



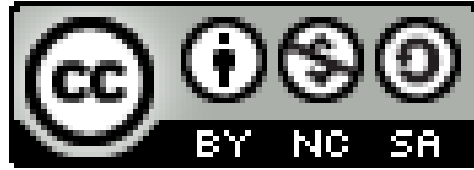
INSTITUTO TECNOLÓGICO DE COSTA RICA
ESCUELA DE INGENIERÍA EN COMPUTACIÓN

“Proyecto Antelope Tools: Sensor de salud de nodos para redes blockchain
basadas en Antelope + EOSIO”

Proyecto Final de Graduación para optar por el título de
Bachillerato en Ingeniería en Computación

Helbertson Torres Mora

San José, noviembre, 2022



Esta obra está bajo una [licencia de Creative Commons Reconocimiento-NoComercial-CompartirIgual 4.0 Internacional](https://creativecommons.org/licenses/by-nc-sa/4.0/).

RESUMEN

Las redes blockchain basadas en el protocolo EOSIO dependen del mecanismo de consenso de Delegated Proof of Stake (DPoS), donde mediante votaciones se selecciona a los productores que validarán los bloques y recibirán tokens como recompensa. Por esta razón, es de gran importancia la búsqueda por aumentar la transparencia de estas redes blockchain y dar a conocer la legitimidad de las organizaciones productoras de bloques.

Ahora bien, el proyecto Antelope Tools Dashboard, desarrollado por la empresa Edenia monitorea el estado de algunas de estas redes y muestra información relevante de las entidades que las conforman, no obstante, un aspecto importante como el conocer si los nodos están funcionando está ausente, es decir, no se mostraba en dicho sistema un sensor de salud de los endpoints de los nodos.

Por este motivo durante la práctica, se implementó una solución que consulta de manera constante los API endpoints, mediante el protocolo HTTP, para determinar el estado de salud de los nodos respecto al estado de la red blockchain. Además, ya que, para obtener la información se necesita consultar las tablas públicas directamente de la blockchain y luego obtener un archivo que brindan los productores se construyó un API para ofrecer los datos procesados y evitar desarrollar nuevamente la misma lógica en otros sistemas.

Asimismo, para recuperar datos de la blockchain se consulta mediante el endpoint de un nodo, pero, si este no está saludable, los datos no se recuperan, por lo que, aprovechando que existen múltiples nodos en una red, se implementó el uso de una lista de endpoints lo que mejorará la disponibilidad de los datos, ya que tendrían que fallar todos los endpoints de la lista para que no se puedan obtener. También, se efectuaron cambios en la interfaz, prueba, reparación de errores y se incrementó la información desplegada de los productores de bloques.

Como resultado, se logró mejorar el sistema para que presente información valiosa de los productores de bloques y sus nodos, lo anterior permite que el acceso a estos datos sea más sencillo y si los participantes usan la herramienta puedan efectuar un mejor juicio para votar por aquellos productores que demuestren un compromiso con la red.

Palabras claves: Antelope, EOSIO, Blockchain, sensor de salud, nodos, BP JSON, API endpoints

ABSTRACT

The Blockchain networks based on the EOSIO protocol depend on the consensus mechanism called Delegated Proof of Stake (DPos), where voting is used to select block producers that will validate blocks and receive tokens as rewards. For this reason, the quest to increase the transparency of these blockchain networks and make the legitimacy of the block-producing organizations known is of great importance.

Now, the Antelope Tools Dashboard system, developed by Edenia, monitors the state of some of these networks and displays relevant information about the entities that compose them, however, an essential aspect, such as knowing which nodes are working is not present, which means that there was no health sensor for the node's endpoints.

Consequently, during the internship a solution was implemented that queries constantly the API endpoints, through HTTP protocol, to determine the health of the nodes concerning the blockchain state. In addition, since, to obtain the information, it is necessary to query the public tables directly from the blockchain and then get a file provided by the producers, an API was built to offer the processed data and avoid redeveloping the logic in other systems.

Likewise, to retrieve data from the blockchain, it is queried by a node's endpoints, but if it is not working, the data is not obtained, for this reason, taking advantage of that the blockchain has multiples nodes, it was implemented the use of an endpoints list, this would improve the availability of the data since all the endpoints in the list would have to fail to be unobtainable. Also, changes were made to the UI/UX, testing, bug fixes, and increased information displayed from the block producers.

As a result, it was achieved to improve the system by showing valuable information about block producers and their nodes, the previous allows access to the data easier, and those participants that use the tool can make a better judgment to vote for those block producers who demonstrate a commitment to the network.

Key words: Antelope, EOSIO, Blockchain, health check, nodes, BP JSON, API endpoints

Tabla de Contenido

1.	Introducción	7
2.	Contexto del proyecto.	7
2.1.	Descripción general de la empresa	7
2.2.	Antecedentes del proyecto	9
2.2.1.	Descripción de EOSIO Blockchain	9
2.2.2.	Detalles de Antelope Tools	11
2.2.3.	Cambio de EOSIO Dashboard a Antelope Tools	13
3.	Descripción del problema	14
3.1.	Problema	14
3.2.	Solución	14
3.3.	Stakeholders	15
3.4.	Necesidades y expectativas	17
3.5.	Requerimientos no funcionales	18
3.6.	Perspectiva, supuestos y dependencias del producto	19
3.7.	Características generales	20
4.	Análisis de riesgos	20
5.	Objetivos y Alcance del sistema	24
5.1.	Objetivo general	24
5.2.	Objetivos específicos	24
5.3.	Alcance	25
6.	Modelo de Diseño	25
6.1.	Arquitectura del sistema	25
6.1.1.	Patrón demux	26
6.1.2.	Implementación del patrón	27
6.2.	Modelos de subsistemas	28
6.3.	Interfaces de usuario	30
6.4.	Componentes y servicios	40
6.5.	Diseño de bases de datos	44
7.	Conclusiones	47

7.1.	Cumplimiento de los objetivos	47
7.2.	Productos entregados	48
7.3.	Conclusiones del proyecto	49
7.4.	Recomendaciones	50
7.5.	Trabajo a futuro	50
7.6.	Experiencias	51
8.	Glosario de términos	51
9.	Referencias	54

Índice de Figuras

Figura 1.	Organigrama de Sistemas Edenia Internacional S.A.	9
Figura 2.	Mapa mental de expectativas	17
Figura 3.	Patrón arquitectónico demux. (EOS Costa Rica, 2022)	27
Figura 4.	Arquitectura del sistema. (EOS Costa Rica, 2020)	28
Figura 5.	Diagrama de paquetes.	29
Figura 6.	Página principal antes del cambio.	30
Figura 7.	Página principal con el menú lateral escondido.	31
Figura 8.	Página principal con el nuevo menú lateral.	31
Figura 9.	Página de productores de bloques antes del cambio.	32
Figura 10.	Página de productores de bloques después del cambio.	33
Figura 11.	Búsqueda sin resultados.	33
Figura 12.	Página del generador antes del cambio.	34
Figura 13.	Página del generador después del cambio.	35
Figura 14.	Página con la lista de endpoints antes del cambio.	36
Figura 15.	Página de endpoints con sensor de salud.	36
Figura 16.	Página de endpoints con filtrado de nodos saludables.	37
Figura 17.	Ventana de ayuda con el significado del sensor de salud.	37
Figura 18.	Diálogo con la información del endpoint.	38
Figura 19.	Página de nodos antes del cambio.	39
Figura 20.	Página de nodos después del cambio.	39
Figura 21.	Diagrama de actividad. Efectuar una consulta mediante el API de eosjs.	41
Figura 22.	Ejemplo de objeto chains.json.	42
Figura 23.	Diagrama de flujo. Estado de salud del endpoint.	43
Figura 24.	Diagrama de actividad con el servicio de producers.	44
Figura 25.	Modelo Entidad-Relación de la base de datos.	46
Figura 26.	Modelo lógico de la base de datos.	46

Índice de Tablas

Tabla 1. Stakeholders del Proyecto	16
Tabla 2. Requerimientos no funcionales	19
Tabla 3. Cumplimiento de objetivos	48
Tabla 4. Productos entregados	49

1. Introducción

En el presente documento abarca la información del proyecto de práctica profesional con la empresa Edenia para realizar un aumento de funcionalidad al sistema Antelope Tools con el fin de brindar más datos de los diferentes productores de bloques de las redes blockchain, con protocolo EOSIO, en búsqueda de fortalecer la fiabilidad, confianza y transparencia de dichas redes.

Del mismo modo, se presenta tanto el quehacer de la empresa como el contexto, descripción del problema, stakeholders y objetivos del proyecto, asimismo, la definición de la administración de riesgos, la solución desarrollada para la implementación del sensor de salud de los nodos y, por último, el cumplimiento de los objetivos y las conclusiones.

2. Contexto del proyecto.

2.1. Descripción general de la empresa

Para comenzar, la empresa Sistemas Edenia Internacional S.A., como se indica en Edenia (1996), se dedica al desarrollo de soluciones de software y correr la infraestructura necesaria en blockchain. Según Edenia (1996), la empresa se divide en tres marcas operativas: Edenia Labs, Costa Rica Servers y EOS Costa Rica.

Por un lado, en Edenia Labs diseñan y desarrollan soluciones y aplicaciones web3 innovadoras de primer nivel usando lo último en tecnologías de la industria 4.0.

Por otro lado, en Costa Rica Servers gestionan servidores bare metal ecoeficientes para ofrecer colocación, hospedaje web y servicios de configuración de red con monitoreo y disponibilidad 24/7.

Y, por último, EOS Costa Rica donde desarrollan software open-source, soluciones empresariales, herramientas para infraestructura y recursos educativos para desarrolladores usando el protocolo EOSIO, esto con el fin de promover el desarrollo de aplicaciones descentralizadas que facilitan la soberanía individual, de organizaciones y empresas. Además, proveen una plataforma abierta para desarrolladores con el fin de incrementar las oportunidades y capacitación de la comunidad blockchain.

Ahora bien, para el período de práctica profesional se trabajó en la división de EOS Costa Rica, según EOS Costa Rica (2021a) ofrecen como servicio el diseñar soluciones para empresas globales de gran tamaño y PYMES, entidades de gobierno, organizaciones sin fines de lucro, y proyectos de código abierto utilizando la tecnología blockchain en como sectores como el financiero, cadena de suministros, videojuegos y salud. Recuperado de EOS Costa Rica (2021a) entre sus últimos proyectos se pueden mencionar los siguientes:

- ***EOS Rate:*** una aplicación de código abierto que permite a titulares de tokens EOS acceder a un sistema de calificaciones y portal de votación para productores de bloques y proxies en la blockchain EOS.
- ***Antelope Tools:*** un proyecto de código abierto que permite a usuarios visualizar la distribución de recompensas en las redes Antelope + EOSIO en la búsqueda de rendición de cuentas y transparencia. Este es el proyecto en el que se trabajó el aumentó de funcionalidad.
- ***Smart EIR:*** una plataforma basada en blockchain que brinda soluciones para entradas y salidas de puertos para la industria de logística de contenedores y transmisión en tiempo real de datos requeridos para el intercambio de carga.
- ***Proyecto Empresarial Privado:*** una aplicación para un cliente estadounidense en la industria de contabilidad y finanzas para mejorar la trazabilidad de transacciones Inter compañía.

Dentro del equipo en San José, Costa Rica de EOS Costa Rica, ver organigrama en la Figura 1. Organigrama de Sistemas Edenia Internacional S.A., el practicante fue responsable de diseñar y desarrollar software para el proyecto Antelope Tools, antes denominado como EOSIO Dashboard, y de coordinar con el resto del equipo, teniendo un rol de desarrollador Full stack.

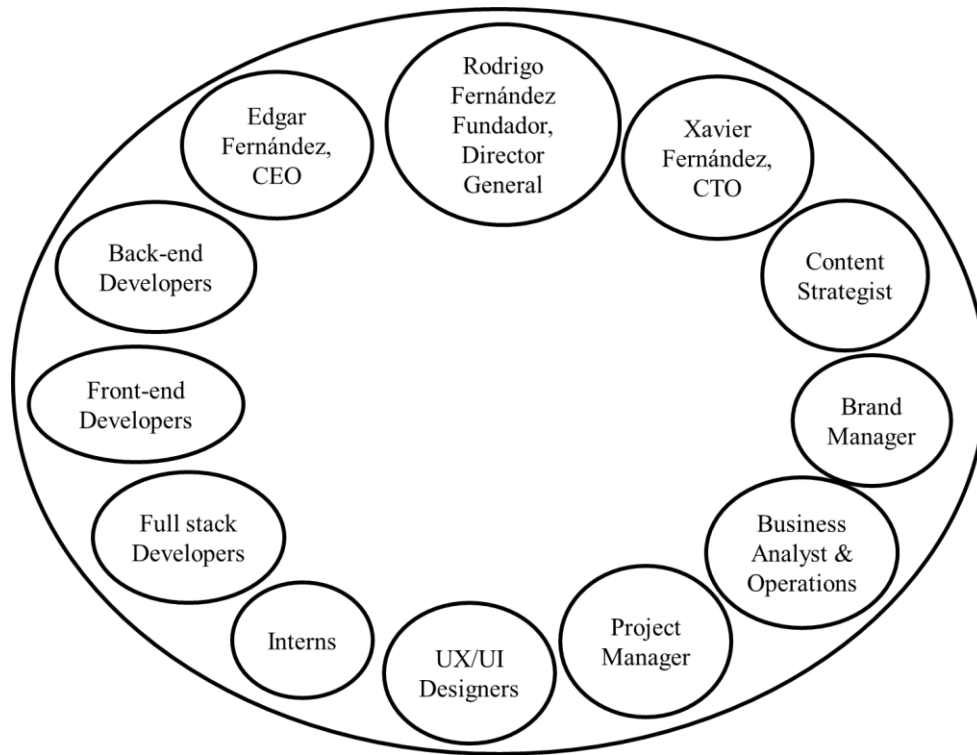


Figura 1. Organigrama de Sistemas Edenia Internacional S.A.

2.2. Antecedentes del proyecto

2.2.1. Descripción de EOSIO Blockchain

Primero, es necesario conocer generalmente el contexto de la tecnología, EOSIO es un protocolo de blockchain que tal como lo indica EOS Costa Rica (2021a) menciona que una blockchain registra transacciones en orden cronológico, donde los registros son inmutables, permitiendo así la trazabilidad completa de transacciones en tiempo real. Del mismo modo, como indican en EOS Costa Rica (2021a) EOSIO fue creado en 2018 por la compañía Block One (B1) y es un software de código abierto para la creación de redes blockchain que cuentan con el token nativo EOS.

Es importante diferenciar EOSIO de EOS, por un lado, EOSIO es el software que permite la ejecución de aplicaciones blockchain descentralizadas, en cambio, EOS es una red blockchain basada en EOSIO, es decir, es un ejemplo de cómo usar el protocolo, porque EOSIO permite la construcción de redes blockchain personalizadas.

En una red blockchain las transacciones se agrupan en bloques que son encadenados por el hash criptográfico del bloque anterior, estos bloques deben ser validados antes de ser considerados inmutables, pero, el método utilizado en cada red blockchain podría variar, como se expresa en EOS Costa Rica (2022) en una red blockchain EOSIO se usa el Delegated Proof of Stake (DPoS) como su mecanismo de consenso el cual determina cuales productores de bloques serán los encargados de validar las transacciones para crear un bloque. Esto se realiza mediante una votación donde los poseedores de tokens, es decir, quienes tienen criptomonedas en la red, pueden votar por cualquiera de los productores de bloques en la red.

Los productores de bloques son equipos u organizaciones que pueden operar uno o más nodos, además, los que se encuentren entre los primeros 21 productores con más votos se encargan de la validación de bloques. Como automáticamente cada cierto tiempo se efectúan las votaciones los productores de bloques que están en el Top pueden cambiar.

Al incluir votaciones como mecanismo de consenso es necesaria la participación de la comunidad, ya que, esta tiene gran impacto en la red blockchain, porque el DPoS permite el acuerdo para la validación de los bloques y si las votaciones evitan que se concentre el poder sobre la red en un solo grupo se asegura que la red continúe, ya que, se evita un ataque del 51%.

Además, es importante conocer el concepto de estado de salud de un nodo, el cual hace referencia a que los nodos pueden encontrarse en tres estados distintos según el estado de la blockchain, esto se debe a que cada nodo mantiene una copia del estado de la red, que puede ser completa si es almacenado desde el primer bloque de la red.

Ahora bien, el primer estado se manifiesta si tanto el nodo como la blockchain tiene el mismo estado, entonces, el estado de salud de dicho nodo es “sincronizado”. El segundo, se da cuando el estado del nodo está atrasado según los bloques reversibles, aquellos que todavía no son inmutables. Y, el tercero a cuando se encuentra atrasado según los bloques irreversibles o inmutables.

Del mismo modo, se tiene el concepto de transparencia de la red, que hace referencia a que en la red las acciones que se estén realizando sean honestas y se brinde la rendición de cuentas correspondientes por parte de productores de bloques. Para que haya un incremento en la transparencia de la red es

necesario que los participantes tengan acceso a información relevante para una mejor toma de decisiones.

2.2.2. Detalles de Antelope Tools

El sistema Antelope Tools, antes se llamaba EOSIO Dashboard, pero, por motivos que se detallan en la siguiente sección fue cambiado. Dicho sistema ya existía antes de la práctica, fue desarrollado al identificar que no existían soluciones de código abierto para visualizar información confiable sobre los nodos en las redes EOSIO y su actividad en tiempo real, así como los diferentes productores de bloques, total de nodos, etc. Esto con el fin de que la comunidad pueda tener dicha información al alcance.

El sitio se encuentra disponible bajo la dirección <https://antelope.tools/>, como indica Edenia (2020), y al ser una herramienta open-source el código fuente se encuentre disponible de manera pública en GitHub en el repositorio de Edenia.

Antelope Tools es un explorador de bloques y monitor red, que facilita la visualización de datos relevantes acerca de los nodos, productores de bloques y la distribución de recompensas en la red blockchain que siguen el protocolo Antelope o EOSIO. Es descrito como un Monitor de Infraestructura EOSIO y un tablero de nodos. Tal como se indica en EOS Costa Rica (2021b) en la plataforma se visualiza la información relevante de las redes blockchain desarrolladas con tecnología EOSIO, tales como: LACChain EOSIO, Proton Mainnet, Telos Mainnet, Wax Mainnet, Jungle Testnet, y EOS Mainnet.

Según EOS Costa Rica (2021b) entre las funciones que tenía el sistema antes de la práctica se encuentran las siguientes:

- El monitoreo de distintas redes blockchain basadas en EOSIO tales como EOS Mainnet, LACChain EOSIO, Proton Mainnet, Telos Mainnet, Wax Mainnet, Jungle Testnet, entre otras.
- Datos en tiempo real de transacciones y bloques, y un gráfico que muestra la información de las transacciones por segundo o por bloque.

- Una lista de productores de bloques o entidades participantes (en el caso de LACChain EOSIO) y datos de la distribución de recompensas de BP como las recompensas diarias, la ubicación de los productores pagados y los tipos de cambio.
- Un directorio de los nodos de cada red con información relevante como la ubicación, la disponibilidad de los puntos finales y la salud de la red.
- Características y herramientas específicas de la red, incluyendo herramientas para crear acuerdos de productores de bloques, plantillas de archivos JSON de BP y acciones ejecutables para contratos inteligentes.

Por otro lado, EOS Costa Rica (2021b) indica que Antelope Tools no es el primer sistema que han realizado para promover la transparencia de la blockchain EOSIO, debido a que, el primero es EOS Rate, que corresponde a una herramienta de código abierto para que los poseedores de tokens EOS califiquen a los productores de bloques y capten la opinión colectiva global.

Como destaca EOS Costa Rica (2021a) entre proyectos similares se pueden encontrar los siguientes exploradores de bloques EOS Network Monitor, Bloks.io, EOS Authority.

Por un lado, EOS Network Monitor es una herramienta para comprobar endpoints públicos de la red EOS y mostrar información general utilizando el archivo BP JSON, sin embargo, a diferencia de Antelope Tools, el sistema EOS Network Monitor solo muestra los datos de la red EOS y Jungle, además, que incluye una funcionalidad similar al sensor de salud que se requería desarrollar para Antelope Tools.

Para obtener la información cada productor de bloques debe brindar un objeto de tipo JSON conocido como BP JSON que es definido como un estándar, este objeto debería contener datos como el nombre de la cuenta del productor, nombre del candidato, página web, redes sociales, logos, entre otros. Igualmente, debe incluir los diferentes API endpoints del productor de bloques. Esto porque dentro de la red se tienen distintos tipos de nodo, que corresponden a equipos físicos con algún propósito en específico, y cada nodo puede ofrecer diversos servicios mediante sus API's dependiendo de su configuración. No obstante, el BP JSON puede que no contenga información requerida o el formato es incorrecto, ya que, no todos los productores de bloques siguen el estándar.

Los participantes en las redes blockchain EOSIO que tienen tokens EOS pueden votar por los productores de bloques que creen que son los mejores candidatos, para realizar la validación de las transacciones de la red. Por lo tanto, monitorear lo que sucede en la red es esencial para mejorar la transparencia de las redes.

Como fue mencionado EOSIO despliega un mecanismo de consenso llamado Delegated Proof of Stake (DPoS). Este mecanismo funciona de forma autónoma y aprovecha un sistema de votación para elegir a los veintiún productores de bloques que dirigen la red. Y por ello, reciben como recompensa tokens.

Es por ello, que es importante que la información de los productores de bloques se encuentre disponible y sea de fácil acceso para los participantes de las redes blockchain EOSIO.

2.2.3. Cambio de EOSIO Dashboard a Antelope Tools

Antelope Tools es la nueva marca de un sistema denominado EOSIO Dashboard, como se citó la empresa Block One (B1) creó el protocolo EOSIO, y esta compañía recibía fondos para brindar mantenimiento al software entregando nuevas versiones, sin embargo, para la versión 3 se incluyeron cambios que según la comunidad de estas redes, se alejaba de sus intereses.

Por esta razón, la comunidad decidió dejar de entregar los fondos a B1 y realizar una bifurcación del protocolo, llamándolo Antelope. Así que, de una manera simplificada este es el motivo del cambio de nombre del sistema. También, porque el sistema no solo es un Dashboard, sino que también tiene funcionalidad de monitor de bloques y otras secciones como el generador de BP JSONs, por lo que, la palabra Tools permite abarcar de mejor manera las funcionalidades del sistema.

3. Descripción del problema

3.1. Problema

Como se mencionó anteriormente el sistema Antelope Tools se encarga de mostrar la información de los productores de bloques y sus nodos, para fortalecer la transparencia y fiabilidad de las redes.

Sin embargo, el problema radica en que el sistema no ofrece funcionalidades para conocer el estado de salud de los nodos de la blockchain, lo que es necesario para que los usuarios puedan tener acceso a información de calidad para aumentar la transparencia de la red y mejorar las elecciones.

Tiene un impacto directo en la comunidad EOSIO porque con más información pueden votar por los productores de bloques comprometidos por hacer que la blockchain continúe funcionando como fue concebida. Además, impacta la imagen de la empresa Edenia, ya que, es un proyecto open-source que les brinda visibilidad y demuestra el apoyo y compromiso de la empresa hacia la comunidad EOSIO.

3.2. Solución

Para la solución del problema, se agregó más información de los productores de bloques y sus nodos extraídos de sus archivos BP JSON, debido a que, el estándar fue actualizado, lo que incluyó nuevos campos en este, también, se mejoró considerablemente la validación de las entradas del generador de BP JSON desarrollado por Edenia, para que, de este modo aquellos productores de bloques que utilicen la herramienta brinden la información de acuerdo con el estándar en cuestión.

También, permitir a los usuarios visualizar el listado y el estado de salud de los nodos y sus endpoints de tipo API pertenecientes a la red blockchain, así como filtrar los endpoints según dicho estado, para mostrar aquellos que están funcionando.

Al agregar el estado de salud se aumenta la transparencia en la red debido a que al manifestarse una democracia es de suma importancia contar con información relevante que permita a la comunidad realizar votaciones informadas considerando datos objetivos que les permita efectuar un mejor juicio.

Por ejemplo, un productor en el Top 21 con endpoints que la mayor parte del tiempo falla perjudica la red al no aprovecharse el tiempo para producir bloques y también a los votantes que emplearon sus tokens para elegir a dicho productor.

También, como parte de la solución se incluye la reparación de bugs, errores visuales en el frontend y la realización de pruebas y mantenimiento en el código. Para asegurar un nivel de calidad dentro del rango de aceptación, considerando algunos atributos de calidad definidos en el ISO 25010.

3.3. Stakeholders

En la tabla 1 se despliega la información de los stakeholders del proyecto.

Rol	Nombre	Responsabilidades	Criterios de éxito
Project Manager Intern	Michael Castillo	<p>Organización y estructura de los distintos proyectos correspondientes a los practicantes de la empresa.</p> <p>Responsabilidad dentro del manejo y promoción del SCRUM, como Máster.</p> <p>Apoyo metodológico para la mejora de las etapas del proyecto.</p>	<p>Organizar y administrar las diferentes ceremonias de scrum como el daily standup, planning, demo days y retrospectiva para acompañar y facilitar el avance del proyecto.</p>
Líder Técnico	Xavier Fernández	<p>Coordinación del personal técnico</p> <p>Seguimiento y responsabilidad de la hoja de ruta de los distintos proyectos en marcha.</p> <p>Gestión de clientes externos</p> <p>Desarrollo de soluciones de software</p>	<p>Definir las tareas de la pila del producto con el objetivo de mejorar el sistema para la comunidad.</p> <p>Priorizar las tareas que se realizan en cada sprint para evitar tareas distractoras y entregar valor a los usuarios.</p>
Business Analyst Interino	Leonardo Calderón	<p>Co gestión de programa de practicantes.</p> <p>Seguimiento de proyectos en contacto con las partes.</p> <p>Control de horas de trabajo de colaboradores y practicantes.</p>	<p>Dar seguimiento al trabajo de los practicantes.</p>

Senior UI/UX	Jorge Murillo	Diseño de productos de Web3, en su aspecto y su calidad.	Diseñar las interfaces gráficas del sistema para asegurar una buena experiencia de usuario.
Diseñador UI/UX	Edwin Rodríguez	Diseño gráfico del aspecto visual del sitio, así como la facilitación de recursos gráficos a los desarrolladores	
Senior DevOps	Andrés Gómez	Mantenimiento de servidores e infraestructura de red. Diseño y mejora de soluciones de ciberseguridad. Co gestión de proyectos internos y externos.	Brindar mantenimiento a los servidores y a la infraestructura de red.
Senior Front-end	Terencio Matarrita	Desarrollo de soluciones de software de frontend. Co Gestión de proyectos externos e internos. Revisión del código desarrollado.	Guiar a los practicantes respecto al estándar del código de la empresa. Revisar el código desarrollado por los practicantes para asegurar el cumplimiento de calidad.
Back-end Developer	Leíster Alvarez	Desarrollo de soluciones de software a nivel de lógica de negocios, backend. Co gestión de proyectos internos y externos. Revisión del código desarrollado.	
Full Stack Developer	Angelo Castro	Desarrollo de soluciones de software de frontend y backend. Co gestión de proyectos internos y externos. Revisión del código desarrollado.	
Practicante Full Stack Developer	Fabian Vives	Diseño, desarrollo y pruebas de soluciones de software full stack.	Diseñar y desarrollar software de calidad para que el sistema sea de utilidad a los usuarios.
Practicante Full Stack Developer	Helbertson Torres	Diseño, desarrollo y pruebas de soluciones de software full stack.	Efectuar pruebas al sistema para asegurar que el mismo tenga una calidad aceptable.

Tabla 1. Stakeholders del Proyecto

3.4. Necesidades y expectativas

El enfoque del sistema Antelope Tools es ser un monitor de redes blockchain, que se basan en el protocolo EOSIO, con funcionalidades de un explorador de bloques para promover la transparencia y fiabilidad de las redes al facilitar la visualización de información relevante de los productores de bloques.

Para lo anterior, se necesitaba aumentar la cantidad de información que se mostraba de los productores de bloques y sus nodos. Además, se espera que los datos sean rápidamente accesibles y de manera sencilla para el usuario. El mapa de expectativas se muestra en la Figura 2. Mapa mental de expectativas

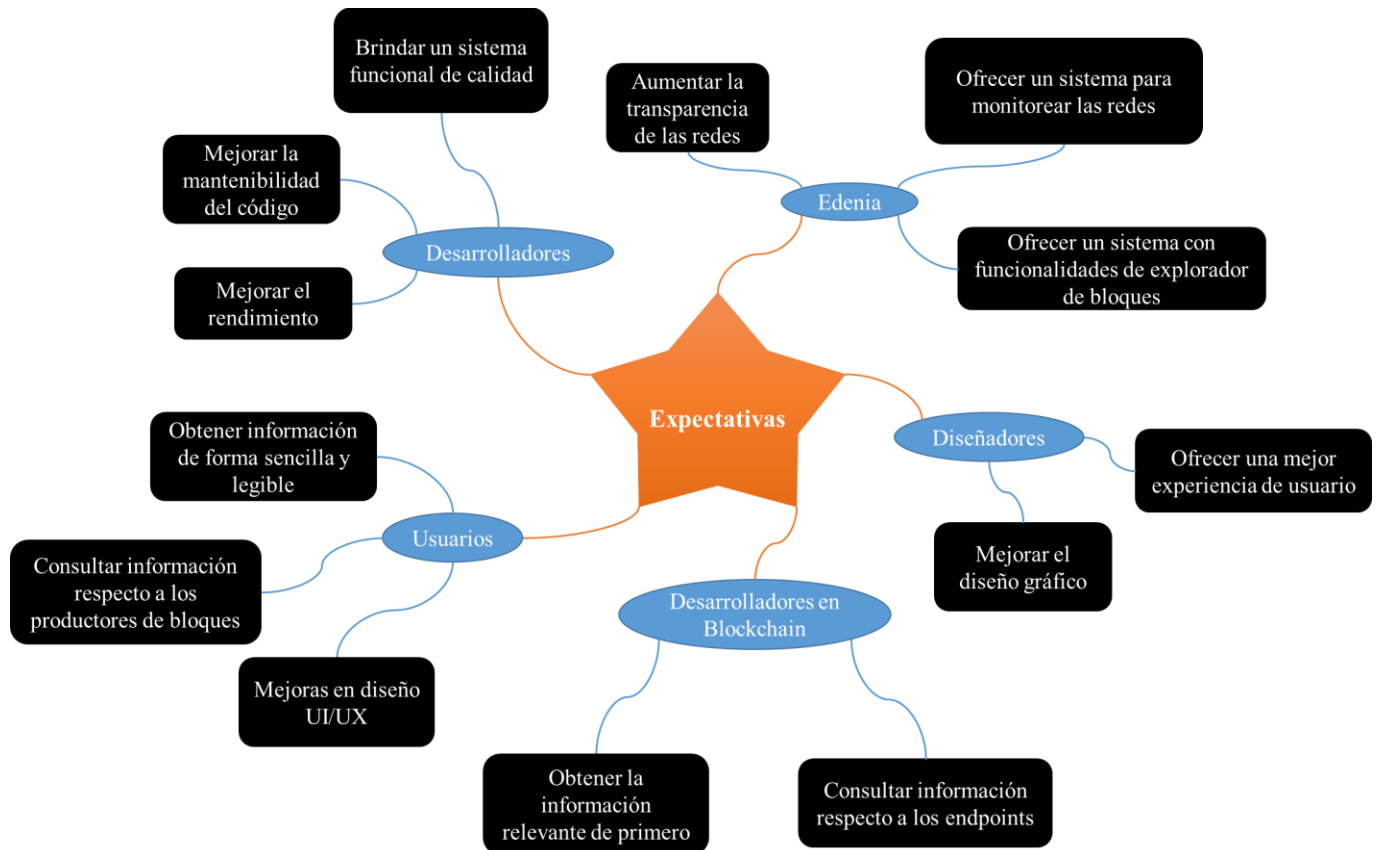


Figura 2. Mapa mental de expectativas

3.5. Requerimientos no funcionales

Entre los requerimientos no funcionales que deberán ser considerados durante el desarrollo del proyecto se destacan los siguientes: Completitud funcional, Correctitud funcional, Comportamiento temporal, Reusabilidad y Usabilidad. Los cuáles serán considerados según su definición en el ISO 25010.

A continuación, en la tabla 2 se muestra el detalle de los requerimientos no funcionales del proyecto.

Requerimiento no funcional	Sub Característica	Definición	Descripción
Adecuación Funcional:	Completitud funcional	Grado en el cual el conjunto de funcionalidades cubre todas las tareas y los objetivos del usuario especificados.	El sistema cumple con las características generales o requerimientos funcionales descritos.
	Correctitud funcional	Capacidad del producto o sistema para proveer resultados correctos con el nivel de precisión requerido.	<p>El sistema debe mostrar resultados correctos.</p> <p>El sistema muestra los datos del productor de bloques y sus nodos según su BP JSON y no el de algún otro productor.</p> <p>El sistema muestra el sensor de salud del nodo según el estado de este según el estado de la red blockchain.</p>
Eficiencia de desempeño	Comportamiento temporal	Los tiempos de respuesta y procesamiento y los ratios de throughput de un	El sistema presenta tiempos de respuestas aceptables en dispositivos móviles y

		sistema cuando lleva a cabo sus funciones bajo condiciones determinadas en relación con un banco de pruebas (benchmark) establecido.	desktop, según la puntuación que sea brindada por la herramienta de Google Lighthouse.
Mantenibilidad	Reusabilidad	Capacidad de un activo que permite que sea utilizado en más de un sistema software o en la construcción de otros activos.	El software es construido de manera que la lógica del componente puede ser reusado por otro componente. El estado usado en un componente puede ser reutilizado por otro sin la necesidad de hacer cambios en la lógica implementada.
Usabilidad	Capacidad para ser usado	Capacidad del producto que permite al usuario operarlo y controlarlo con facilidad.	El sistema permite que el usuario pueda obtener fácilmente y de forma legible la información de los productores de bloques y sus nodos.
	Protección contra errores de usuario	Capacidad del sistema para proteger a los usuarios de cometer errores.	El sistema evita que el usuario cometa errores al manipularlo.

Tabla 2. Requerimientos no funcionales

3.6. Perspectiva, supuestos y dependencias del producto

El proyecto obtiene información de los productores de bloques mediante endpoints, por lo tanto, es necesario que se encuentren disponibles para recuperar ciertos datos que se consultan de manera directa a la blockchain.

Del mismo modo, la información que puede mostrarse de los productores de bloques es dependiente de que los mismos productores de bloques sigan el estándar y brinden la mayor cantidad de información posible en sus BP JSON.

3.7. Características generales

- El sistema debe mostrar la información de los productores de bloques según el estándar definido del BP JSON.
- El sistema debe mostrar el estado de salud de los nodos.
- El sistema debe listar los diferentes APIs y servicios ofrecidos por cada nodo.
- El sistema debe validar entradas en el frontend para evitar introducir datos con formato incorrecto y promover brindar información relevante.
- El sistema debe mostrar el reporte sobre los nodos que están saludables y los que no.
- El sistema debe mostrar las calificaciones de los productores de bloques de EOS Rate.

4. Análisis de riesgos

A continuación, se presenta en el análisis de riesgos realizado para el proyecto:

- a) El sistema ya existe y es complejo

Descripción:	El sistema es un sistema grande, donde varias personas han colaborado durante su desarrollo, además, de contar con distintas partes fundamentales como: Hasura, Hapi y el webapp con React.
Categoría del riesgo:	Técnico
Posible causa del riesgo:	El sistema está en desarrollo aproximadamente desde el año 2020. Asimismo, para cada red, Antelope Tools tiene una configuración.
Impacto:	Alto
Probabilidad de ocurrencia:	Alto
Exposición ante el riesgo:	Alto
Estrategia de mitigación:	1. Familiarizarse con el sistema utilizándolo.

2. Revisar la estructura del código.
3. Solicitar ayuda a los desarrolladores involucrados cuando se tenga alguna duda.

b) Conocimiento básico de blockchain

Descripción:	Se posee poco conocimiento acerca de la tecnología blockchain y sobre el protocolo EOSIO lo que es necesario para entender la lógica del negocio y brindar una solución adecuada.
Categoría del riesgo:	Técnico
Posible causa del riesgo:	Desconocimiento del practicante.
Impacto:	Crítico
Probabilidad de ocurrencia:	Alto
Exposición ante el riesgo:	Alto
Estrategia de mitigación:	<ol style="list-style-type: none">1. Investigar sobre conceptos básicos acerca de blockchain.2. Visitar las guías para desarrolladores de EOS Costa Rica.3. Realizar los cursos de EOSIO Training.4. Solicitar ayuda a los stakeholders con mayor experiencia en la tecnología cuando aparezca una duda.

c) Desconocimiento en tecnologías

Descripción:	No se posee conocimiento de algunas tecnologías utilizadas en el proyecto, tales como PostgreSQL, GraphQL, Hasura o Hapi. Sin embargo, se ha trabajado con tecnologías similares.
Categoría del riesgo:	Técnico
Posible causa del riesgo:	Desconocimiento del practicante.

Impacto:	Alto
Probabilidad de ocurrencia:	Bajo
Exposición ante el riesgo:	Medio
Estrategia de mitigación:	<ol style="list-style-type: none">1. Revisar la documentación técnica de cada tecnología.2. Revisar el código existente que hace uso de dichas tecnologías.

d) Tiempo insuficiente

Descripción:	Que el tiempo de 16 semanas no sea suficiente para cumplir las necesidades y expectativas del proyecto.
Categoría del riesgo:	Gestión
Posible causa del riesgo:	Mala gestión del tiempo o mala estimación.
Impacto:	Medio
Probabilidad de ocurrencia:	Medio
Exposición ante el riesgo:	Medio
Estrategia de evasión:	<ol style="list-style-type: none">1. Realizar diariamente el stand up para informar lo más pronto posible cuando un bloqueo ocurre.2. Priorizar correctamente las tareas para evitar enfocarse en tareas menos relevantes o distractoras.

e) Calidad inferior a la esperada

Descripción:	Que la calidad del sistema con respecto a los requerimientos no funcionales no se encuentre en un rango aceptable según lo definido.
Categoría del riesgo:	Técnico

Posible causa del riesgo: Poca atención en el aseguramiento de la calidad.

Impacto: Medio

Probabilidad de ocurrencia: Medio

Exposición ante el riesgo: Medio

Estrategia de evasión:

1. Realizar pruebas al software considerando los requerimientos no funcionales y crear tareas para solucionar los errores que sean encontrados.
2. Revisar y efectuar pruebas en el nuevo código desarrollado o cambiado antes de confirmar los pull request.

f) Perder avance

Descripción: Que mientras se trabaja en una tarea se pierda el avance realizado por alguna causa externa.

Categoría del riesgo: Externo

Posible causa del riesgo: Daño de equipo, pérdida de servicio eléctrico, sobrecargas, falla inesperada en software o hardware.

Impacto: Medio

Probabilidad de ocurrencia: Bajo

Exposición ante el riesgo: Medio

Estrategia de evasión:

1. Hacer uso del gestor de versiones.
2. Realizar commits locales con los cambios frecuentemente.
3. Publicar los cambios remotamente con frecuencia.
4. Utilizar un regulador de voltaje adecuado.

g) Entender mal un requerimiento funcional

Descripción:	Si no se comprende correctamente el requerimiento que se desea implementar se puede desarrollar una solución con un comportamiento distinto al deseado, lo que implicaría usar tiempo para realizar los cambios pertinentes.
Categoría del riesgo:	Gestión
Posible causa del riesgo:	No comunicar las dudas que se tiene respecto a algún requerimiento funcional, no preguntar si esta complementa comprendida la tarea por realizar o esperar hasta terminar la implementación para mostrarla.
Impacto:	Alto
Probabilidad de ocurrencia:	Medio
Exposición ante el riesgo:	Medio
Estrategia de evasión:	<ol style="list-style-type: none">1. Preguntar lo antes posible si existen dudas respecto a un requerimiento.2. Repetir a la persona indica lo que se entendió para que le indique si es correcto.3. Mostrar en que se ha estado trabajando para recibir retroalimentación pronto.

5. Objetivos y Alcance del sistema

5.1. Objetivo general

Desarrollar y diseñar software que aumenten el sistema Antelope Tools para reportar el estado de salud de distintos nodos en redes blockchain basadas en EOSIO mediante la consulta de información sobre cuentas y contratos inteligentes de redes públicas para aumentar la transparencia en la red.

5.2. Objetivos específicos

- I. Desarrollar una solución que liste, filtre y realice un reporte estadístico de los nodos mediante determinar su estado de salud para facilitar el acceso a la información.

- II. Desplegar más información de los nodos mediante consultas a los archivos BP JSON y al sistema EOS Rate en la búsqueda de incrementar la transparencia en la red.
- III. Implementar la validación del ingreso de datos en los diferentes formularios del frontend para evitar información incompleta o incorrecta.
- IV. Efectuar mejoras sobre el código existente para mostrar la información sin necesidad de tener que refrescar la página y emplear mejoras en SEO para mejorar el posicionamiento del sitio web en las listas de buscadores.
- V. Realizar Pruebas y Mantenimiento de código para reparar los errores que se encuentren y validar el funcionamiento correcto del sistema.

5.3. Alcance

En el alcance del proyecto se incluyó, por un lado, la implementación de un sensor de salud para los nodos, así como el despliegue de más información de estos y tareas para realizar mejoras en distintos aspectos del sistema como validar las entradas, fortalecer el SEO del sitio y reparación de errores.

Sin embargo, no se consideró la selección de tecnologías o el diseño de la arquitectura del sistema porque este ya existe. Además, de destacar que el enfoque es que el Antelope Tools sea un monitor de red con funcionalidades de exploradores de bloques, por lo que, sus funcionalidades se encuentran dentro de este ámbito.

6. Modelo de Diseño

6.1. Arquitectura del sistema

A continuación, en esta sección se explicará la arquitectura del sistema, por lo que, primero es necesario comprender acerca el patrón arquitectónico en el que se basó para ser construido, el cual es el patrón demux presentado para infraestructura de backend para dapps basadas en el protocolo EOSIO, como detalla EOS Costa Rica (2022).

6.1.1. Patrón demux

Este patrón se inspira en el patrón de Arquitectura Flux y Redux. Donde, por un lado, Redux hace referencia a un contenedor de estado predecible para aplicaciones de JavaScript de manera global. Y por el otro, el patrón Flux busca ser la solución al problema de escalabilidad que tiene el patrón de arquitectura Model-View-Controller, por lo que, agrega el concepto de acciones que mediante un dispatcher son procesadas produciendo interacciones con el mecanismo de persistencia, la diferencia radica en que la vista también puede provocar acciones, este comportamiento produce que el flujo de los datos en la aplicación sea en una sola dirección.

Así que demux, como se indica en EOS Costa Rica (2022), toma elementos de estas dos propuestas para intentar resolver el problema de construir aplicaciones que requieren consultar el estado de una blockchain, debido a que, realizar muchas lecturas de forma directa son costosas tanto en tiempo como poder de cómputo, por lo que, el patrón mantiene una capa de persistencia separada y utiliza la blockchain como única fuente de verdad, esto porque funciona como un gran registro lo que permite que si los datos en la capa de persistencia son eliminados pueden ser recuperados mediante la consulta a la blockchain.

Para continuar, ver Figura 3. Patrón arquitectónico demux. (EOS Costa Rica, 2022), demux propone leer de la blockchain acciones y actualizar la capa de persistencia con estas, de este modo se puede exponer una API que permita consultar el estado de una manera más eficiente de lo que sería consultar directamente la blockchain. Asimismo, como en el patrón Flux existe la posibilidad que la capa de vista provoque cambios en el estado de la blockchain mediante transacciones y a su vez dicho cambio provocará acciones que actualizarán la capa de persistencia.

Igualmente, el patrón admite que cuando una acción es manejada se genere un efecto secundario que puede realizar transacciones en la blockchain o notificar a servicios externos de la acción realizada.

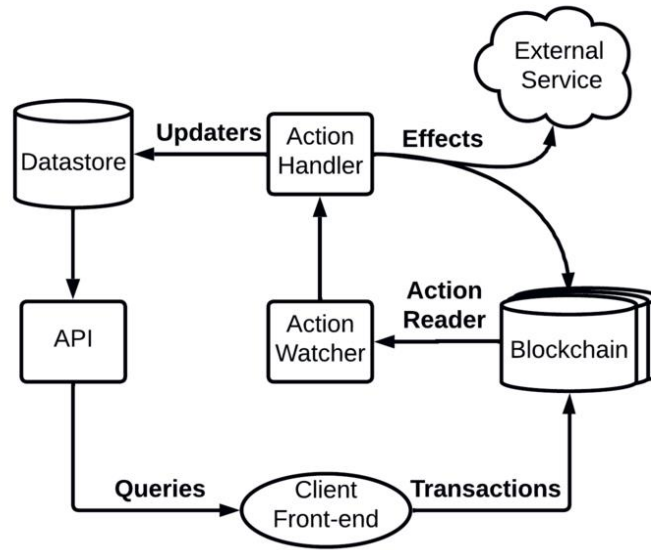


Figura 3. Patrón arquitectónico demux. (EOS Costa Rica, 2022)

6.1.2. Implementación del patrón

Aunque existe una implementación llamada demux js, el sistema Antelope Tools no la integra, sino que se implementa su propia arquitectura considerando las propuestas del patrón. La arquitectura, presente en la Figura 4. Arquitectura del sistema. (EOS Costa Rica, 2020), puede dividirse en el frontend en la webApplication construida con la biblioteca React y el backend desarrollado con node js utilizando Hapi, en la capa de persistencia se tiene una base de datos PostgreSQL la cual se consulta mediante Hasura GraphQL y un servicio con una Wallet para realizar transacciones en la red blockchain.

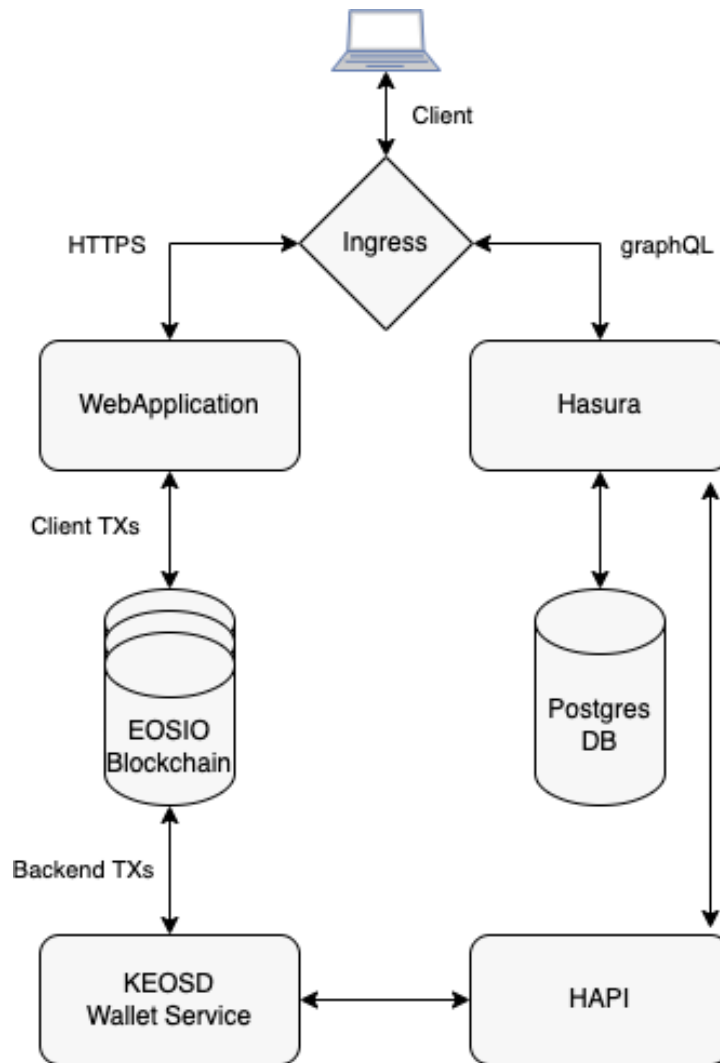


Figura 4. Arquitectura del sistema. (EOS Costa Rica, 2020)

6.2. Modelos de subsistemas

Como se indica en la sección anterior, el sistema puede dividirse en Hapi, Hasura, Wallet y el webapp React, donde cada uno es mantenido en un contenedor de Docker, además del volumen con la base de datos en Postgres, tal como se observa en la Figura 5. Diagrama de paquetes. Por un lado, con Hasura se interactúa con la base de datos PostgreSQL a través del lenguaje de consulta GraphQL, lo que permite fácilmente gestionar los datos sin preocuparse por la creación del esquema y las operaciones CRUD.

Por otro lado, a través de Hapi se expone una API del lado del backend, la cual puede ser consultada mediante solicitudes HTTP que la aplicación en React emplea para desplegar los datos. Por último, se tiene el servicio del Wallet, que se encarga de realizar transacciones en la blockchain.

Continuando, si se enfoca en la funcionalidad para determinar el estado de salud de un nodo, se puede destacar que en Hapi se definen los servicios que se ejecutan cada cierto tiempo para consultar la información de los nodos y actualizar la base de datos. Del mismo modo, en el webapp se puede destacar las carpetas de hooks y gql que son utilizadas para consultar el backend, y por último la carpeta de routes y components que contienen los componentes React para desplegar los datos en el frontend.

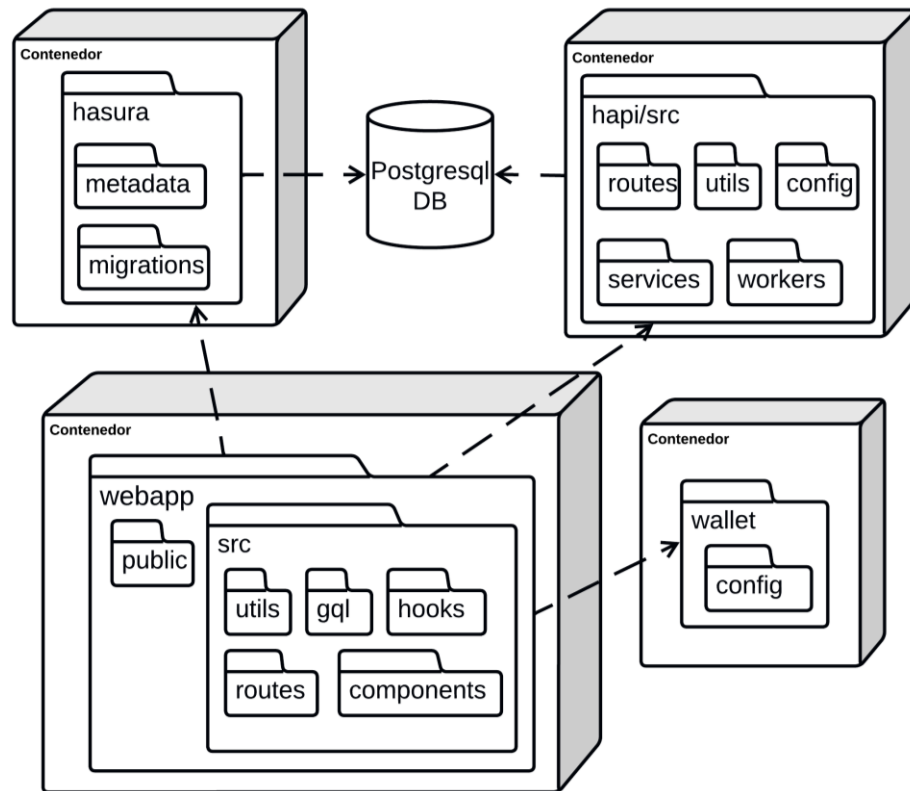


Figura 5. Diagrama de paquetes.

6.3. Interfaces de usuario

Respecto a los cambios realizados por el estudiante durante la práctica se destaca que en la página principal del sistema como puede verse en Figura 6. Página principal antes del cambio., se agregó la funcionalidad de esconder o desplegar el menú del sidebar, para aprovechar el espacio cuando el sidebar se encuentre escondido. Los cambios son visibles en las figuras Figura 7. Página principal con el menú lateral escondido. y Figura 8. Página principal con el nuevo menú lateral.

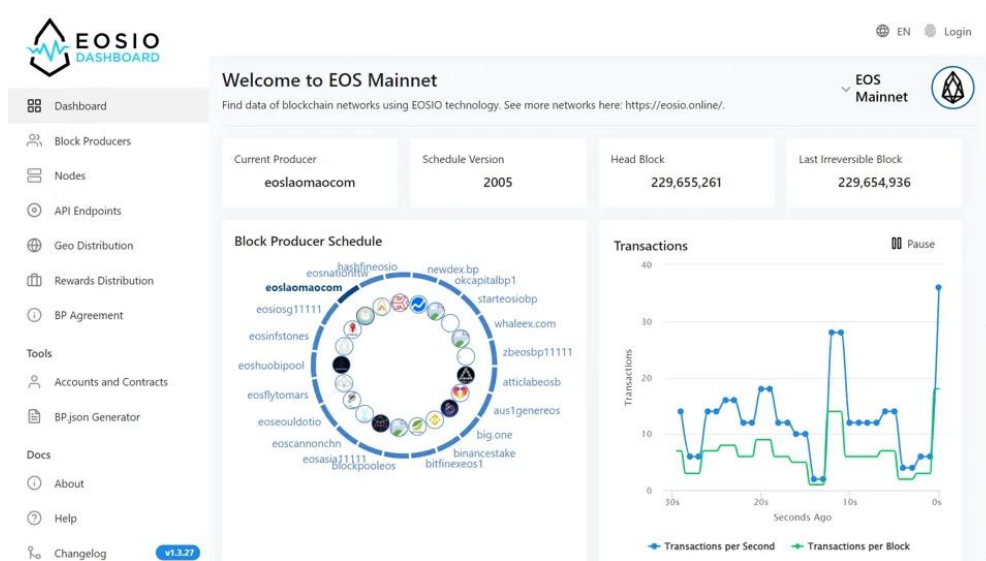


Figura 6. Página principal antes del cambio.

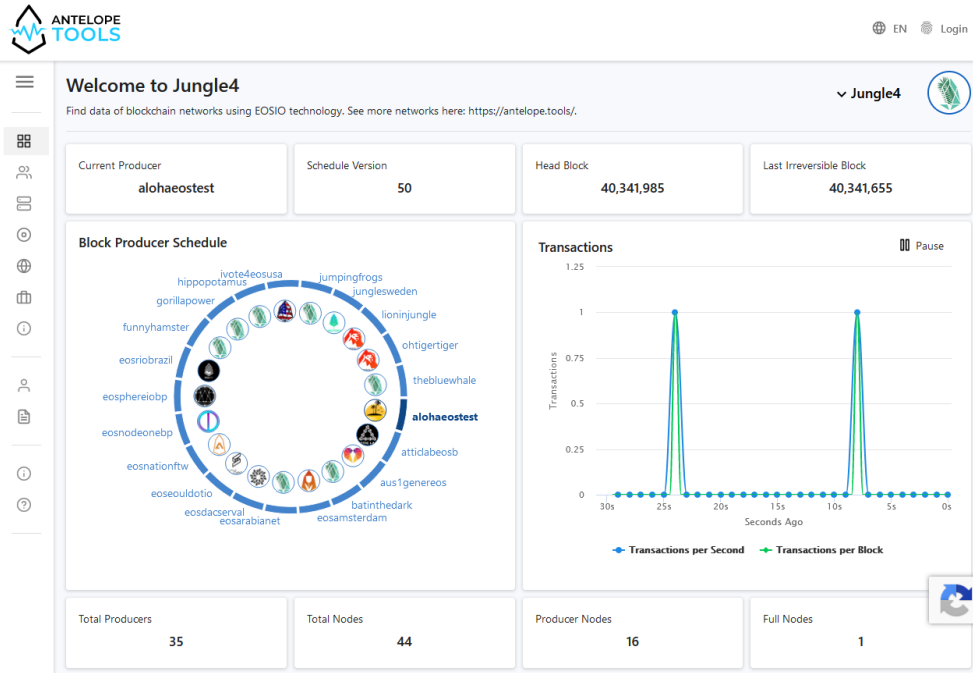


Figura 7. Página principal con el menú lateral escondido.

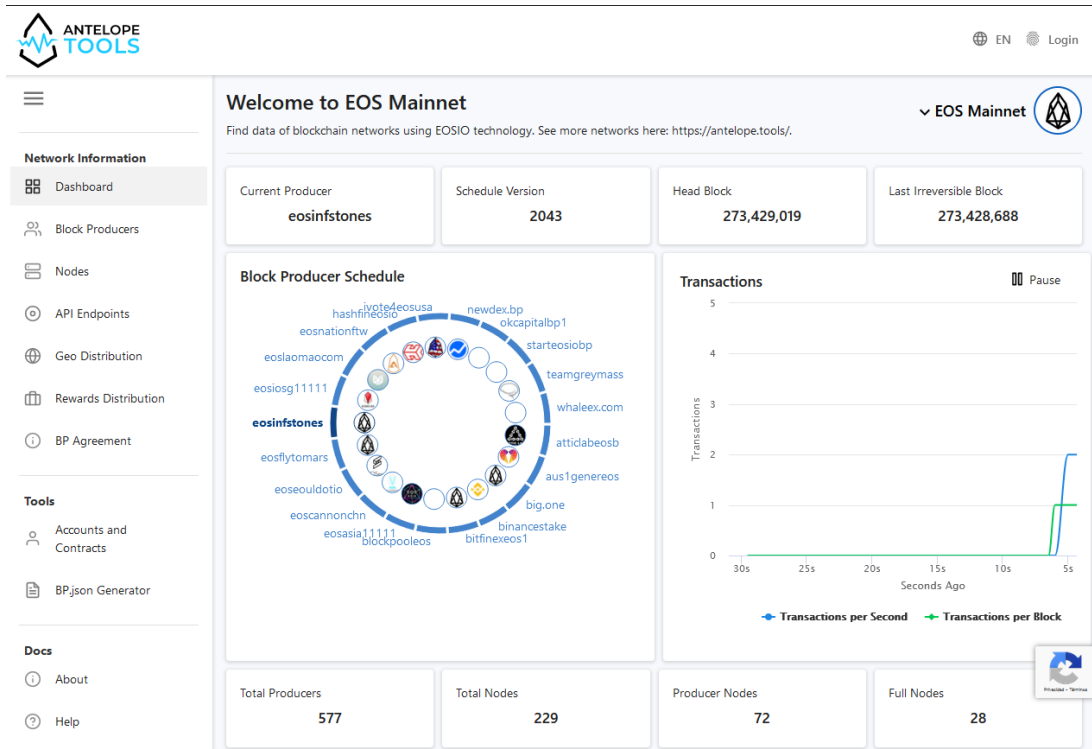
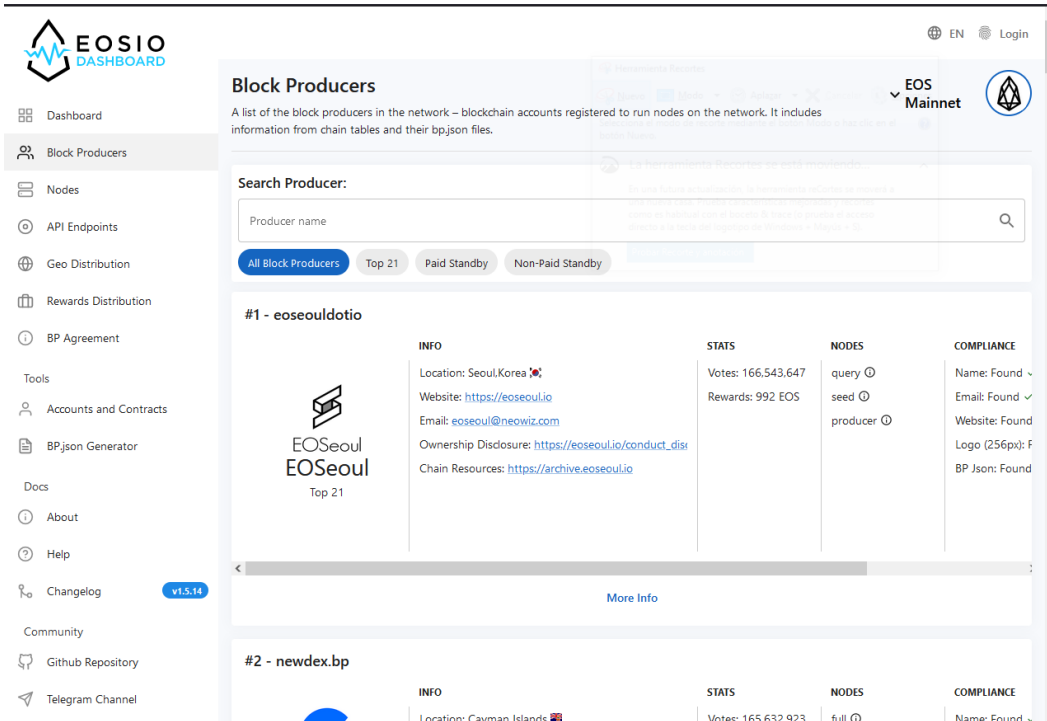


Figura 8. Página principal con el nuevo menú lateral.

En la página de productores, se corrigieron errores como el botón de “More Info” que no tenía utilidad cuando el sitio se muestra en dispositivos con pantallas grandes, se agregó una validación de que el logo del productor corresponde a una imagen de al menos 100 píxeles de ancho y 100 píxeles de largo, ya que, en el caso contrario debe mostrar un logo por defecto. Por último, debido a que el estándar de BP JSON había cambiado, se agregaron campos que no se estaban desplegando.

Los cambios pueden ser observados al comparar las imágenes de la Figura 9. Página de productores de bloques antes del cambio. y Figura 10. Página de productores de bloques después del cambio. Asimismo, de mostrar retroalimentación cuando no se encuentran resultados, debido a que antes no lo mostraba, el cambio se aprecia en la Figura 11. Búsqueda sin resultados.




Block Producers
A list of the block producers in the network – blockchain accounts registered to run nodes on the network. It includes information from chain tables and their bpjson files.

Search Producer:

Producer name

All Block Producers | Top 21 | Paid Standby | Non-Paid Standby

#1 - eoseouldotio	INFO	STATS	NODES	COMPLIANCE
 EOSeoul Top 21	Location: Seoul, Korea 🇰🇷 Website: https://eoseoul.io Email: eoseoul@newowiz.com Ownership Disclosure: https://eoseoul.io/conduct_disi Chain Resources: https://archive.eoseoul.io	Votes: 166,543,647 Rewards: 992 EOS	query Ⓞ seed Ⓞ producer Ⓞ	Name: Found Email: Found Website: Found Logo (256px): F BP Json: Found

More Info

#2 - newdex.bp	INFO	STATS	NODES	COMPLIANCE
	Location: Cayman Islands 🇰🇲	Votes: 165,632,923	full Ⓞ	Name: Found

Figura 9. Página de productores de bloques antes del cambio.

#1 - edeniaedenia


INFO	STATS	NODES	COMPLIANCE	SOCIAL	
	Location: San Jose 🇸🇨 Website: https://edenia.com Email: bp@edenia.com Ownership Disclosure: https://edenia.o Chain Resources: https://archive.edenia Other Resources: https://libre.eosio.online https://libre-explorer.edenia.cloud https://libre-hyperion.edenia.cloud	Votes: 1 Rewards: 508 LIBRE	query ⓘ seed ⓘ producer ⓘ	BP Json ✓ Name ✓ Email ✓ Website ✓ Logo (256px) ✓ Country ✓	hive github reddit twitter youtube facebook telegram

Figura 10. Página de productores de bloques después del cambio.

Search Producer:

Producer Name

🔍

All
Top 21
Paid Standby
Non-Paid Standby

No results found

Figura 11. Búsqueda sin resultados.

Continuando, en el generador de BP JSON se agregó la validación de entradas en el formulario, como se observa en la Figura 13. Página del generador después del cambio., además, de indicar visualmente cuáles campos poseen datos con formato incorrecto o son obligatorios para generar el archivo. Puede compararse con la Figura 12. Página del generador antes del cambio.

EOSIO DASHBOARD

EN Login

EOS Mainnet

BPjson Generator

A visual representation of the network's current nodes.

BP JSON Generator

A simple way to create and update your node information on chain.

Drop your BP json file here

Account Name:

Organization Info

Candidate Name * Website

Code of Conduct Ownership disclosure Email *

Github User + Chain Resources Other Resources +

Branding

Logo 256px Logo 1024px Logo SVG

Location

Name Country

Latitude Longitude

0 0

Social

Keybase Telegram Twitter

Figura 12. Página del generador antes del cambio.

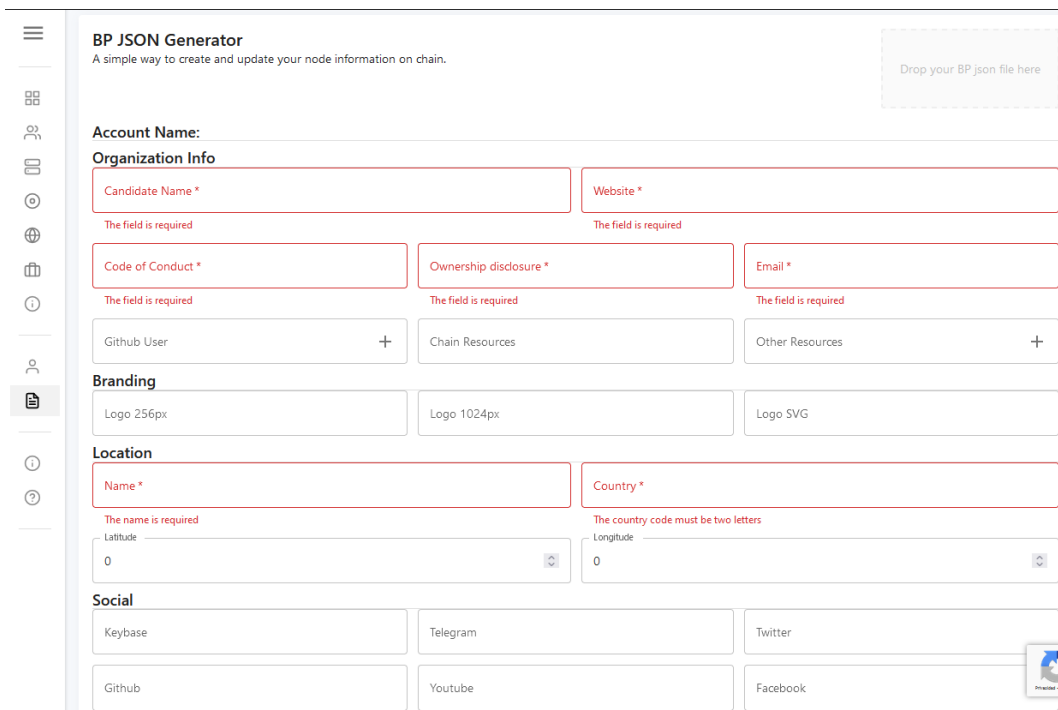


Figura 13. Página del generador después del cambio.


En la lista de endpoints se agrega un ícono al lado de la dirección URL que indica el estado de salud de dicho endpoint, como se observa en la Figura 15. Página de endpoints con sensor de salud. donde se emplea una analogía similar a las luces de un semáforo, asimismo, de incluir un mensaje de ayuda para explicar el significado de cada ícono, ver Figura 17. Ventana de ayuda con el significado del sensor de salud.

Del mismo modo, al presionar con un clic el ícono de salud se despliega una pequeña ventana emergente con la información del endpoint, como se muestra en la Figura 18. Diálogo con la información del endpoint. , indicando el estado de la respuesta HTTP, el tiempo del último bloque y el momento en que fue realizada la consulta.

Además, se agregó un filtro para mostrar solo aquellos endpoints con un estado saludable o desincronizado, tal y como se ve en la Figura 16. Página de endpoints con filtrado de nodos saludables. Todos los cambios pueden compararse con la página antes del cambio de la Figura 14. Página con la lista de endpoints antes del cambio.

API Endpoints

An updated list of public API endpoints provided by node operators in the network.

LACChain EOSIO 

Endpoints available by Entity
Updated at: October 12, 2022

Items per page
80

Entity	HTTP	HTTPS	P2P
Inter-American Development Bank	N/A	N/A	eos.boot1.testnet.lacchain.net:9876 eos.val2.testnet.lacchain.net:9876
Edenia	http://observer.edenia.cloud http://lacchain.edenia.cloud	https://observer.edenia.cloud https://lacchain.edenia.cloud	validator2.edenia.cloud:9874 observer.edenia.cloud:9873 writer.edenia.cloud:9872
EOS Argentina	http://obs1.testnet.lacchain.eosargentina.io http://wrt1.testnet.lacchain.eosargentina.io	https://obs1.testnet.lacchain.eosargentina.io https://wrt1.testnet.lacchain.eosargentina.io	obs1.testnet.lacchain.eosargentina.io:9873 val1.testnet.lacchain.eosargentina.io:9870 wrt1.testnet.lacchain.eosargentina.io:9875

Figura 14. Página con la lista de endpoints antes del cambio.

Endpoints available by Producer

Updated at: November 10, 2022 10:41 PM

Only endpoints responding Items per page
20






















Producer	HTTP 	HTTPS 	P2P 
eoseouldotio	http://api.eoseoul.io 	https://api.eoseoul.io 	p2p2.eoseoul.io:30333
eosflytomars	http://api.bitmars.one 	https://api.bitmars.one 	p2p.bitmars.one:8080
newdex.bp	http://eos.newdex.one 	https://eos.newdex.one 	eos.newdex.one:9876
binancestake	http://api.binance-eos.com 	https://api.binance-eos.com 	api.binance-eos.com:9875
eosnationftw	http://eos.api.eosnation.io 	https://eos.api.eosnation.io  https://eos.dfuse.eosnation.io  https://eos.firehose.eosnation.io 	eos.seed.eosnation.io:9876
eoscannonchn	http://mainnet.eoscannon.io 	https://mainnet.eoscannon.io 	node1.eoscannon.io:59876
hashfineosio	N/A	N/A	eos.hashfin.com:9876
aus1genereos	http://mainnet.genereos.io 	https://mainnet.genereos.io 	p2p.genereos.io:9876
okcapitalbp1	http://eosapi.okpool.top 	https://eosapi.okpool.top 	eos.okpool.top:9876

Figura 15. Página de endpoints con sensor de salud.

Endpoints available by Producer
Updated at: November 12, 2022 12:16 PM

Only endpoints responding Items per page 20

Producer	HTTP	HTTPS	P2P
teamgreymass	http://wax.greymass.com ✓	https://wax.greymass.com ✓	wax.greymass.com:35777
waxswedenorg	http://api.waxsweden.org ✓	https://api.waxsweden.org ✓	p2p.waxsweden.org:35777
eosdacserver	http://wax.eosdac.io ✓	https://wax.eosdac.io ✓	wax-p2p.eosdac.io:29876
nation.wax	http://wax.api.eosnation.io ✓	https://wax.api.eosnation.io ✓ https://wax.dfuse.eosnation.io ✓	wax.seed.eosnation.io:9876
pink.gg	http://wax.pink.gg ✓	https://wax.pink.gg ✓	peer1.wax.pink.gg:36715 peer2.wax.pink.gg:36715
eosriobrazil	http://wax.eosrio.io ✓ http://eu.wax.eosrio.io ✓	https://wax.eosrio.io ✓ https://eu.wax.eosrio.io ✓	br.eosrio.io:35668
amsterdamwax	http://wax.eu.eosamsterdam.net ✓	https://wax.eu.eosamsterdam.net ✓	wax.eu.eosamsterdam.net:9101 waxp2p.us.eosamsterdam.net:9101

Figura 16. Página de endpoints con filtrado de nodos saludables.

Help





-  The endpoint is working
-  The endpoint works but is outdated with the last block
-  Error in the endpoint request
-  The endpoint is not responding

Figura 17. Ventana de ayuda con el significado del sensor de salud.

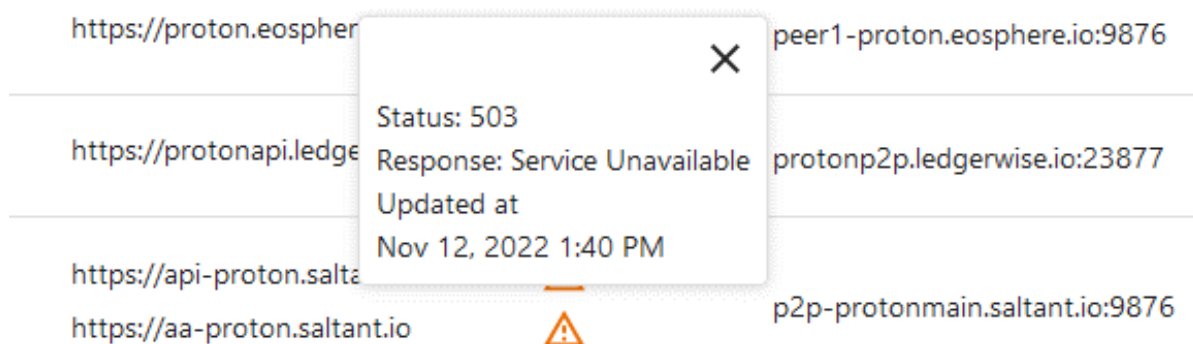
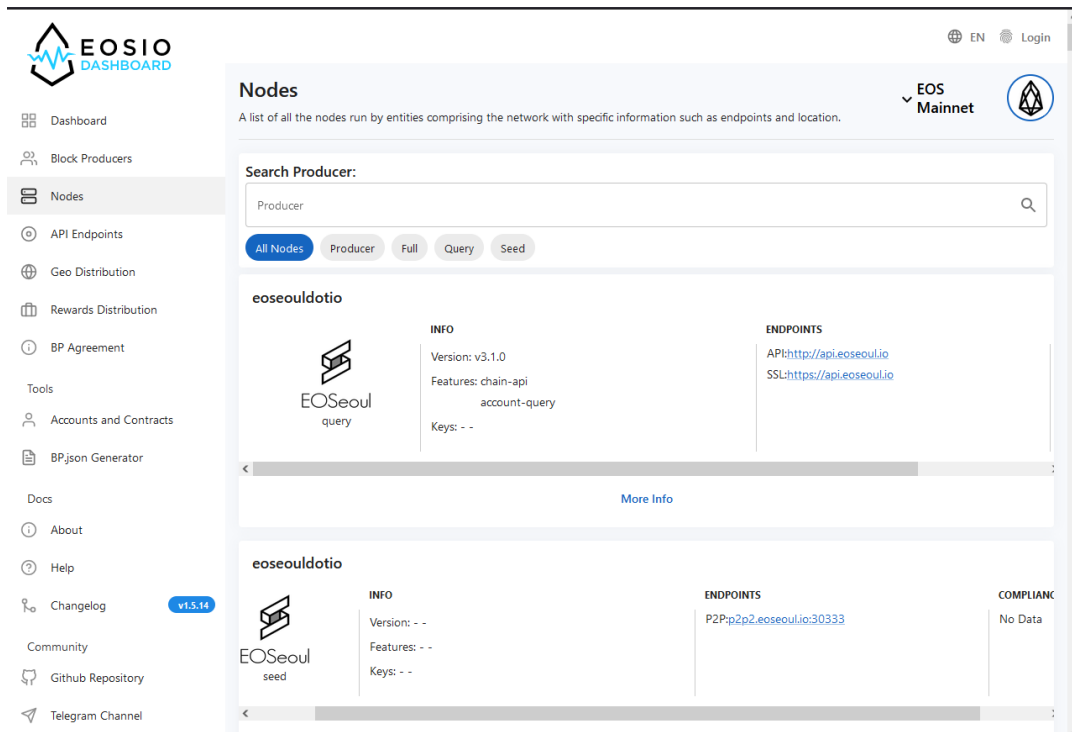


Figura 18. Diálogo con la información del endpoint.

En la página de nodos debido a que solo los nodos de tipo query tienen más información, en los otros nodos se muestran pocos datos, esto sumado al hecho de que un productor de bloques con solo sus nodos ocupa la mayoría de la vista vertical, como se muestra en la Figura 19. Página de nodos antes del cambio. , es por esta razón, que se solicitó el presentar los nodos por productor de manera horizontal.

Asimismo, se implementó el sensor de salud con el ícono como en la lista de endpoints para indicar cuántos de estos responden en el nodo. Como se ve en la Figura 20. Página de nodos después del cambio.



EOSIO DASHBOARD

EN Login

Nodes
A list of all the nodes run by entities comprising the network with specific information such as endpoints and location.

Search Producer: [Producer] [Search]

All Nodes | Producer | Full | Query | Seed

eoseouldotio

EOSeoul query

INFO
Version: v3.1.0
Features: chain-api, account-query
Keys: - -

ENDPOINTS
API: <http://api.eoseoul.io>
SSL: <https://api.eoseoul.io>

[More Info](#)

eoseouldotio

EOSeoul seed

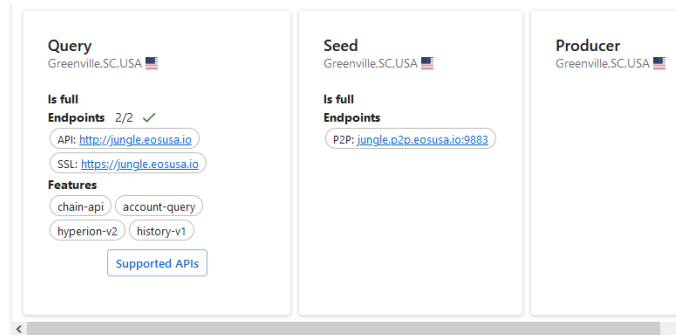
INFO
Version: - -
Features: - -
Keys: - -

ENDPOINTS
P2P: <p2p2.eoseoul.io:30333>

COMPLIANCE
No Data

Figura 19. Página de nodos antes del cambio.

#12 - ivote4eosusa

Query
Greenville, SC, USA

Is full

Endpoints 2/2 ✓
API: <http://jungle.eosusa.io>
SSL: <https://jungle.eosusa.io>

Features
chain-api, account-query, hyperion-v2, history-v1

[Supported APIs](#)

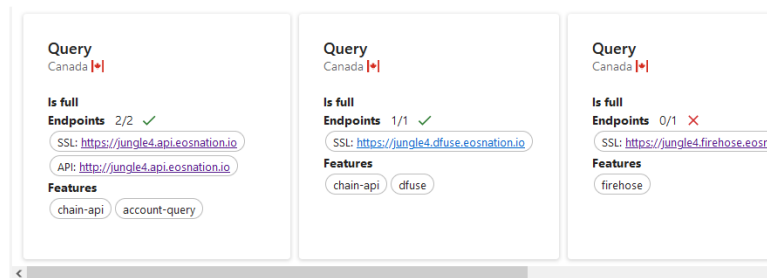
Seed
Greenville, SC, USA

Is full

Endpoints
P2P: [jungle.p2p.eosusa.io:9883](p2p.jungle.p2p.eosusa.io:9883)

Producer
Greenville, SC, USA

#10 - eosnationftw

Query
Canada

Is full

Endpoints 2/2 ✓
SSL: <https://jungle4.api.eosnation.io>
API: <http://jungle4.api.eosnation.io>

Features
chain-api, account-query

Query
Canada

Is full

Endpoints 1/1 ✓
SSL: <https://jungle4.dfuse.eosnation.io>

Features
chain-api, dfuse

Query
Canada

Is full

Endpoints 0/1 ✗
SSL: <https://jungle4.firehose.eosn>

Features
firehose

Figura 20. Página de nodos después del cambio.

6.4. Componentes y servicios

Para la funcionalidad del estado de salud de los nodos, se modificaron los siguientes servicios esenciales en el backend: servicio de producer, eosio, lacchain y se añadió el servicio de endpoints.

En la Figura 24. Diagrama de actividad con el servicio de producers. se presenta el diagrama de actividad con la lógica que se realiza en el servicio del producer y endpoints para obtener la información actualizada de los productores de bloques.

Ahora bien, el servicio de producer se encarga de consultar mediante la biblioteca eosjs la cuenta eosio de la red actual, consultando la tabla producers del contrato inteligente, de esta manera se obtienen todas las direcciones web registradas por los productores de bloques. Es importante destacar que la consulta se realiza mediante el endpoint de un nodo, lo que implica que, si el endpoint en el momento de la consulta no está respondiendo la petición, los datos de los productores de bloques no se obtendrán.

Entonces, para resolver este inconveniente se aprovecha que en la red blockchain existen múltiples nodos con diferentes endpoints, así que, para aumentar la disponibilidad de los datos se implementó el uso de una lista de endpoints, en lugar de depender solo de uno. De esta manera es menos probable que una consulta mediante el API de eosjs no obtenga respuesta. La lógica anterior es descrita con mayor detalle en la Figura 21. Diagrama de actividad. Efectuar una consulta mediante el API de eosjs..

Continuando con el proceso de obtención de los urls de los productores de bloques, si ningún endpoint responde la petición con la información, se consulta directamente los registros de los productores de la base de datos del sistema.

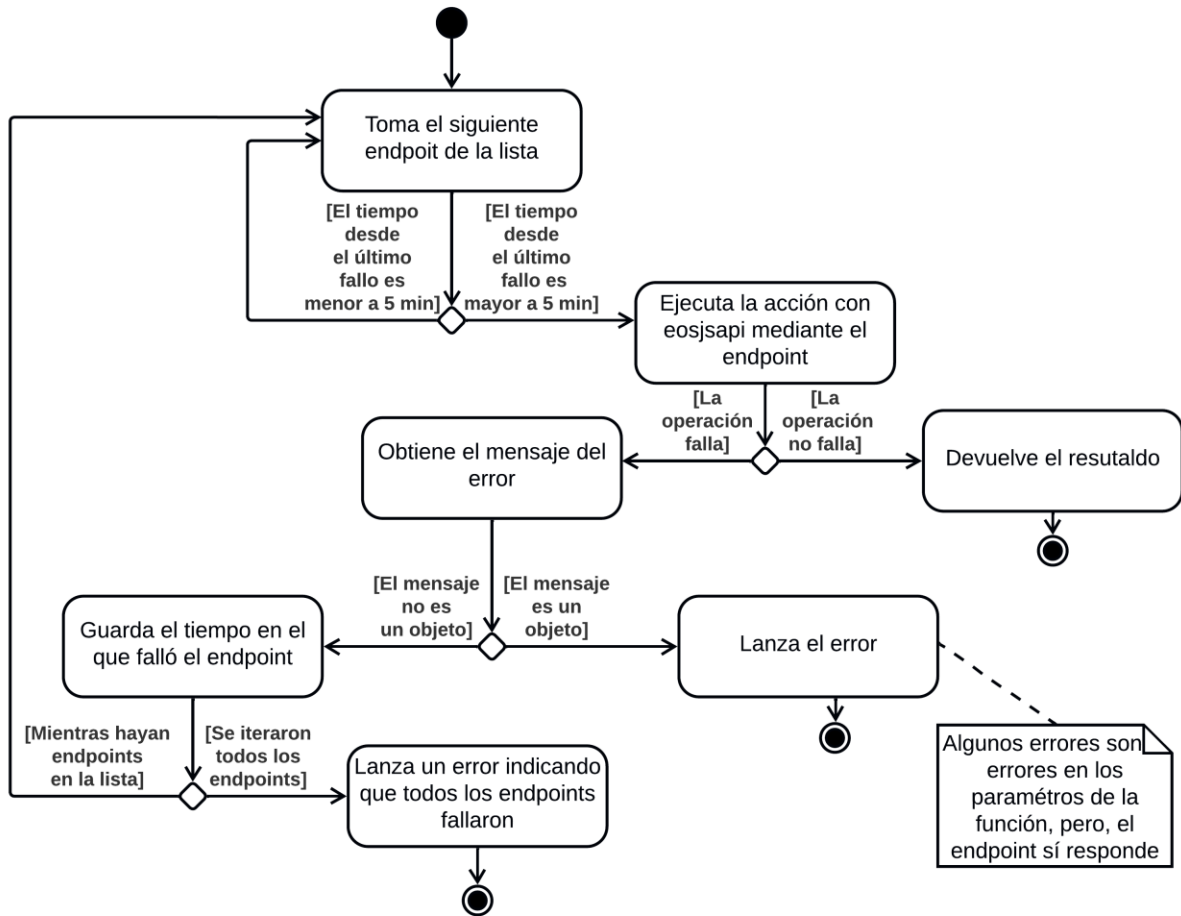


Figura 21. Diagrama de actividad. Efectuar una consulta mediante el API de eosjs.

Luego de esto se solicita, para cada productor de bloques, el recurso bajo la ruta /chains.json, la cuál debería devolver un archivo JSON, como se muestra en la Figura 22. Ejemplo de objeto chains.json., con el chain_id como llave y la ruta del BP JSON como valor, si el productor sigue el estándar el objeto chains.json muestra en qué redes blockchain participa. Con el chain_id de la red actual se obtiene la dirección en la que puede encontrarse el BP JSON.

```
{
  "chains": {
    "aca376f206b8fc25a6ed44dbdc66547c36c6c33e3a119ffbeaf943642f0e906": "/bp.json",
    "73e4385a2708e6d7048834fbc1079f2fabb17b3c125b146af438971e90716c4d": "/jungle4.json",
    "384da888112027f0321850a169f737c33e53b388aad48b5adace4bab97f437e0": "/proton.json",
    "71ee83bcf52142d61019d95f9cc5427ba6a0d7ff8accdd9e2088ae2abeaf3d3dd": "/proton-testnet.json",
    "4667b205c6838ef70ff7988f6e8257e8be0e1284a2f59699054a018f743b1d11": "/telos.json",
    "1eaa0824707c8c16bd25145493bf062aecddfeb56c736f6ba6397f3195f33c9f": "/telos-testnet.json",
    "1064487b3cd1a897ce03ae5b6a865651747e2e152090f99c1d19d44e01aea5a4": "/wax.json",
    "f16b1833c747c43682f4386fca9cbb327929334a762755ebec17f6f23c9b8a12": "/wax-testnet.json"
  }
}
```

Figura 22. Ejemplo de objeto chains.json.

Después de esto, se consulta mediante el protocolo HTTP las direcciones del BP JSON y se almacenan dichos archivos en la base de datos. Debido a que los productores de bloques y sus BP JSON no cambian con mucha frecuencia el sistema ejecuta cada 4 horas este servicio, lo que vuelve a consultar la información de los productores y actualiza la información en la base de datos.

Como será explicado en la sección de diseño de base de datos, se agregaron nuevas tablas por lo que es responsabilidad de este mismo servicio actualizarlas usando la información del BP JSON.

Por otro lado, se incluyó el servicio de sincronización de endpoints, este servicio se ejecuta cada 2 minutos, y consulta cada endpoint de tipo API y SSL para almacenar la respuesta en la base de datos y brindar la información necesaria para determinar el estado de salud. Para esto se calcula la diferencia del tiempo actual del cliente y la fecha registrada del último bloque del endpoint consultado, la lógica se presenta en el diagrama de flujo de la Figura 23. Diagrama de flujo. Estado de salud del endpoint.

Entonces, si dicha diferencia es menor o igual a 3 minutos el endpoint se considera saludable porque está funcionando, si esta condición no se cumple se utiliza la respuesta HTTP para determinar si el endpoint brindó una respuesta exitosa, de error o no devolvió ninguna.

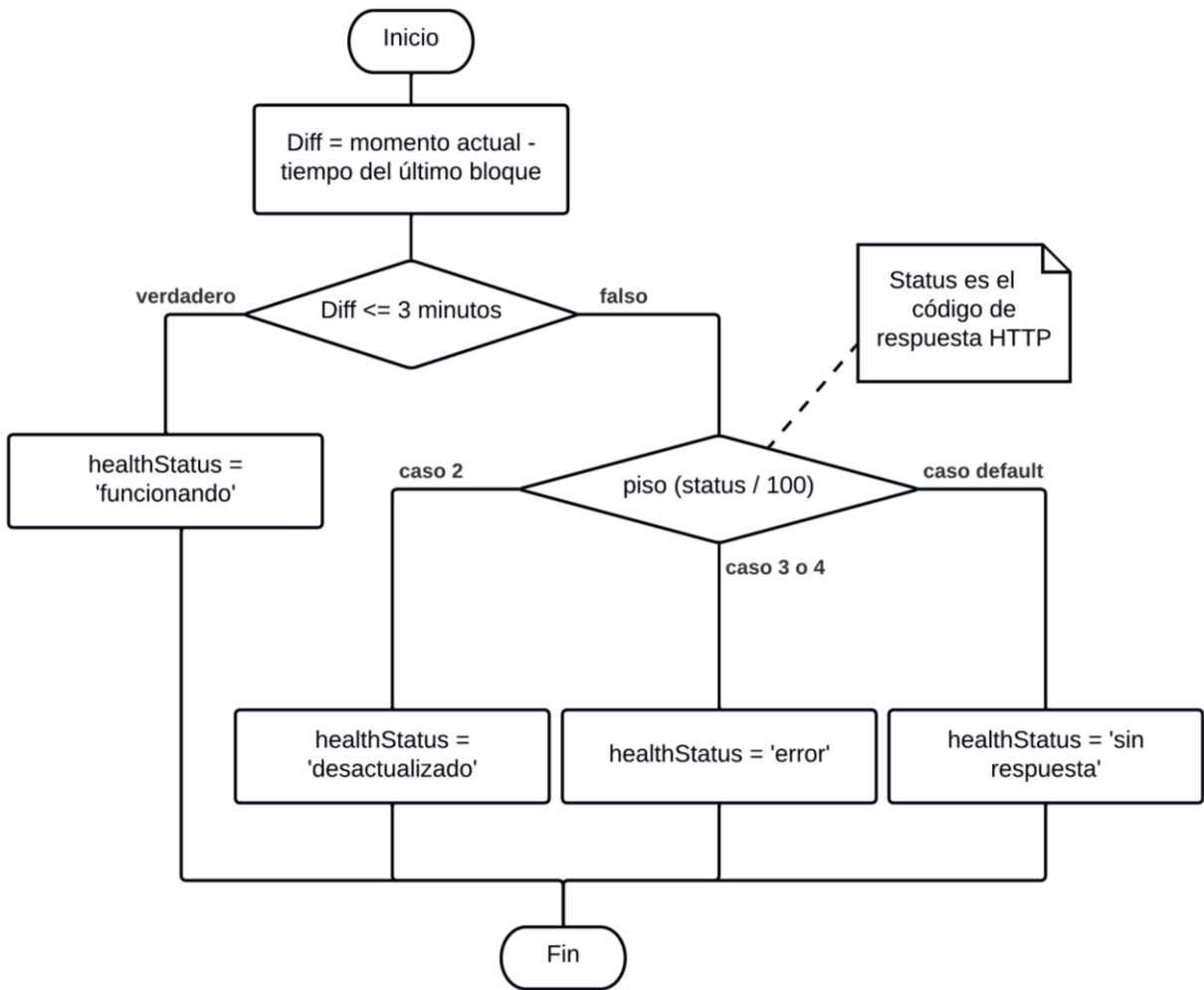


Figura 23. Diagrama de flujo. Estado de salud del endpoint.

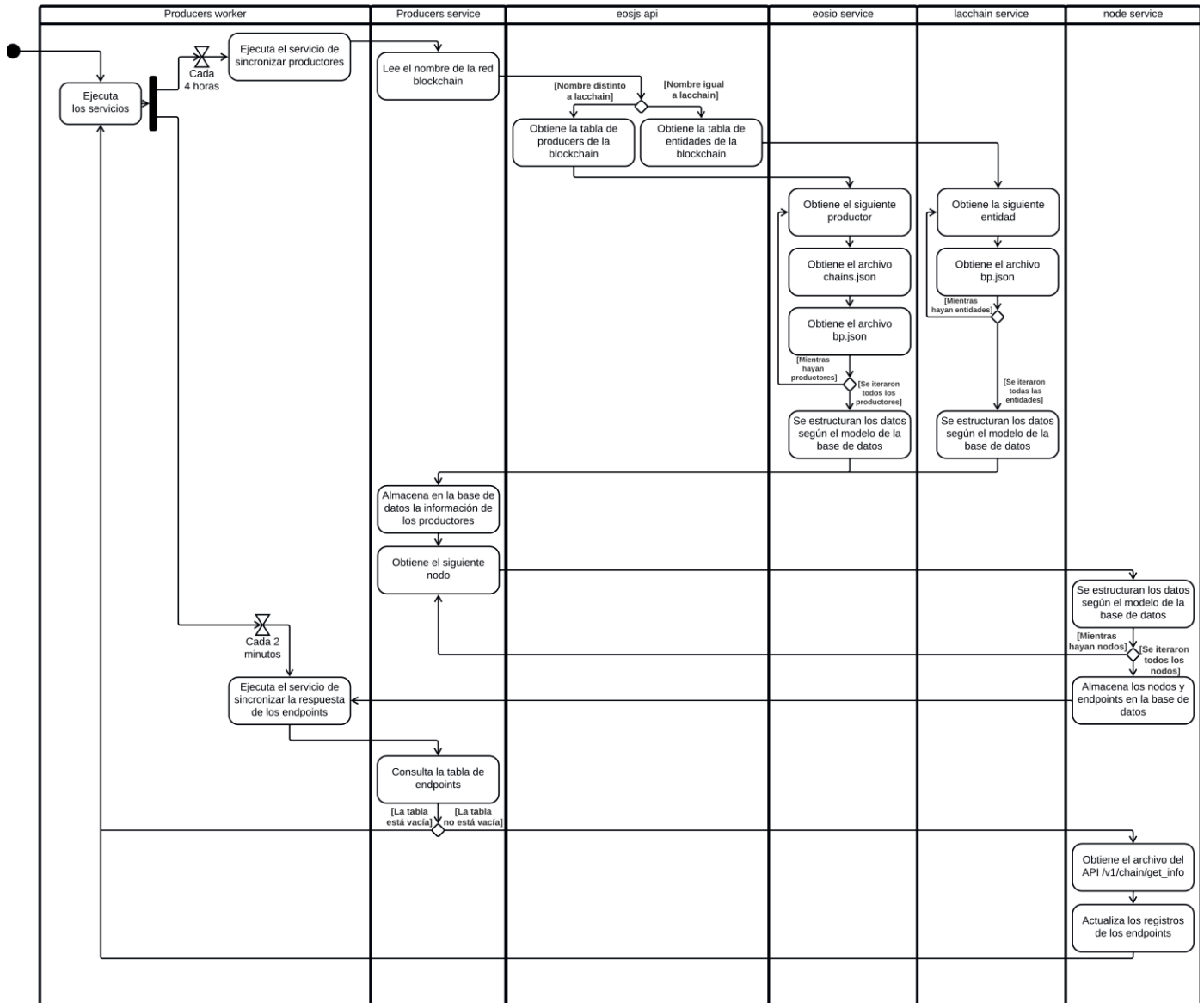


Figura 24. Diagrama de actividad con el servicio de producers.

6.5. Diseño de bases de datos

En esta sección se explicará el diseño de la base de datos implementado para agregar la funcionalidad de determinar el estado de salud de los nodos. Antes de empezar, al ser un proyecto existente el diseño ya estaba creado, sin embargo, el mismo constaba únicamente de la tabla producer para almacenar la información de los productores de bloques, incluyendo el archivo BP JSON completo con todos los datos de los nodos y sus endpoints.

A pesar, de que esta propuesta es funcional para mostrar los datos, se determinó que para la implementación del nodo de salud era poco conveniente, debido a que, las consultas tendrían que realizarse sobre los atributos del objeto JSON en la base de datos, así que se implementó un cambio en búsqueda de facilitar la consulta y actualización de los datos para construir el sensor. Por lo que, el cambio puede ser visto en el Modelo Entidad-Relación en la Figura 25. Modelo Entidad-Relación de la base de datos. y el modelo lógico en la Figura 26. Modelo lógico de la base de datos.

Primero, se decidió normalizar el campo del BP JSON, pero, siempre considerando que la arquitectura del sistema necesita que si los datos son eliminados en algún momento puedan ser recuperados al consultar la blockchain y mantener esta última como la fuente de la verdad.

Entonces, se siguió las primeras formas normales, no obstante, en ciertos casos se omitieron, por esto se crea una nueva tabla para los nodos de los productores y otra para sus endpoints, siguiendo la primera forma también tendría que dividirse la ubicación de los nodos en una tabla y hacer lo mismo con el país, aunque, de momento para la nueva funcionalidad no se requiere filtrar por esta información.

Por lo anterior, en la entidad de nodo se guarda la ubicación como un objeto de tipo JSONB, como el nodo puede tener más de un tipo ese campo es multivaluado y por último los nodos pueden contener más información, pero, esto aplica solo a ciertos casos como, por ejemplo, cuando es un nodo de tipo query. Por ello, se agrega una tabla que contiene la versión del protocolo que ejecuta el nodo y las características que ofrece el API. Estos cambios se realizaron para evitar datos nulos o vacíos, ya que, si hay pocos nodos con esta información adicional se desaprovecha el espacio.

Por último, para los endpoints se persiste su tipo, valor, fecha de creación y actualización, con el valor es posible realizar la petición y guardar el tiempo del último bloque junto la respuesta de la solicitud. Al igual que como fue explicado antes, para no agregar más complejidad de la necesaria al modelo, el tipo se mantiene como un string y la respuesta como un objeto.

Por otro lado, mediante Hasura se construyó una RESTful API para devolver los datos de los productores de bloques con sus BP JSONs y de esta manera pueda ser consultada por otros sistemas,

de esta manera se evita volver a construir la lógica para obtener dichos datos y se podrían reducir las consultas directas a la blockchain.

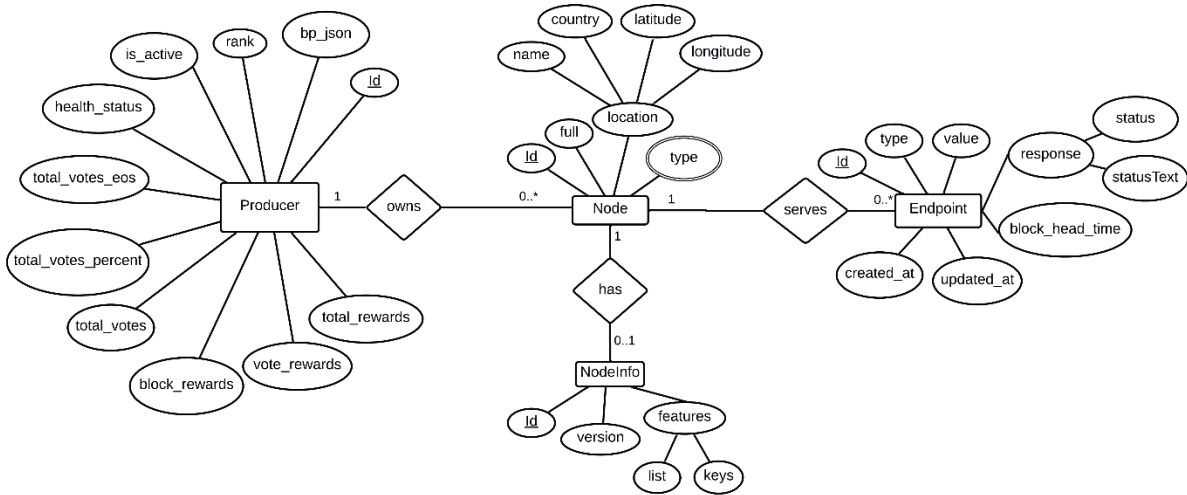


Figura 25. Modelo Entidad-Relación de la base de datos.

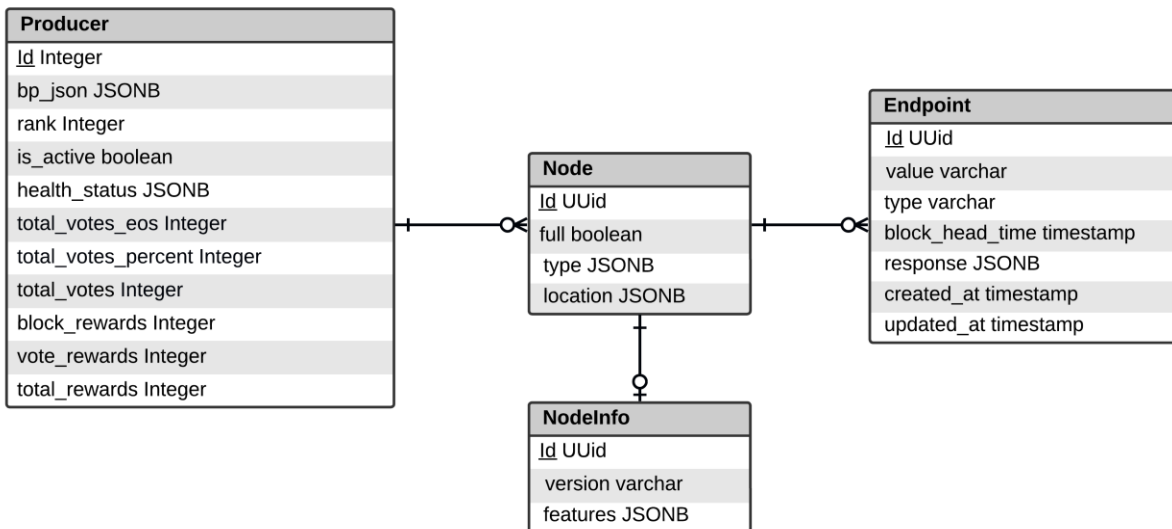


Figura 26. Modelo lógico de la base de datos.

7. Conclusiones

7.1. Cumplimiento de los objetivos

Se logró completar el objetivo general del proyecto al 100%, el desglose del cumplimiento de los objetivos específicos se encuentra en la Tabla 3. Cumplimiento de objetivos, dichos objetivos estaban sujetos a cambios gracias al desarrollar bajo una metodología agile, por lo que, al final se decidió priorizar la reparación de errores en el sistema, brindar mantenimiento al código, efectuar cambios visuales para mejorar la experiencia de usuario e implementar el sensor de salud de los nodos.

Por el motivo anterior, la funcionalidad de integrar la puntuación de EOS Rate a la información de cada productor se omitió y del mismo modo, la implementación del reporte estadístico de los nodos.

Objetivo	% de Cumplimiento	Justificación
Desarrollar una solución que liste, filtre y realice un reporte estadístico de los nodos mediante determinar su estado de salud para facilitar el acceso a la información.	90 %	Se logró desarrollar una solución que lista y filtra los endpoints y nodos con su sensor de estado de salud, se permite filtrar la lista de endpoints para obtener aquellos que se encuentran saludables. Además, de brindar una opción para copiarla lista fácilmente y se construyó un API para obtener dichos datos. Del mismo modo, se implementó en la página de nodos se lista APIs y servicios que soporta cada nodo. Sin embargo, al priorizar los otros objetivos, se presentó falta de tiempo para implementar el reporte estadístico.
Desplegar más información de los nodos mediante consultas a los archivos BP JSON y al sistema EOS Rate en la búsqueda de incrementar la transparencia en la red.	90 %	Se aumentó la cantidad de información que se muestra de los blocks producers según el estándar del BP JSON, en las páginas de los productores de bloques, nodos y endpoints. También, se efectuaron cambios en las interfaces gráficas de cómo se despliega esta información para facilitar su lectura y aprovechar de mejor manera el espacio.

		No obstante, por falta de tiempo la integración con la puntuación de EOS Rate no fue implementada.
Implementar la validación del ingreso de datos en los diferentes formularios del frontend para evitar información incompleta o incorrecta.	100 %	Se implementó exitosamente la validación de datos en la página de cuentas y contratos, y en el formulario del generador de BPJSON, evitando que datos con un tipo o formato incorrecto no sean aceptados.
Efectuar mejoras sobre el código existente para mostrar la información sin necesidad de tener que refrescar la página y emplear mejoras en SEO para mejorar el posicionamiento del sitio web en las listas de buscadores.	100 %	Se realizaron diversas mejoras para evitar tener que refrescar la página para obtener la información actualizada, como en la sección de cuentas y contratos, además, de la página con la lista de endpoints que automáticamente solicita la última comprobación del estado de salud.
Realizar Pruebas y Mantenimiento de código para reparar los errores que se encuentren y validar el funcionamiento correcto del sistema.	100 %	Las pruebas se enfocaron en la correctitud y completitud funcional, con estas se lograron detectar y corregir errores en el código existente y validar su funcionamiento correcto. Además, en componentes de React había repetición de código para la misma lógica, por lo que, se efectuó refactorización en varias partes del código para construir componentes reutilizables.

Tabla 3. Cumplimiento de objetivos

7.2. Productos entregados

El porcentaje de cumplimiento se despliega en la Tabla 4. Productos entregados.

Producto	% de Cumplimiento	Justificación
Código fuente	96 %	El software se encuentra en el repositorio de GitHub de Edenia bajo la dirección: https://github.com/edenia/antelope-tools .
Casos de uso	100 %	Se entregaron los casos de uso de la solución de sensor de salud y llenado del formulario para la generación de un BP JSON.

Diagramas de ER y lógico del modelo de la base de datos	100 %	Se realizaron los diagramas del modelo de la base de datos utilizado para la implementación del sensor de salud de los nodos.
Diagramas de componentes, actividad y flujo	100 %	Se realizó el diagrama de componentes de la arquitectura del sistema y diagramas UML relacionados con la lógica de la implementación del sensor de salud de los nodos.
Guiones de prueba	100 %	Se entregaron los guiones de prueba indicando los atributos de calidad según el ISO 25010 y los diferentes escenarios para los casos de prueba.
Demo	100 %	Se realiza un video mostrando los principales cambios y funcionalidades agregadas por el estudiante.

Tabla 4. Productos entregados

7.3. Conclusiones del proyecto

Cuando se desarrolla con tecnologías que se desconocen la investigación y el experimentar son procesos fundamentales para llevar a cabo una buena implementación que aproveche las ventajas que ofrecen. Por esto, al comprender conceptos básicos y el contexto de las redes blockchain con protocolo EOSIO junto con conocimientos previos se logró satisfactoriamente implementar un sensor de salud para los nodos mediante consultas HTTP.

Al trabajar en un sistema existente, es necesario utilizar cierto tiempo al inicio del proyecto para familiarizarse con este, es importante que exista una buena documentación técnica y se siga un estándar para facilitar la comprensión e integrarse más rápido al proyecto.

Del mismo modo, conociendo la arquitectura del sistema permite tener una visión general de su funcionamiento y de esta manera diseñar software para una nueva implementación o cambios siguiendo la arquitectura.

El patrón arquitectónico permite entender de forma más fácil la arquitectura del sistema, en el caso de demuxjs es interesante como este aprovecha conceptos de otros patrones arquitectónicos para proponer una implementación que considere como funciona la tecnología blockchain.

Por último, ahora con los cambios implementados y el sensor de salud, los usuarios del sistema pueden obtener más información de los productores de bloques y consultar fácilmente si los endpoints de los nodos están funcionando, lo que puede fortalecer la transparencia en las redes blockchain disponibles en el sistema.

7.4. Recomendaciones

Tal y como fue planeado, una buena funcionalidad que podría agregarse es la de incluir la puntuación de la comunidad por medio del sistema EOS Rate, para brindar un punto de vista subjetivo que pueda ser comparado con la información objetiva de los productores de bloques y así puedan darse mejores elecciones en las votaciones.

Del mismo modo, para robustecer la información objetiva se recomienda implementar el reporte estadístico del estado de salud de los endpoints. Asimismo, se podrían aplicar mejoras en la carga de los BP JSON de los productores de bloques para evitar la pérdida de datos.

7.5. Trabajo a futuro

Con respecto al trabajo a futuro, el sistema todavía necesita que se realice refactorización en ciertos componentes que poseen la misma lógica o una similar, asimismo, aún se pueden realizar mejoras a nivel UI/UX para facilitar al usuario utilizar el sistema y consultar la información de manera cómoda y sencilla.

Del mismo modo, ya que se implementó el sensor de salud de los endpoints y se construyó un API para obtener los datos de los productores de bloques sería útil realizar pruebas de carga y tiempos de respuesta para determinar los límites que soporta el sistema y de ser necesario llevar a cabo optimizaciones en la obtención y procesamiento de los datos.

Además, sería de gran utilidad implementar un sensor de salud para nodos con endpoints p2p, de manera similar a lo realizado con los endpoints de tipo API, para brindar más información del funcionamiento actualizado de los nodos de tipo seed.

7.6. Experiencias

Respecto a las experiencias, se identifica que gracias a la práctica en Edenia se logró fortalecer diferentes habilidades blandas como el trabajo en equipo al colaborar con los otros desarrolladores y empleados de otras áreas como diseñadores UI/UX y administrativos, además de utilizar en ocasiones técnicas como pair programming.

Asimismo, la práctica fue realizada de forma remota, por lo que, se fortaleció habilidades como la administración del tiempo y adaptabilidad, al seguir un horario laboral establecido, reuniones de ceremonias de scrum como las reuniones diarias, de planificación y demos, también al seguir las convenciones y estándares empleados en la empresa como el formato de los commits, pull requests y del código.

8. Glosario de términos

Antelope: Protocolo de blockchain que se desprende de EOSIO, debido a diferencias entre la comunidad y la empresa responsable de brindar mantenimiento a EOSIO.

API: Conjunto de funciones y procedimientos que permiten la comunicación entre programas mediante dicha interfaz.

Ataque del 51%: Se refiere cuando un grupo tiene el 51% o más del control sobre la red, debido a que son mayoría y controlarán que bloques son válidos y cuáles no.

Blockchain: Es un registro de transacciones agrupadas en bloques, los cuales están vinculados o encadenados mediante hashes logrando trazabilidad e inmutabilidad en los bloques. (EOS Costa Rica, 2020).

BP JSON: Archivo en formato JSON que brindan los productores de bloques para la consulta de su información siguiendo el estándar definido por EOS RIO (2018). Donde se indica cuáles datos pueden incluir como nombre de la organización, sitio web, contacto de correo electrónico, ubicación, nodos, entre otros.

Chain id: Hash criptográfico que representa un identificador para una blockchain. (EOS Costa Rica, 2020).

Chain json: Archivo en formato JSON que contiene las direcciones URL donde se pueden obtener los archivos BP JSON. (EOS RIO, 2018).

Contrato inteligente: Corresponde a un programa que es ejecutado sobre la infraestructura de la blockchain.

Cuenta: En el protocolo EOSIO se requiere de una cuenta para poder transferir o enviar cualquier transacción válida en la blockchain. La cuenta es una cadena de caracteres alfanuméricos que es legible para humanos. (EOS Costa Rica, 2020).

Cuenta eosio: Corresponde a la cuenta principal en las blockchains basadas en EOSIO.

Delegated Proof of Stake: Mecanismo de consenso utilizado en las redes EOSIO, donde quienes mantienen tokens en el blockchain pueden seleccionar los productores de bloques a través de un sistema de aprobación continua. (EOS Costa Rica, 2020).

Endpoint: Corresponde a la dirección para interactuar con un nodo, donde los endpoints pueden ser de tipo API, SSL o peer-to-peer.

EOS: corresponde a la primera red pública lanzada sobre EOSIO en su primera versión.

EOSIO: Software para desarrollar blockchain personalizables.

EOSJS: API que permite consultar los endpoints de los nodos y tablas de contratos inteligentes de la blockchain.

Estado de salud de nodo: El estado de salud indica cuál es el estado del nodo respecto al de la blockchain, es decir, si se encuentra sincronizado a la blockchain, a los bloques reversibles o a los bloques irreversibles.

Explorador de bloques: Un explorador de bloques es un sistema web que permite a un usuario buscar y navegar por los bloques de una blockchain.

GraphQL: Es un lenguaje de consultas de API's. GraphQL proporciona una descripción completa y comprensible de los datos de su API, da a los clientes el poder de pedir exactamente lo que necesitan

y nada más, hace que las APIs evolucionen con el tiempo y permite potentes herramientas para desarrolladores. (EOS Costa Rica, 2020).

Hapi: Un servicio de backend para la lógica de negocio personalizada, integrada con Hasura, utilizando acciones. (EOS Costa Rica, 2020).

Hasura: Un servidor GraphQL ultrarrápido que le ofrece una API GraphQL en tiempo real sobre Postgres. (EOS Costa Rica, 2020).

Head Block: Corresponde al último bloque del que tiene registro un nodo.

HTTP: Protocolo para la transferencia de hipertexto.

JSON: Del inglés JavaScript Object Notation, es un formato para la estructuración de datos fácil de leer.

Mainnet: Red blockchain principal desplegada a producción.

Mecanismo de consenso: Algoritmo que permite determinar cuál es la verdad absoluta en la blockchain.

Nodos: Son el pilar fundamental de la red blockchain, ya que, intercambian información del último estado de la red para mantener los demás nodos actualizados. (EOS Costa Rica, 2020).

Pair Programming: Consiste en una metodología donde dos personas o más, desarrollan software de manera conjunta donde una de las personas escribe el código y el resto se encarga de supervisar y guiarlo.

Productores de bloques: Los productores de bloques o block producers, son entidades descentralizadas de las redes EOSIO (EOS Costa Rica, 2020) donde las primeras 21 con más votos se encargan de construir los bloques.

Protocolo EOSIO: Protocolo para la construcción de blockchains.

Pull Request: Un pull request es una manera de solicitar que se agreguen los cambios de una rama git a otra.

SEO: Optimización de motores de búsqueda hace referencia a características que posee el sitio web para mejorar su posicionamiento en los resultados de búsquedas.

Tabla: Corresponde a la abstracción donde los contratos inteligentes almacenan los datos.

Testnet: Una red testnet es una instanciación de la blockchain EOSIO que está destinada a las pruebas. Por lo general, el token nativo no tiene valor y se les regala a los desarrolladores para que puedan hacer pruebas en la red. (EOS Costa Rica, 2020).

Top 21: Corresponde a los primeros 21 productores de bloques con mayor de cantidad de votos, por lo que, son los encargados de producir y validar los bloques.

Transparencia: Hace referencia a actuar de manera honesta, rindiendo cuentas.

UI/UX: Corresponde al diseño de interfaces de usuario buscando una buena experiencia de usuario.

Wallet: Las billeteras son clientes que almacenan llaves privadas asociadas con los permisos de una o más cuentas. Idealmente, una billetera tiene un estado bloqueado (encriptado) y desbloqueado (sin encriptar) que está protegido por una contraseña de alta entropía. (EOS Costa Rica, 2020).

9. Referencias

Edenia. (2020). *Antelope + EOSIO Blockchain Network Monitor and Infrastructure Dashboard*.

<https://antelope.tools/>

Edenia. *Edenia: Blockchain nodes | Bare metal servers | Web3 dApps development*. (2022).

Edenia. <https://edenia.com/>

EOS Costa Rica. (2021a). *Desarrollo blockchain empresarial | EOSIO*. EOS Costa Rica.

<https://es.eoscostarica.io/>

EOS Costa Rica. (2021b). *EOS Costa Rica Blog: EOSIO Dashboard*. Official Launch of EOSIO

Dashboard: EOSIO Networks Data Visualization. <https://eoscostarica.io/blog/eosio-dashboard-launch/>

EOS Costa Rica. (2021c). *GitHub - edenia/antelope-tools: Network and Infrastructure Monitor for Antelope + EOSIO blockchains*. GitHub. <https://github.com/edenia/antelope-tools>

EOS Costa Rica. (2018). *GitHub - eoscostarica/eos-rate: EOS Block Producer Rating Portal*
:star2: GitHub. <https://github.com/eoscostarica/eos-rate>

EOS Costa Rica. (19 febrero del 2022). *Mecanismo de Consenso | EOS Costa Rica | Guías para desarrolladores*. EOS Costa Rica. <https://guias.eoscostarica.io/docs/aprender-eosio/mecanismo-de-consenso/>

EOS RIO. (2018). EOSIO BP Information Standard. GitHub. <https://github.com/eosrio/bp-info-standard>

Minuta de Reunión – Visita de presentación final

Nombre del Estudiante: Helbertson Torres Mora
Nombre del Profesor: Luis A. Montoya Poitevien
Consecutivo: 006

Visita No: 4 Fecha: 14 de noviembre del 2022
Hora de inicio: 4:00 p.m. Hora de finalización: 4:50 p.m.

TIPO DE REUNIÓN:

- () Presencial
() Telefónica
(X) Chat – Video Conferencia
() Otro. Especifique: _____

CONVOCADOS:

Luis A. Montoya Poitevien – Profesor Asesor
Michael Castillo – Contraparte de la empresa Sistemas Edenia Internacional S.A.
Helbertson Torres Mora – Estudiante del curso de Práctica Profesional

Se realizó una reunión para la presentación del avance del proyecto alcanzado durante el período de práctica profesional, dicha reunión mediante la plataforma Zoom.

PUNTOS TRATADOS Y ACUERDOS:

1. Presentación final por parte del estudiante exponiendo su avance en el proyecto del curso de práctica profesional.

Compromiso	Responsable
a. El estudiante realizará el debido respaldo del informe, minutas y productos generados en el sitio del Tec Digital.	Estudiante

Firmas:

LUIS
ARTURO
MONTROYA
POITEVIEN
(FIRMA)

Firmado digitalmente por
LUIS ARTURO
MONTROYA
POITEVIEN (FIRMA)
Fecha: 2022.11.17
12:42:57 -06'00'

Profesor
Luis A. Montoya Poitevien



Estudiante
Helbertson Torres Mora



Empresa Sistemas Edenia
Internacional S.A.
Michael Castillo