transferencia (~kb).
- A veces se producen latencias y cortes.
- Normativa exigente respecto a la trazabilidad inmutable (GMP, DFA).
- Se apoya en soluciones de almacenamiento y backup en la nube.
- Se producen los cortes en Azure.

Introducción
Estado del Arte

Contexto

- ✓ La tecnología actual está en una profunda transformación impulsada por la convergencia de **Web 3.0**, IoT, arquitecturas distribuidas y tecnologías blockchain.
- ✓ La Web 3.0 propone una internet descentralizada, centrada en el usuario y transparente, devolviéndole el control y la **propiedad de sus datos**.

Problema

- ○ Existe una dependencia crítica de infraestructuras centralizadas (como AWS, Google Cloud o Azure), lo que genera desafíos en términos de privacidad, trazabilidad, disponibilidad y, sobre todo, **soberanía digital.**
- ○ Un fallo en un proveedor centralizado puede provocar **interrupciones masivas** en servicios críticos (Azure).

Propuesta

- ✓ Se presenta el diseño e implementación de una PoC para un **sistema de almacenamiento distribuido** como alternativa para la gestión de archivos en entornos IoT.
- ✓ La solución integra dispositivos IoT, arquitecturas P2P y tecnología blockchain para ser **segura, resiliente, económica y auditable**.

Introducción
Justificación

Comparativa de las soluciones de almacenamiento analizadas:

Solución	Open Source	Blockchain	Seguridad	Orientación IoT
IPFS	✓	✗	No cifrado, no trazabilidad	Parcialmente
STORJ	Parcialmente (algunos componentes clave no)	✓	✓	✗
SIA	✓	✓	Parcialmente (no trazabilidad)	✗
FILECOIN	✓	✓	Parcialmente (no cifrado)	✗
IAGON	✓	✓	✓	✗

Introducción
Objetivos

Objetivo General – Diseñar e implementar una PoC para un **sistema de ficheros distribuido** basado en tecnologías blockchain y P2P que permita al usuario almacenar, compartir y gestionar archivos a través de dispositivos IoT de bajo consumo y coste, garantizando la trazabilidad, disponibilidad y soberanía de la identidad y del dato.

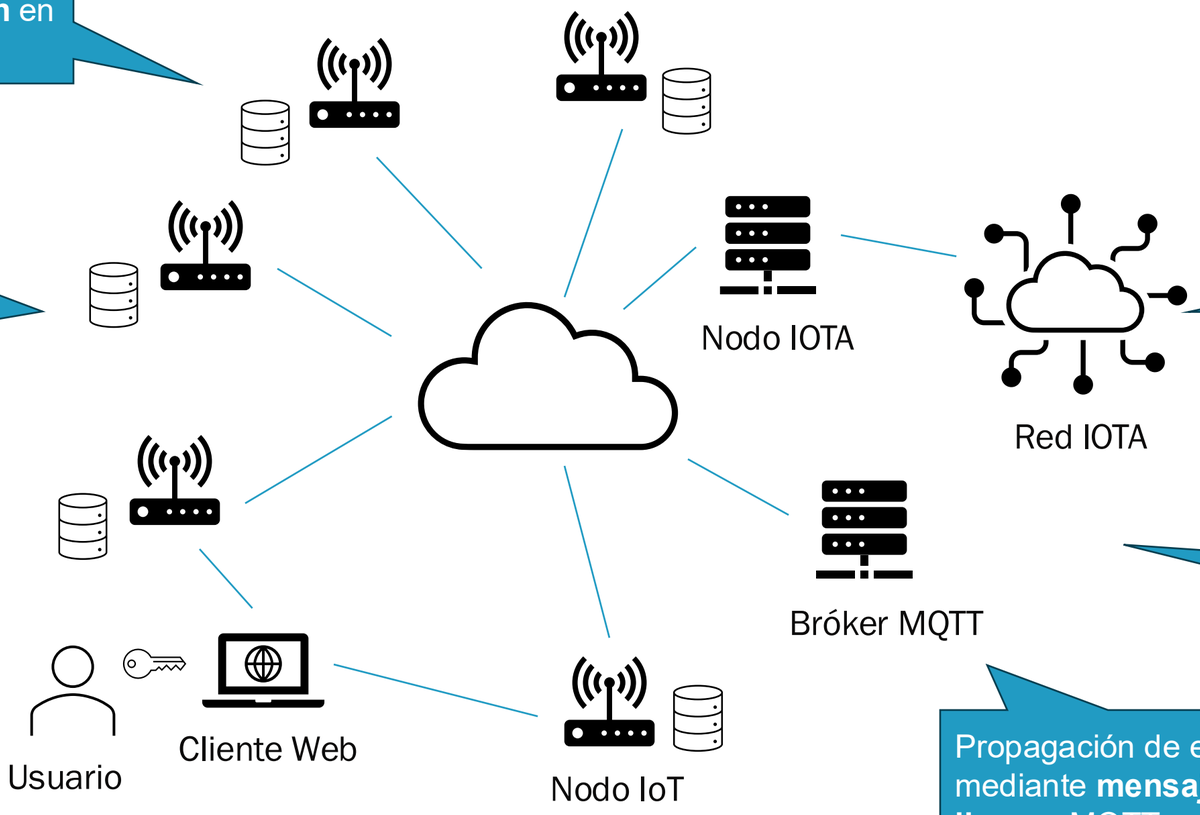
- Investigar y seleccionar tecnologías adecuadas (IoT, P2P, blockchain).
- Diseñar la arquitectura del sistema, incluyendo componentes y modelo de datos.
- Implementar un prototipo funcional sobre dispositivos de bajo consumo (Orange Pi One).
- Evaluar el comportamiento del sistema (disponibilidad, redundancia, integridad).
- Validar la trazabilidad inmutable mediante blockchain.
- Publicar el código bajo licencia GPL.

Diseño del Sistema
Arquitectura

Arquitectura **distribuida y descentralizada** basada en nodos ligeros (IoT)

Usuarios, nodos y archivos se identifican mediante **hashes criptográficos únicos** (hash SHA-256)

Los datos **se almacenan cifrados y se replican** en varios nodos



Tangle de IOTA como **capa de control** para registro de eventos de forma inmutable y auditable

SQLite para datos estructurados (usuarios, nodos, eventos) y **filesystem para ficheros cifrados y metadatos**

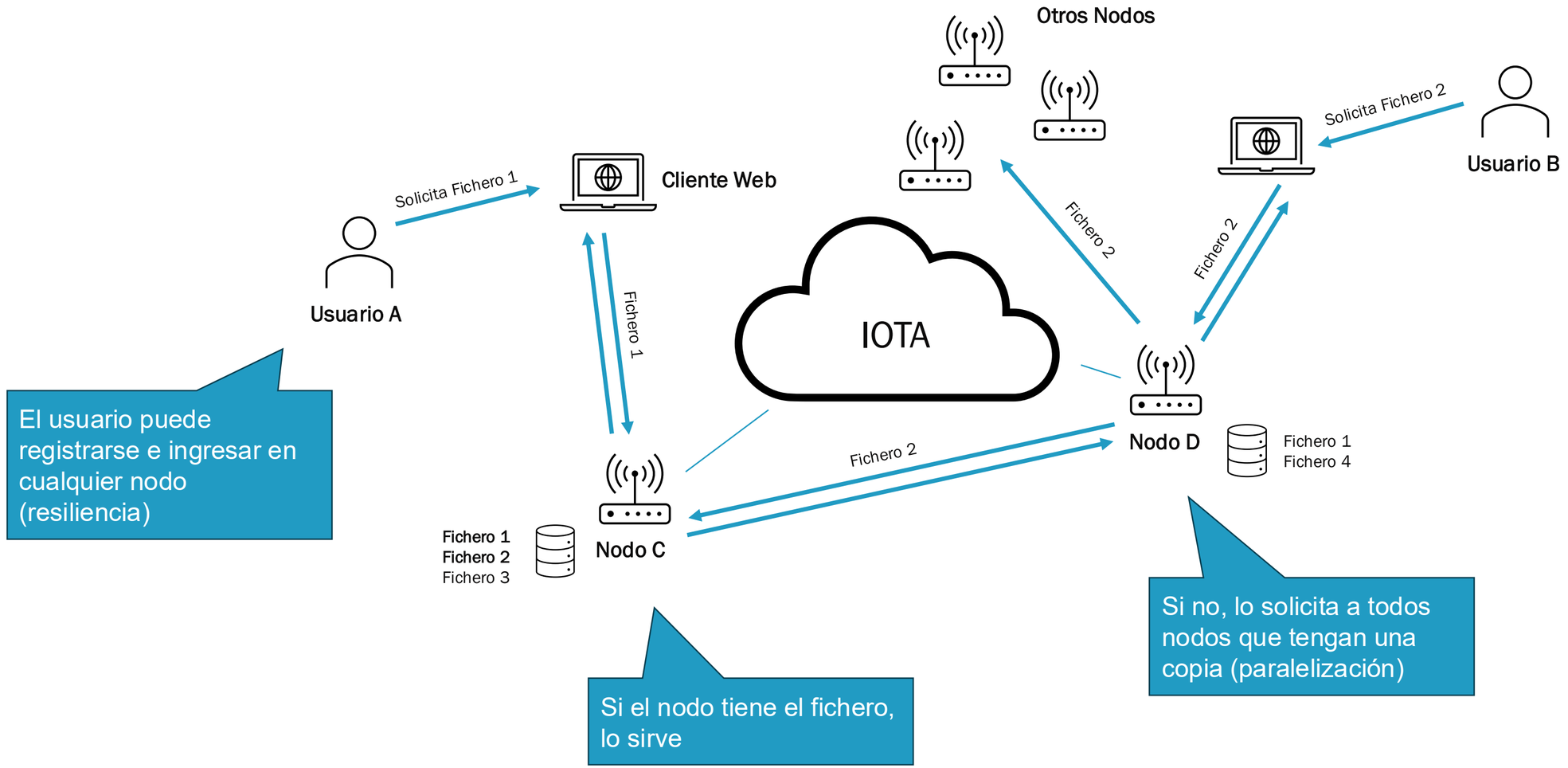
Ficheros cifrados end-to-end con **criptografía híbrida** (AES-256, X25519)

Los eventos se **firman digitalmente** por el nodo emisor para verificar autenticidad e integridad

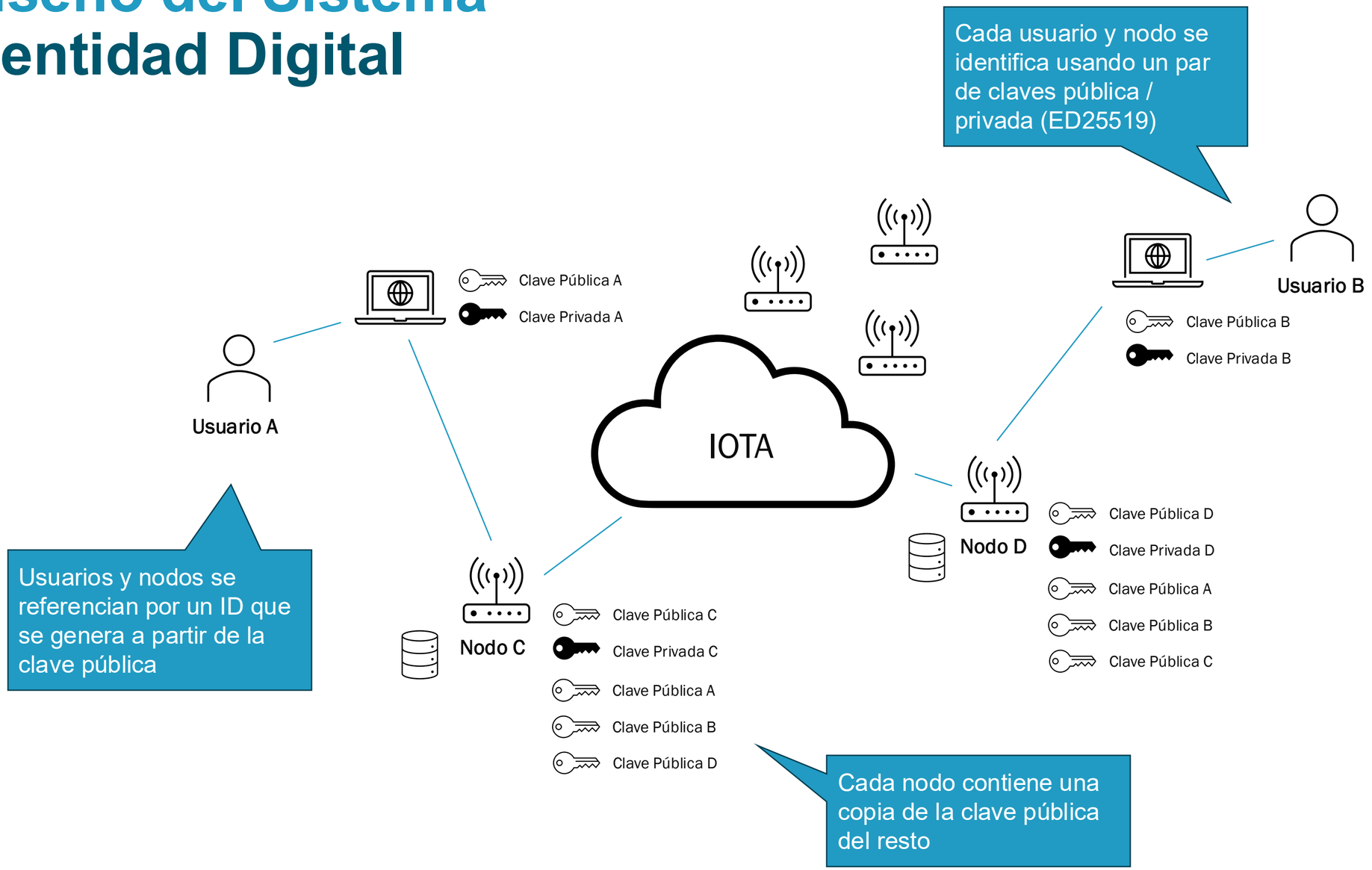
Propagación de eventos mediante **mensajes ligeros** MQTT para notificaciones rápidas de bajo ancho de banda

Cliente HTML5 (AJAX), comunicación API REST, identidad cifrada en LocalStorage (AES-GCM / PBKDF2)

Diseño del Sistema
Compartición de Ficheros



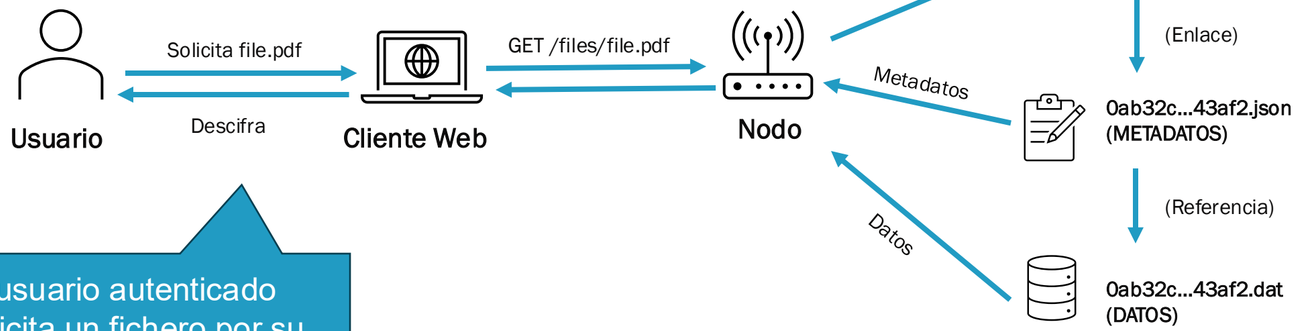
Diseño del Sistema
Identidad Digital



Diseño del Sistema
Sistema Virtual de Ficheros

El filesystem del nodo almacena un enlace con ese nombre (entrada virtual) que apunta a un fichero de metadatos identificado por su ID (hash SHA-256 de su contenido)

Metadatos contienen el tipo de fichero, nodos réplica, información criptográfica, permisos, …



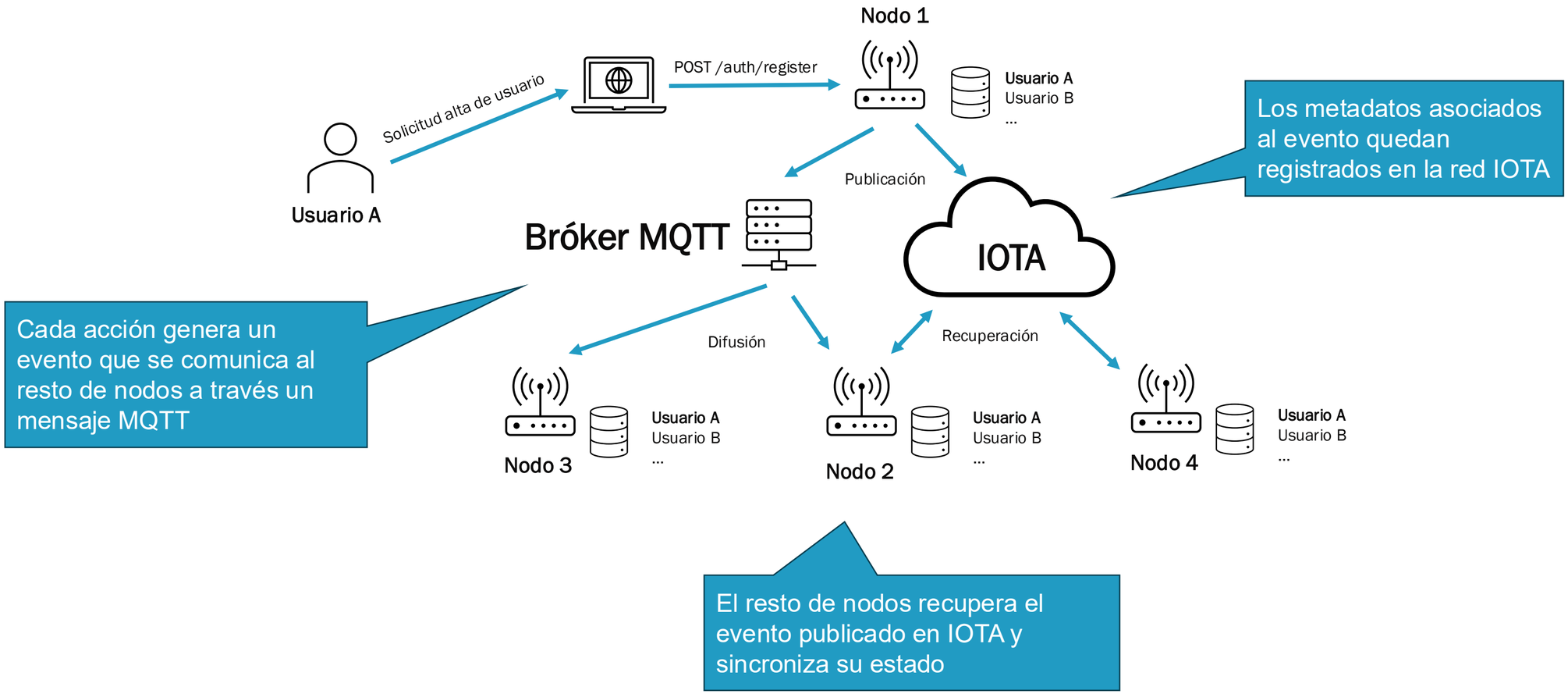
El usuario autenticado solicita un fichero por su nombre

El nodo devuelve los datos cifrados (almacenados en otro fichero) + la información criptográfica necesaria para descifrarlos (cabeceras HTTP)

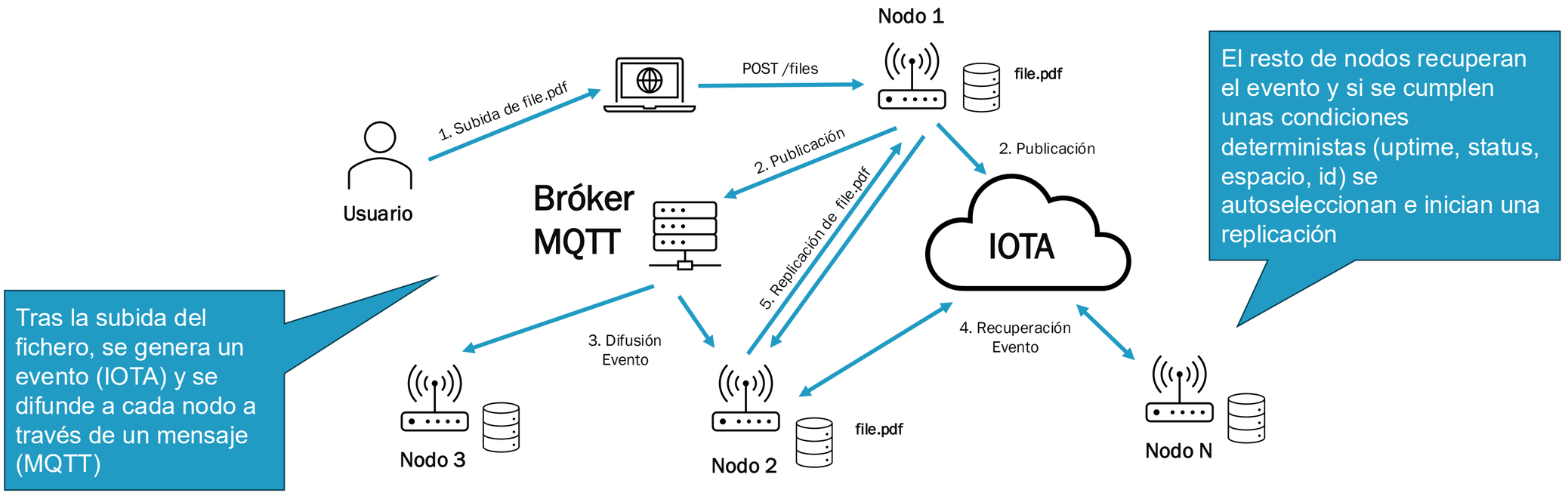
Estructura de fichero (conceptual)

- Metadatos
- Datos Cifrados con clave simétrica (AES-254)
- Clave simétrica cifrada con la clave publica del usuario 1 (X25519)
- Clave simétrica cifrada con la clave publica del usuario N (X25519)

Diseño del Sistema
Sincronización por Eventos



Diseño del Sistema
Replicación de Ficheros



Implementación
Tecnologías Clave

Elección de Tecnologías – Cada componente responde a criterios de rendimiento, compatibilidad con dispositivos IoT de bajo consumo, madurez tecnológica y adecuación al enfoque del proyecto.

- **IOTA Tangle + Docker + Hornet (Testnet)**: eventos inmutables en blockchain.
- **MQTT Mosquitto**: sincronización ligera entre nodos.
- **SQLite + Ext4**: persistencia ligera de datos.
- **Python + FastAPI** + **Uvicorn + Pydantic + PyNacl + Paho-mqtt**: Backend API RESTful y gestor de eventos.
- **Orange Pi One**: hardware IoT.
- **JavaScript + JQuery** + **Bootstrap + Webcrypto + Libsodium**: frontend.
- **Let's Encrypt**: cifrado de comunicaciones TLS.

Implementación
Entorno y Componentes

Infraestructura piloto:

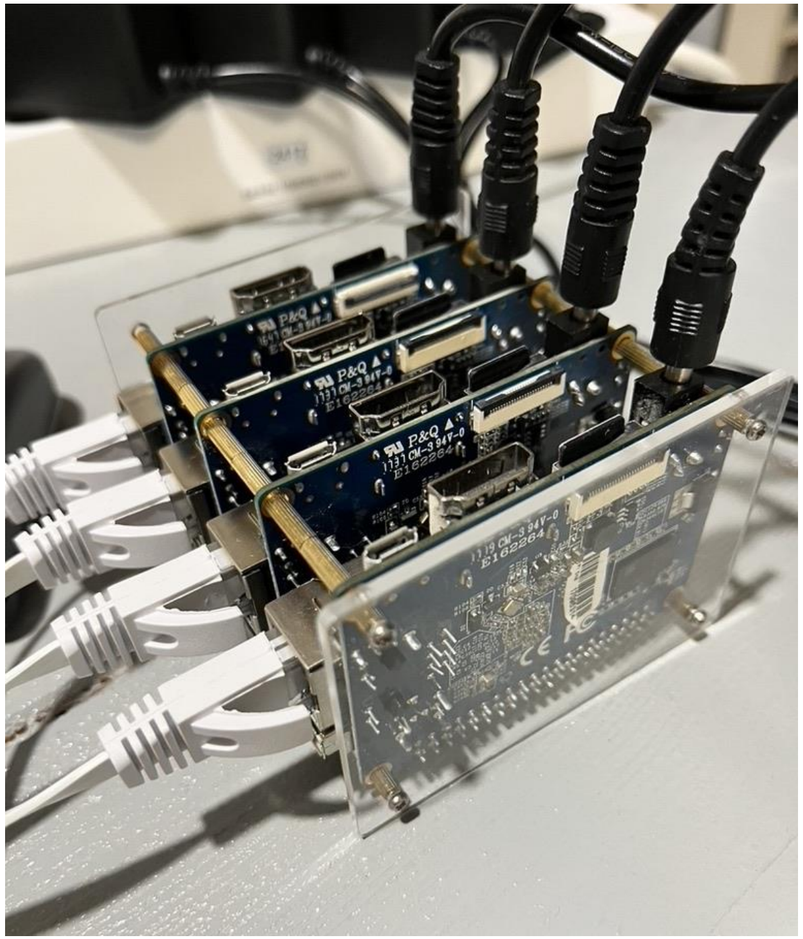
El prototipo se implementó sobre un mini clúster de 4 tarjetas **Orange Pi One** (SoC ARM H3 Quad Core, 512MB RAM, 32GB SD, 10/100M Ethernet) con Armbian IoT Noble 25.2 (Linux v6.12).

Se utilizó un VPS en la nube (1GB RAM, 60GB HD, 1 vCPU Intel 2.60GHz) con Ubuntu Server 22.04 LTS para alojar el nodo0, un Docker Hornet IOTA (Testnet), el servicio MQTT Mosquitto, Servidor Nginx y un DNS dinámico para cada nodo.

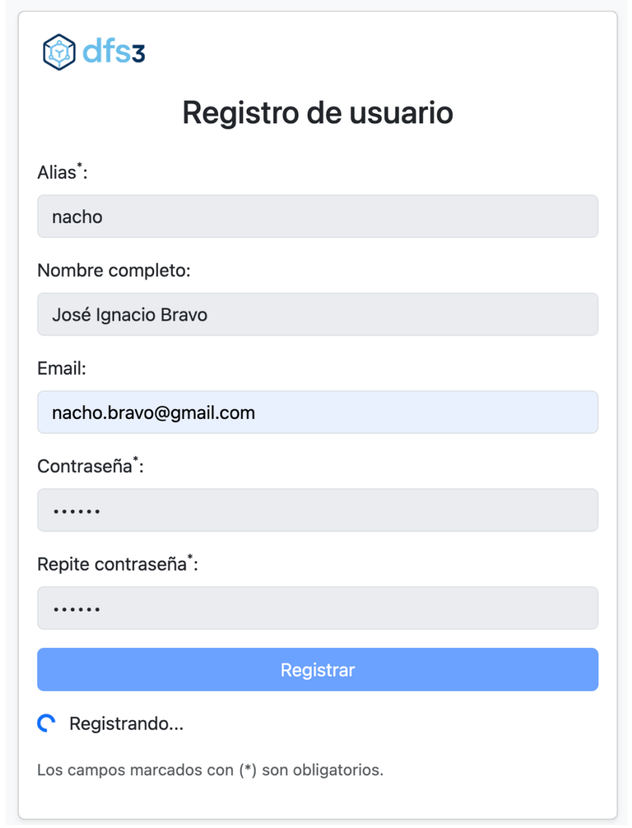
Principales módulos:

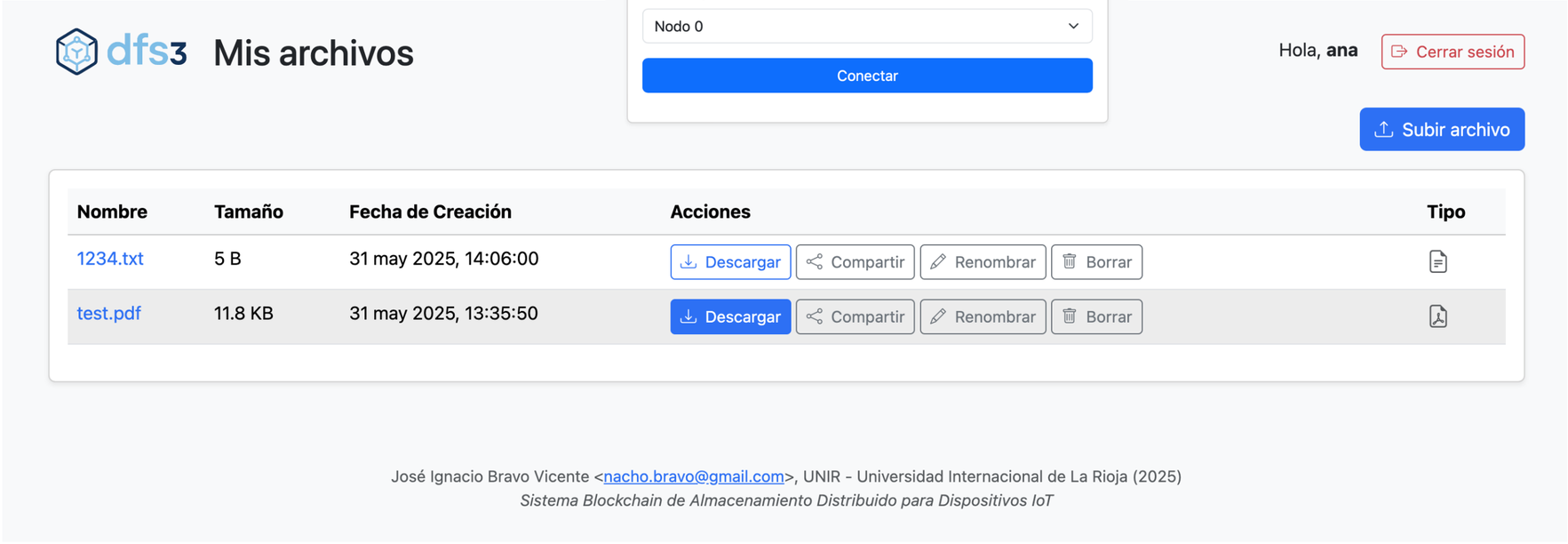
- Nodo IoT (Backend):
  - Gestor de eventos (IOTA + MQTT) para comunicación entre nodos.
  - API REST para interfaz con el cliente web.
  - Módulos de almacenamiento, criptográfico, caché, etc.

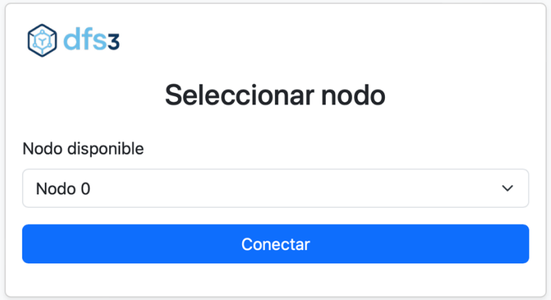
- Cliente Web (Frontend):
  - Interfaz de usuario para registro de usuario y operaciones con ficheros.
  - Lógica criptográfica para cifrado en origen y firma digital.

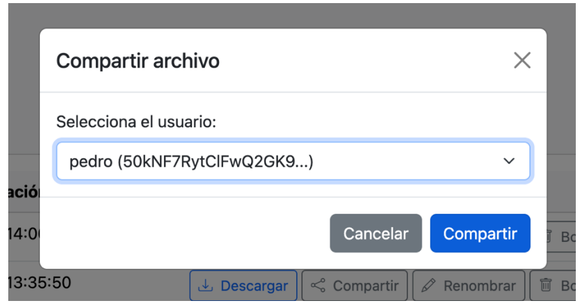


Validación y Pruebas
Casos de Uso



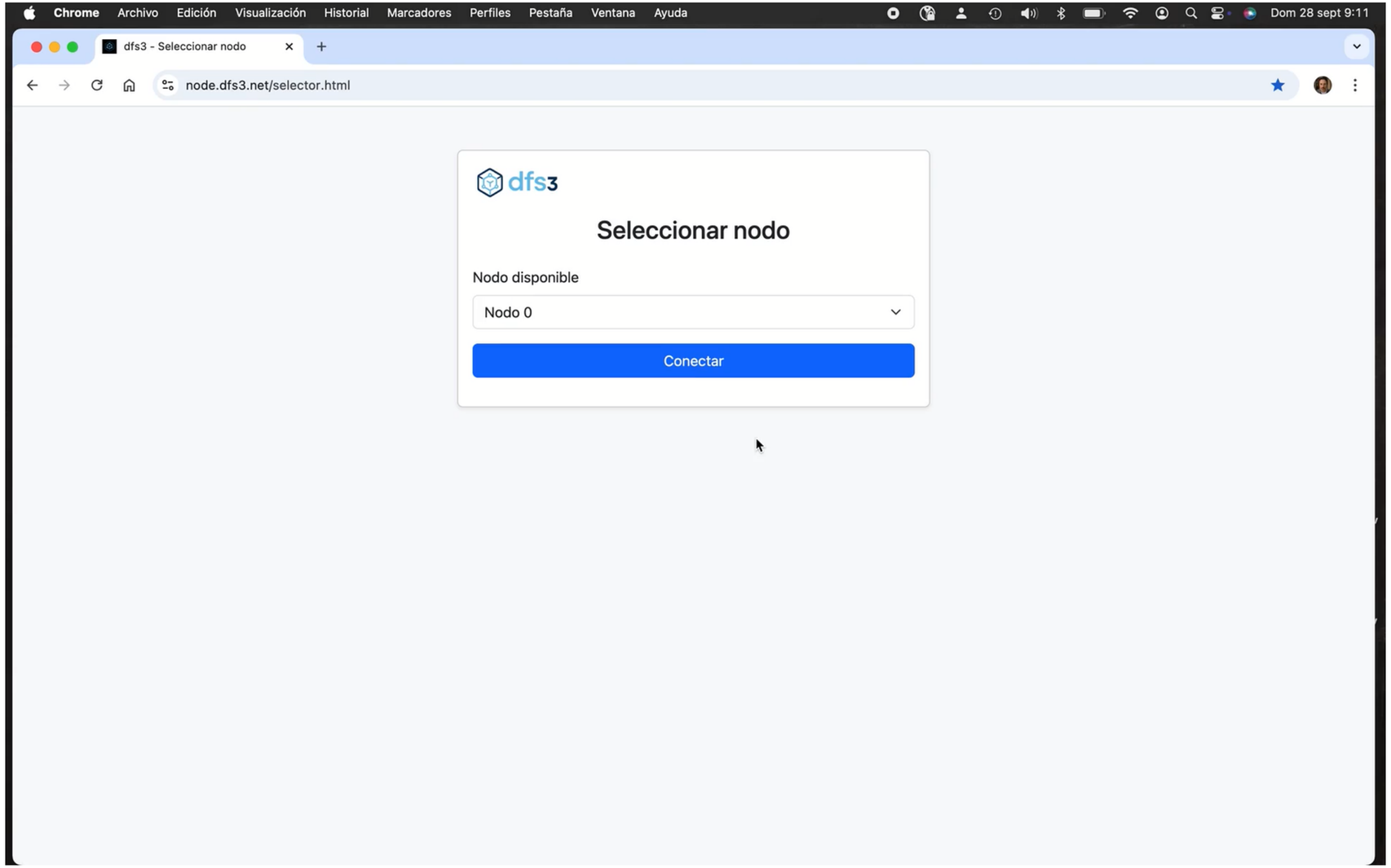






José Ignacio Bravo Vicente <nacho.bravo@gmail.com>, UNIR - Universidad Internacional de La Rioja (2025)
Sistema Blockchain de Almacenamiento Distribuido para Dispositivos IoT

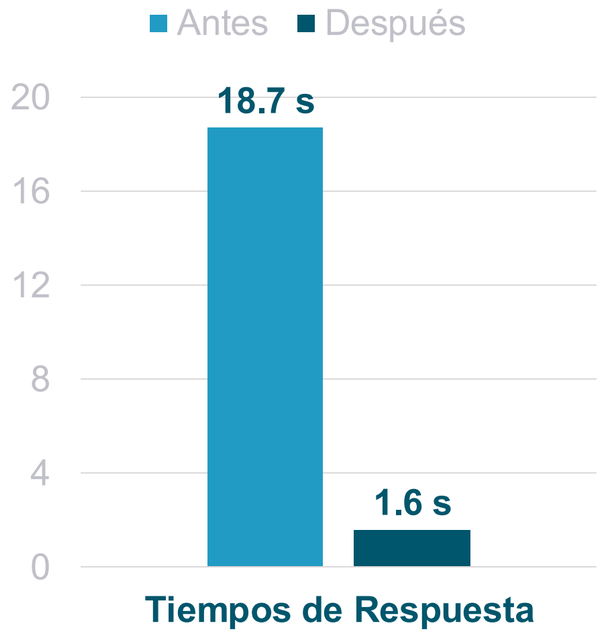
Validación y Pruebas
Demo



Validación y Pruebas
Resultados

Comportamiento general – El sistema demostró ser viable y capaz de mantener un comportamiento coherente y estable ante cargas de trabajo moderadas en dispositivos con recursos limitados.

- **Limitaciones** y cuellos de botella propios del hardware IoT (Fast Ethernet), alta latencia de la red IOTA.
- **Optimizaciones**: modo streaming para evitar cargar archivos completos en memoria, envío de eventos asíncronos en paralelo para minimizar tiempos en la respuesta, etc.
- **Caída de nodos replicadores**: se simuló la caída de nodos que almacenaban copias de archivos. El sistema demostró que los archivos siguen siendo accesibles tras la desde nodos alternativos gracias a las réplicas redundantes.
- **Modelo de consistencia**: la adopción de la consistencia eventual (propagación progresiva de cambios mediante eventos firmados e inmutables) proporcionó robustez ante pérdidas temporales de mensajes o desconexiones
- **Pruebas de Seguridad**: se validó la inmutabilidad de los eventos con la verificación de firmas digitales, el cifrado de archivos en origen y la denegación de acceso a usuarios no autorizados.



~1,6 s

Tiempo medio de subida / descarga

Ficheros de 10mb

< 40%

Consumo de CPU con carga

20 peticiones concurrentes

124 ms

Tiempo medio de respuesta

Operaciones GET API

Conclusiones
Líneas de Mejora y Trabajos Futuros

Conclusiones

- ✓ Demostrada la **viabilidad técnica** de la idea inicial.
- ✓ El resultado destaca por su **naturaleza híbrida** (software + sistemas embebidos).
- ✓ Alineación con los principios **Web 3.0,** identidad digital descentralizada y soberanía del dato.

Retos y Desafíos

- ○ Implementación compleja debido a su **capa criptográfica** avanzada.
- ○ La **falta de madurez** y estabilidad de las herramientas y librerías IOTA.
- ○ La **limitación de recursos hardware** ha obligado a poner el foco en la optimización continua.

Mejoras Técnicas

- ✓ Migrar a **lenguajes más eficiente**s (C/C++, Rust o Go) para reducir el consumo de recursos.
- ✓ Evolucionar la replicación hacia el uso de **Erasure Coding** (Reed-Solomon) para optimización de espacio y transferencias más eficientes.
- ✓ Integrar la **autenticación mediante JWT** con visado para mejorar la escalabilidad.

Evolución Futura

- ○ Modelo económico con **incentivos basados en tokens** (MIOTA) para recompensar la participación de los nodos.
- ○ Evolucionar para que los nodos también ejecuten tareas distribuidas de procesamiento, alineado con el modelo de computación en la niebla o **Fog Computing**

muchas gracias