ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA EN SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN

SISTEMA PARA EL CONTROL FINANCIERO A LA POLÍTICA BASADO EN DLT

DISEÑO Y DESARROLLO DEL PROTOTIPO DEL SISTEMA DE INTERCONEXIÓN DE DATOS PARA CONTROL DEL FINANCIAMIENTO A LA POLÍTICA

TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO

REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN CIENCIAS DE LA COMPUTACIÓN

VÍCTOR IVÁN MAIGUASHCA CUEVA

victor.maiguashca@epn.edu.ec

DIRECTOR: LUIS ENRIQUE MAFLA GALLEGOS

enrique.mafla@epn.edu.ec

QUITO, FEBRERO 2023

CERTIFICACIONES

Yo, Víctor Iván Maiguashca Cueva, declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

_____

VÍCTOR IVÁN MAIGUASHCA CUEVA

Certifico que el presente trabajo de integración curricular fue desarrollado por Víctor Iván Maiguashca Cueva, bajo mi supervisión.

_____

LUIS ENRIQUE MAFLA GALLEGOS

DIRECTOR

Certificamos que revisamos el presente trabajo de integración curricular.

_____	_____
NOMBRE_REVISOR1	**NOMBRE_REVISOR2**
REVISOR1 DEL TRABAJO DE INTEGRACIÓN CURRICULAR	**REVISOR2 DEL TRABAJO DE INTEGRACIÓN CURRICULAR**

DECLARACIÓN DE AUTORÍA

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el (los) producto(s) resultante(s) del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.

VÍCTOR IVÁN MAIGUASHCA CUEVA

LUIS ENRIQUE MAFLA GALLEGOS

DEDICATORIA

Este trabajo se lo dedico a las personas que son importantes en mi vida y que han estado en los momentos precisos de mi vida acompañándome en este largo viaje de vida:

A mis padres, Byron y Nancy, que han sabido guiarme desde mis primeros pasos hasta este último que será mi graduación y que espero continúen a mi lado para guiarme en las siguientes etapas de mi vida.

A mi hermano, Byron, que, aunque parece que me hace de menos lo que ha hecho es entrenarme para soportar los momentos difíciles de la vida.

A mi mejor amigo, José que siempre ha estado dándome ánimo para continuar, para seguir adelante y no rendirme.

A todos con quien he compartido mis batallas diarias, compartido sentimientos desde tener el corazón destrozado hasta la mayor de las felicidades.

AGRADECIMIENTO

Quisiera extender un agradecimiento a todas las personas, lugares y circunstancias que han influido en mí, ya que gracias a ellas he llegado a ser quien soy, a quienes han estado conmigo a lo largo de estos largos años universitarios.

A Aaron, con quien nos lucimos con los mejores proyectos hechos a última hora.

A Boris, quien no solo me ayudaba con explicaciones sino también consiguiendo “las mafias”.

A Jefferson, con quien estuvimos perdidos luego del cambio de carrera en una facultad totalmente nueva.

También quiero extender mis más sinceros agradecimientos a mi tutor, Luis Enrique Mafla Gallegos, por su paciencia y gracias a quien pude realizar este Trabajo de Integración Curricular.

ÍNDICE DE CONTENIDO

Contenido

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

RESUMEN

En este Trabajo de Integración Curricular se realiza la implementación de un sistema de interconexión de datos para el control financiero a las organizaciones políticas durante las campañas electorales, de acuerdo a lo dispuesto en el Código de la Democracia. La metodología de este trabajo se estructura en tres fases: análisis, diseño y prototipo. En el análisis se obtienen los requerimientos de este sistema a partir de leyes y normativas. Dichos requerimientos fueron organizados en tres capas: infraestructura física, middleware, aplicación. La fase de diseño sigue la misma estructura del análisis y se presenta el diseño de la infraestructura física, el diseño de middleware y el diseño de la aplicación. Para el diseño del middleware y diseño de la aplicación se utiliza la plataforma de software libre Hyperledger Fabric que responde a los requerimientos obtenidos en la fase de análisis mediante el uso de tecnología de contabilidad distribuida y el uso de cadenas de bloques. El prototipo sigue la misma estructura del análisis y diseño, su implementación y pruebas están basadas en la red de prueba de Hyperledger Fabric, Fabric Samples. El diseño, prototipo y pruebas realizadas en el presente Trabajo de Integración Curricular, demuestran la factibilidad de implementar un Sistema de Interconexión de datos misma que asegura la integridad del financiamiento y gasto electoral, de acuerdo a los requerimientos legales y normativos vigentes.

PALABRAS CLAVE: Control financiero a la política, Hyperledger Fabric, Fabric-Samples, red de cadenas de bloques permisionadas.

ABSTRACT

In this curricular integration work, the implementation of a data interconnection system for oversight of political campaign financing during election campaigns is carried out according to the provisions of the Democracy Code. The methodology of this work is structured in three phases: analysis, design, and prototype. In the analysis, the requirements of this system are obtained from laws and regulations. These requirements were organized into three layers: physical infrastructure, middleware, application. The design phase follows the same structure as the analysis and it presents the design of the physical infrastructure, the design of the middleware, and the design of the application. For the middleware and application design, the Hyperledger Fabric open-source software platform is used, which responds to the requirements obtained in the analysis phase through the use of distributed ledger technology and the use of blockchain. The prototype follows the same structure as the analysis and design, its implementation and tests are based on the Hyperledger Fabric test network, Fabric-Samples. The design, prototype, and testing carried out in this Curricular Integration Work demonstrate the feasibility of implementing a data interconnection system that ensures the integrity of electoral financing and spending, in accordance with current legal and regulatory requirements.

KEYWORDS: Oversight of political campaign financing, Hyperledger Fabric, permissioned blockchain network.

1 INTRODUCCIÓN

El Código de la Democracia dispone la implementación de un sistema de control para el financiamiento de las campañas políticas. En dicho sistema deben participar las distintas entidades de control del gobierno central del país, con el objetivo de verificar la legitimidad del origen del financiamiento que reciben los partidos políticos [1]. En este trabajo de integración curricular (TIC) se desarrolla una solución para la implementación del mencionado sistema de interconexión. Para ello se utiliza una tecnología de contabilidad distribuida, de control descentralizado, mediante la cual, se implementa dicho control del financiamiento de las campañas políticas.

La metodología de este TIC está basada en el análisis, diseño e implementación del prototipo de este mencionado sistema de control. En la fase de análisis se extrae los requerimientos mediante el análisis de los artículos del Código de la Democracia [1], Reglamento para el Control y Fiscalización del Gasto Electoral [2] y de la norma internacional ISO 54001:2019 [3]. De los requerimientos legales y normativos se deduce los requerimientos tecnológicos organizados en tres capas: requerimientos de infraestructura física, middleware y aplicación.

La fase de diseño se organiza usando la misma estructura del análisis. En diseño de la infraestructura física se discute las posibilidades para implementar los requerimientos obtenidos del análisis. El diseño del middleware describe las configuraciones del sistema en términos de Hyperledger Fabric. En el diseño de aplicación se presenta el diseño de las funciones (consultas y transacciones) que implementa el sistema.

Para desarrollar el mencionado sistema de interconexión se usaron tecnologías de cadenas de bloques (*Blockchain*) y contratos inteligentes (S*martcontract*). El TIC fue desarrollado en la plataforma de cadenas de bloques de software libre Hyperledger Fabric [4], la cual proporciona las facilidades para cubrir los requerimientos. Hyperledger Fabric satisface los requerimientos debido a que posee facilidades para asegurar la integridad e inmutabilidad de la información de gasto electoral y para gestionar identidades y autenticación de manera segura.

La implementación del prototipo está basada en la red de prueba de Hyperledger Fabric, Fabric-Samples[1]. Esta implementación consta del contrato inteligente; es decir, de la definición de la estructura de datos y de las transacciones a partir del análisis y diseño

[1] https://github.com/hyperledger/fabric-samples

realizados. Además, en esta implementación se desarrolló una aplicación para visualizar el funcionamiento y ejecutar las pruebas del funcionamiento de las transacciones que maneja el contrato inteligente y la aplicación de esta tecnología de cadena de bloques.

Mediante el desarrollo de este TIC se logró demostrar la factibilidad de implementar una red segura para el control de gasto electoral, misma que garantiza la integridad del financiamiento del gasto electoral. Este TIC, está fundamentado en las principales asignaturas de la carrera de Ingeniería en Ciencias de la Computación.

1.1 Objetivo general

Demostrar los conocimientos, habilidades y actitudes y valores obtenidos en la carrera, mediante el diseño y desarrollo del prototipo de la red de interconexión de datos para control del financiamiento a la política.

1.2 Objetivos específicos

- Transparentar el control del gasto electoral durante las campañas políticas.
- Proporcionar una estructura descentralizada para el control del gasto electoral.
- Automatizar los procesos de control del gasto electoral.
- Diseñar el sistema de interconexión de datos para control del financiamiento a la política, de acuerdo con lo dispuesto en la transitoria general segunda de la Ley.
- Validar el mencionado diseño mediante la construcción y pruebas de un prototipo.

1.3 Alcance

El trabajo de integración curricular utilizará, de forma exclusiva, información disponible de manera pública en Internet. Para el análisis, diseño y construcción del prototipo, no se solicitará ningún tipo de documentos, información o datos adicionales a la información disponible en Internet.

El TIC está basado en la plataforma Hyperledger Fabric. Dicha plataforma incluye las bases de datos para la implementación de las cadenas de bloques y los correspondientes algoritmos de consenso. Para implementar la membresía segura incluye una Infraestructura de llave pública (PKI), Autoridades de Certificación (CA) y servicios de autenticación, además de servicios de comunicación criptográficamente segura al utilizar Seguridad de Capa de Transporte (TLS) e Interfaces de Programación de Aplicaciones (APIs) para el desarrollo de contratos inteligentes y aplicaciones.

El diseño y construcción del prototipo estará basado en cadenas de bloques, para ello se usará la documentación, código y configuraciones disponibles en el sitio Web de

Hyperledger Fabric [5]. Los contratos inteligentes estarán basados en el código disponible en implementaciones de prueba de la plataforma libre Hyperledger Fabric [6].

Para cumplir con el espíritu de integración curricular del trabajo, se realizará el análisis, diseño y construcción del prototipo con un enfoque sistémico, mismo que involucrará a las principales asignaturas de la unidad profesional de la carrera de Ingeniería en Ciencias de la Computación.

1.4 Marco teórico

En esta sección se presenta los fundamentos teóricos que sustentan el desarrollo de este TIC. Se presenta los temas: sistemas distribuidos, tecnologías de contabilidad distribuidas y Hyperledger Fabric.

1.4.1 Sistemas distribuidos

Un sistema distribuido se define como un conjunto de nodos computacionales interconectados mediante una red, que trabajan juntos como un único sistema. Estos nodos coordinan sus acciones a través del intercambio de mensajes para alcanzar los objetivos de dicho sistema [7].

Si bien un sistema distribuido tiene varias ventajas sobre su contraparte, el sistema centralizado, los sistemas distribuidos presentan desafíos que deben ser resueltos para garantizar la confiabilidad del sistema. Los principales desafíos de los sistemas distribuidos son seguridad, comunicación, consenso. Los desafíos de seguridad en un sistema distribuido se presentan al compartir información sensible de manera segura a través de la red de comunicaciones debido a que no se puede conocer con certeza la identidad de los participantes de la red. Los desafíos en el proceso de consenso entre los nodos del sistema se presentan debido a que los nodos funcionan en sus propios espacios de direccionamiento y no existe forma de saber que pasa fuera del mencionado espacio de direccionamiento; es decir, un nodo no puede conocer de manera confiable el estado en que se encuentran los demás nodos del sistema distribuido. La única forma de conocer dichos estados es a través del intercambio de mensajes, donde se presenta otro desafío del sistema distribuido, la comunicación, ya que la red de comunicaciones no es confiable debido a que los mensajes se pueden perder, llegar duplicados o fuera de orden. La coordinación y el consenso son fundamentales en un sistema distribuido, debido a, que se trata de una sola computación, todos los nodos deben coordinar entre sí para lograr una

solución correcta de dicha computación. Además, no existen soluciones absolutas que garanticen la integridad de las computaciones distribuidas [7].

Para enfrentarse a los problemas de consenso existen algoritmos que facilitan el llegar a un acuerdo entre los nodos participantes de un sistema distribuido. Dichos algoritmos de consenso son tolerantes a fallas y caídas, de modo que, proveen confianza a los nodos que conforman un sistema distribuido. Los algoritmos de consenso se encargan de confirmar la correcta ejecución de una transacción procesada por el sistema distribuido, estar de acuerdo con el orden de las transacciones y con los resultados de dichas transacciones [8].

Existen dos tipos de sistemas distribuidos: permisionados y no permisionados. La diferencia entre estos tipos de sistemas distribuidos se radica en los procesos de identificación y autenticación. En los sistemas permisionados se define un consorcio, lo cual es una lista de acceso a nodos con identidades conocidas. En los sistemas no permisionados no existe la privacidad en las transacciones, pero si anonimato en las identidades, es decir, las identidades no se registran en el sistema [9].

A continuación, se presentan ejemplos de algoritmos de consenso usados en sistemas distribuidos permisionados y no permisionados.

Proof of work o prueba de trabajo es un algoritmo de consenso utilizado en sistemas distribuidos no permisionados como Bitcoin. En este algoritmo existen nodos especiales, llamados minadores en Bitcoin, que deciden el orden en que deben ser aprobadas las transacciones del sistema distribuido. Para ello, dichos nodos minadores resuelven un problema computacionalmente costoso (calculo parcial de la función hash sha). El primer nodo minador que resuelve el mencionado problema es el que determina que transacciones y en qué orden deben ser registradas en la cadena de bloques [10].

Por otra parte, Kafka es un algoritmo de consenso utilizado en sistemas distribuidos permisionados como Hyperledger Fabric. Este algoritmo utiliza un sistema de aprobación de transacciones basado en la votación de los participantes. Kafka utiliza el protocolo *Leader and follower*, por lo que, se designa un nodo líder de entre los nodos participantes de la red con el que deben reportarse y mantenerse sincronizados el resto de nodos del sistema. Para que se apruebe una transacción deben estar sincronizados y votar al menos el 51% de los nodos participantes del sistema [11].

1.4.1.1 Seguridad en sistemas distribuidos

Un sistema distribuido debe garantizar la confidencialidad, integridad y disponibilidad (triada CIA) de la información, de acuerdo a los requerimientos institucionales y legales. La integridad es el control que un sistema implementa para evitar la alteración accidental o no autorizada de los datos del sistema garantizando el no repudio y la autenticidad de la información. Confidencialidad es la restricción en el acceso no autorizado a información privada almacenada en el sistema. La disponibilidad se refiere al acceso confiable a la información en el momento que sea requerido por los participantes [12]. La triada CIA debe ser garantizada durante todo el ciclo de vida de la información: adquisición, procesamiento, almacenamiento, comunicación. Durante las mencionadas fases del ciclo de vida de la información existen riesgos para la CIA de la información [7].

Durante la comunicación y almacenamiento de la información se requiere preservar la integridad y confidencialidad de los datos. Para evitar que los datos sean alterados o visualizados por usuarios malintencionados o no autorizados se debe hacer uso de métodos criptográficos [7].

La criptografía comprende el uso de algoritmos de cifrado y descifrado de mensajes y datos almacenados. Dichos algoritmos están basados en el uso de llaves secretas. El objetivo del cifrado de mensajes es ocultar y proteger su contenido de actores no autorizados, durante la comunicación y almacenamiento de la información. Para cifrar la información se utiliza un algoritmo reversible con el objetivo de poder descifrar dicha información.

Existen dos tipos de algoritmos de cifrado, estas son criptografía simétrica y asimétrica. Los algoritmos de criptografía simétrica utilizan una llave secreta compartida entre emisor y receptor, de modo que, pueden cifrar y descifrar los mensajes respectivamente para mantener una comunicación segura a través de un medio de comunicación o almacenamiento inseguro. Los algoritmos criptografía simétrica no requieren de una gran capacidad de procesamiento para realizarse, por lo cual, posee una gran velocidad de cifrado y descifrado. Un ejemplo de algoritmo de criptografía simétrica es AES, el cual, es un algoritmo empleado en protocolos de comunicación y sistema de almacenamiento criptográficos. AES cifra los datos en bloques de 128 bits y utiliza llaves simétricas de 128, 192 y 256 bits [13].

Los algoritmos de criptografía asimétrica utilizan un par de llaves; una llave privada (secreta) y una llave pública. Dicho par de llaves se generan simultáneamente y se encuentran relacionadas entre sí. Para proteger la confidencialidad, la información se cifra con la llave pública del destinatario de la información; mientras que, para proteger la

integridad, la información se cifra con la llave privada del emisor. Estos algoritmos criptográficos son más complejos por lo que requieren de una mayor capacidad de procesamiento y, por lo tanto, el tiempo de cifrado y descifrado es mucho mayor que el empleado por los algoritmos de criptografía simétrica. Un ejemplo de algoritmo de criptografía asimétrica es RSA, el cual, es un algoritmo utilizado para proteger la confidencialidad e integridad de la información, se utiliza también para firmar digitalmente documentos, de modo que, se utiliza la llave privada para agregar a dicho documentos una sección de código que tiene el mismo valor legal que la firma convencional [13].

1.4.1.1.1 PKI

La infraestructura de llave pública, PKI por sus siglas en inglés, es un repositorio de información que relaciona la identidad digital (par de llaves) con su propietario. La PKI facilita la difusión de la información de los propietarios publicada en dicha infraestructura. Para proveer la confianza que requiere está identidad digital, esta debe ser respaldada por una entidad de confianza llamada Autoridad de Certificación, CA. Un certificado digital consiste de la información de identidad y la llave pública de un propietario firmada mediante criptografía asimétrica por una CA, para ello se utiliza la llave privada de dicha CA. Los certificados digitales se representan actualmente en la versión 3 del estándar X.509. La CA posee un certificado raíz, el cual es la, llave publica de la CA firmada por su correspondiente llave privada, también se conoce como certificado auto firmado. Además, de proporcionar certificados digitales, una CA debe conservar un registro de los certificados digitales revocados, es decir, certificados que ya no son válidos debido a su fecha de expiración, reemplazo por nuevos certificados o certificados en desuso [14].

A continuación, se describe mediante la **Figura *1*** el funcionamiento de una infraestructura de llave pública, para un caso de estudio.

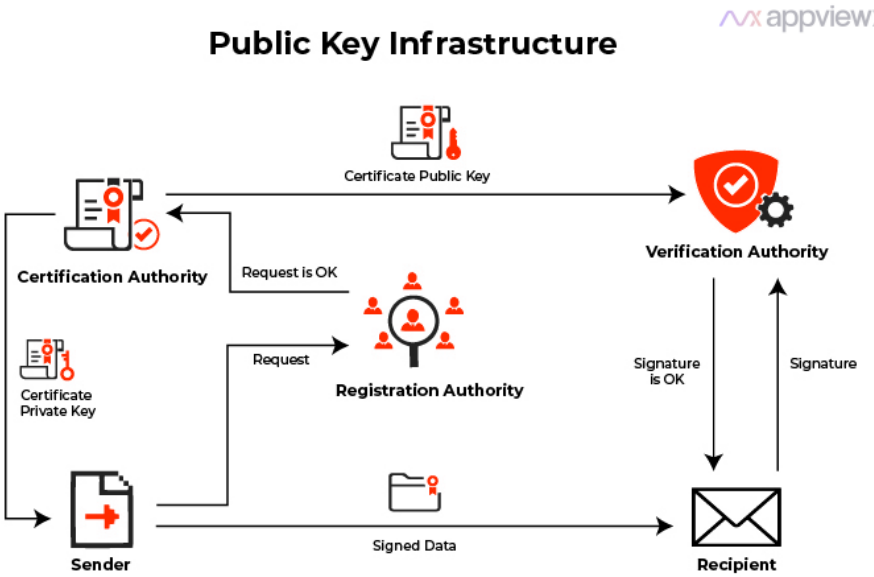

Figura 1: Infraestructura de Llave Pública *[15]*

El remitente requiere enviar un documento con información sensible. Para garantizar la integridad, autenticidad y fecha de creación, dicho documento debe ser firmado digitalmente. Por esta razón, solicita a una autoridad de registro un certificado digital. La autoridad de registro aprueba la solicitud y mediante una autoridad de certificación genera un certificado de clave privada para el usuario. Con este certificado de clave privada el remitente firma los datos y lo envía al destinatario. El destinatario a través de una autoridad de verificación la cual recibió el certificado de clave pública del remitente verifica que el documento que recibió es íntegro y que pertenece al remitente.

1.4.1.1.2 Alta disponibilidad en sistemas distribuidos

Alta disponibilidad en un sistema distribuido es la capacidad que tiene este sistema para mantenerse en funcionamiento en presencia de fallas de hardware, software o incluso de fallas en el suministro de energía eléctrica. Se implementa mediante la redundancia de servidores, bases de datos y redes de comunicaciones. Para mejorar el rendimiento de dichos sistemas redundantes se puede utilizar balanceadores de carga. Los balanceadores de carga supervisan la disponibilidad de los recursos de los servicios redundantes con el objetivo para redirigir las solicitudes para evitar que se agoten los recursos de un servicio en particular, de modo que, todas las solicitudes puedan sean atendidas [16].

1.4.1.2 Comunicación en sistemas distribuidos

La comunicación en sistemas distribuidos se realiza mediante intercambio de mensajes entre los procesos, esto se debe a que no poseen un reloj global y una memoria compartida

entre los nodos del sistema. Para realizar el intercambio de mensajes los nodos se comunican a través de una red de comunicaciones, es decir, un proceso envía una solicitud a un nodo quien recibe el mensaje, lo procesa y envía de vuelta una respuesta. En esta comunicación se debe tomar en cuenta que los nodos de un sistema distribuido presentan una variedad de arquitecturas y de sistemas operativos y por lo tanto estos nodos pueden representar e interpretar los datos de distintas formas [17].

Por esta razón es necesario utilizar protocolos y estándares de comunicación que resuelvan dichos problemas, a continuación, se presentan protocolos y estándares usados en comunicación de sistemas distribuidos.

1.4.1.2.1 Familia de protocolos TCP/IP

La familia de protocolos TCP/IP es un conjunto de protocolos que permiten la transmisión de información a través de una red. Esta familia de protocolos no fue diseñada con la seguridad de la información en mente [18].

Este conjunto de protocolos se divide en 5 capas que pueden ser observadas en la **Figura *2***.

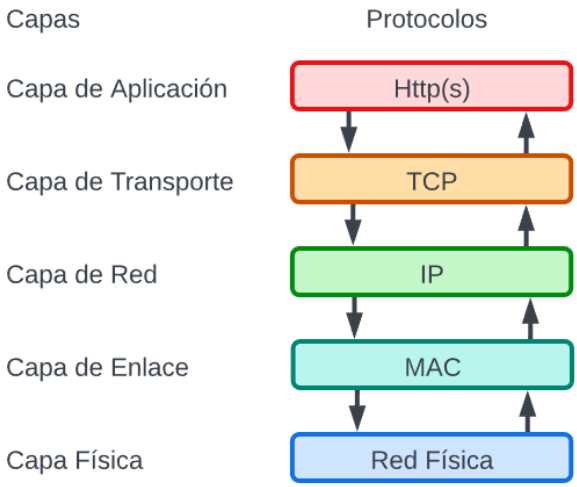

Figura 2: Protocolos por capa en familia de protocolos TCP/IP

La capa física es responsable de la transmisión de datos a través de la red física, incluyendo los medios de transmisión como cables de cobre o fibra óptica. En esta capa la comunicación se realiza mediante unidades de datos llamados bits [18].

La capa de enlace es responsable de la transmisión de tramas de datos de un dispositivo a otro a través del medio físico usando para ello las direcciones MAC de los dispositivos de la red. Se encarga de la corrección de errores en la transmisión [18].

La capa de Internet define el formato y la estructura de los paquetes de datos que se transmiten en la red, incluyendo la dirección IP que identifica a cada dispositivo en la red [18].

La capa de transporte se encarga de empaquetar, fragmentar y numerar los mensajes para enviarlos a través de la red. La capa de transporte usa TCP para asegurar la calidad de servicio de comunicación al verificar que todos los paquetes enviados lleguen a su destino y reconstruyan con éxito el mensaje original [18].

La capa de aplicación proporciona servicios específicos a las aplicaciones, como el protocolo de transferencia de hipertexto HTTP(s) [18].

1.4.1.2.2 RPC

Las llamadas a procedimientos remotos son un paradigma de programación distribuida que permite a una aplicación de un nodo ejecutar un procedimiento en un nodo remoto como si ese último lo hiciera localmente. Convierte las llamadas a procedimientos en mensajes que se envían a través de la red. Un nodo llama a un procedimiento remoto con sus respectivos parámetros para que se ejecuten en el nodo remoto el cual ejecuta el procedimiento y envía una respuesta del procedimiento invocado al nodo que generó la solicitud. RPC es una forma de implementar la programación distribuida permitiendo a los programadores trabajar con modelos similares a los de programación local [17, 7].

XDR, Representación de datos externos, es un estándar para la representación de tipos de datos básicos y para la socialización de estructuras de datos dinámicas, usado en programación distribuida. Debido a que cada sistema computacional por su arquitectura o sistema operativo representa de un modo los tipos de datos; por esta razón es vital tomar en cuenta si se está representando correctamente un dato que fue recibido de una fuente externa [19].

REST, Transferencia de estado representacional, es una arquitectura que permite acceder a un servicio remoto y se puede ejecutar mediante varios lenguajes de programación. Puede, por ejemplo, realizar llamadas a funciones básicas de una base de datos, es decir, crear, leer, modificar y eliminar. Los resultados de las consultas pueden presentarse al usuario en varios formatos siendo Json el más usado [20].

gRPC, Llamada a procedimientos remotos de Google, es una plataforma de código abierto que simplifica el intercambio de mensajes entre clientes y servicios back-end. Un cliente de una aplicación gRPC utiliza una función local que llama a otra función implementada en un nodo remoto; de modo que se presente como un proceso transparente, el cual, abstrae

la comunicación punto a punto, la serialización y la ejecución de funciones entre los nodos. gRPC utiliza HTTP/2 como protocolo de transporte. Además, utiliza una tecnología de código abierto llamada buffers de protocolo [21]. Dichos buffers proporcionan un mecanismo para serializar mensajes estructurados de manera independiente de la plataforma que serializa dichos mensajes. Además, define como deben ser estructurados los datos para posteriormente generar el código fuente que lea y escriba los datos estructurados. Los buffers de protocolo presentan ventajas como compatibilidad con varios lenguajes de programación [21, 22].

1.4.1.2.3 Comunicación segura

Ya que el intercambio de mensajes es el único medio para coordinar un sistema distribuido, es indispensable poder confiar en los nodos que forman parte del sistema. El uso de protocolos de comunicación seguros aumenta el grado de confianza en las comunicaciones. A continuación, se describen dos protocolos de comunicación segura que son utilizados en intranets e Internet.

IPSec, Protocolo de seguridad de internet es un protocolo de la capa de red que provee de seguridad en las comunicaciones a través de intranet. Este protocolo utiliza tanto el cifrado simétrico como asimétrico para cifrar las comunicaciones. Se usa comúnmente para la creación de redes privadas virtuales que interconecten un usuario remoto con una red empresarial [23].

TLS, Seguridad de capa de transporte es un protocolo que permite la comunicación segura entre aplicaciones de tipo cliente – servidor a través de internet. Este protocolo provee seguridad al canal de comunicación entre pares para que exista un flujo de datos ordenado y de confianza [24]. El canal seguro debe cumplir las siguientes propiedades:

- **Autenticación**, el servidor siempre debe estar autenticado mientras que para el cliente la autenticación es opcional. En un sistema distribuido ya que no existe la jerarquía de cliente – servidor por lo que los participantes siempre deben estar autenticados. La autenticación puede ser mediante criptografía asimétrica (RSA), algoritmo de firma digital de curva elíptica (ECDSA), algoritmo de firma digital de la curva de Edwards (EdDSA) o clave simétrica pre-compartida (PSK). La autenticación también se puede realizar mediante una solicitud de certificado [24].
- **Confidencialidad,** los datos que se transportan por el canal luego de establecerse son solo visibles para los pares debido a que estos se encuentran protegidos mediante criptografía [24].

TLS consiste de dos componentes principales:

- **Protocolo de establecimiento,** en el que se autentican los grupos de comunicación, se negocian los parámetros y modos de criptografía, por último, se establecen y comparten las llaves que se usaran para autenticar la comunicación TLS [24].
- **Protocolo de registro,** usa los parámetros definidos por el protocolo de establecimiento para proteger el tráfico en la comunicación entre los pares. Para empezar, se divide el tráfico en series de registros. Posteriormente, son encriptados usando las llaves establecidas. Y finalmente, se envía el tráfico por la red. Por otra parte, el receptor verifica los datos, luego los desencripta y reensambla para entregarlo al cliente en un lenguaje de alto nivel [24].

1.4.2 DLT

Tecnología de contabilidad distribuida, DLT, es una base de datos contable cuyo almacenamiento se encuentra distribuido en nodos ubicados en diferentes localizaciones geográficas. Esta tecnología se usa para distribuir, intercambiar o almacenar datos entre usuarios interconectados mediante redes públicas o privadas [25]. DLT no requiere de una entidad centralizada que supervise o coordine la interacción entre los usuarios [26]. A continuación, describiremos las principales características de esta tecnología.

1.4.2.1 Sistema distribuido con control descentralizado

La tecnología de contabilidad distribuida proporciona beneficios como accesibilidad, eficiencia y confiabilidad en comparación con una tecnología centralizada, esto se debe a que cada miembro puede acceder a los datos ya que se encuentran replicados en todos los nodos participantes. Es decir, pueden acceder localmente a la información que requieren. Al encontrarse distribuido el control de la base de datos se crea una red a prueba de manipulaciones y se asegura una mayor transparencia ya que se informa a cada participante de los cambios en cuanto se presentan.

1.4.2.2 Registros de eventos distribuidos e inmutables

DLT presenta una arquitectura basada en eventos. Un evento hace referencia a la introducción de un cambio en el estado en la base de datos del sistema realizada por el registro de nuevos datos o la modificación de los datos ya existentes en dicho sistema. Para ello se usan detectores de eventos que se encarguen de comunicar los eventos a los

nodos correspondientes del sistema distribuido, para que dichos nodos se encarguen de enviar la respuesta a dichos eventos. Estos registros son inmutables ya que se almacenan una vez se presentan y se verifican en la red y contribuyen a la supervisión del sistema [27].

1.4.2.3 Cadena de bloques

El Blockchain o Cadena de bloques es una tecnología DLT, compartida e inmutable que facilita el registro de transacciones y la trazabilidad del movimiento de los activos de un negocio. Blockchain almacena los datos resultantes de transacciones en bloques de información, conectados de manera secuencial [28].

Para una red de Blockchain todos los nodos participantes son iguales es por eso que se conoce a los nodos participantes como Peer o pares. Esta tecnología de cadena de bloques al ser distribuida reduce el tiempo y los costos para todos los involucrados debido a que se evita un intermediario en las transacciones [28].

1.4.2.3.1 Tipos de redes de cadenas de bloques

De forma general una red de cadena de bloques puede ser de dos tipos:

Red de Blockchain no permisionada, considerada una red de Blockchain publica ya que no existen restricciones para participar en esta red. Un requisito para participar en este tipo de redes es poseer un gran poder computacional. En esta red de Blockchain la identidad de los participantes se mantiene anónima y no existe privacidad en los datos de las transacciones. Al ser pública el funcionamiento es transparente para que cualquier persona revise o audite su funcionamiento lo que proporciona transparencia a la red. Un ejemplo de este tipo de Blockchain es Bitcoin [28].

Red de Blockchain permisionadas, consideradas redes de Blockchain privadas debido a que para participar en esta red se requiere de una entidad que controle el acceso a los datos registrados y que verifique la identidad de los participantes. Un ejemplo de dicha red de Blockchain es Hyperledger Fabric, sobre el cual se realizó la implementación de este TIC [28].

1.4.3 Hyperledger Fabric

Hyperledger Fabric es una plataforma DLT permisionada de código abierto desarrollada por la fundación Linux. Al ser permisionada ofrece privacidad y gobernanza para conseguir confianza y transparencia en la contabilidad de los negocios. [4, 29]

Hyperledger Fabric garantiza la inmutabilidad de las transacciones y datos por medio de las siguientes tecnologías.

- Criptografía de llave pública para la inmutabilidad al codificar el historial de las transacciones y para garantizar la seguridad, confidencialidad e integridad de las comunicaciones [30, 29].
- Infraestructura de llave pública a través autoridades de certificación en cada una de las organizaciones participantes para proporcionar certificados digitales para gestionar de las identidades de los respectivos componentes de la red mediante la cual dichos componentes de la red se identifican y autentican [30, 29].
- Canales mediante los cuales se mantiene la privacidad de los datos al agrupar únicamente a las organizaciones autorizadas para el acceso y modificación de dichos datos [30, 29].
- Datos privados para mantener la privacidad de los datos de las transacciones al compartir hashes como evidencia de las transacciones dentro de un canal de las organizaciones [30, 29].
- Algoritmos de consenso facilitan el ordenamiento de las transacciones

Ya que las transacciones son rastreables e inmutables se crea confianza entre las organizaciones, permitiendo a estas agilizar la tomar decisiones reduciendo costos y riesgos [31].

1.4.3.1 Componentes de una red de Hyperledger Fabric

En esta sección se describe los principales componentes que conforman una red Hyperledger Fabric.

Organizaciones (Organizations), es el conjunto de miembros que posee una organización participante de una red de Hyperledger Fabric. Cada organización de esta red está compuesta por un conjunto de nodos [32]. Este conjunto de nodos posee diferentes funcionalidades especificas las cuales son explicadas a continuación.

Pares (Peer node), es un nodo organizacional que representa la participación de una organización como miembro de un canal de Hyperledger Fabric. La función de este nodo es gestionar el libro mayor y el contrato inteligente. Un Peer puede participar en varios canales dentro de una red, y por cada canal debe gestionar su respectivo libro mayor y contrato inteligente [33].

Ordenadores (Orderer node), es un nodo organizacional que se encarga de distribuir y ordenar la ejecución de las transacciones. El diseño de Hyperledger Fabric recae en algoritmos de ordenamiento y consenso determinísticos con lo cual se asegura que cualquier bloque verificado del libro mayor sea correcto y se encuentre en el orden preciso. Estos algoritmos y de ordenamiento se conocen como políticas de la red. Dichas políticas de la red aseguran que el libro mayor sea idéntico en todos los Peers, es decir, que no se provoquen variaciones del mismo libro como sucede en redes de cadenas de bloques no permisionadas [34].

Autoridades de Certificación CA, son nodos organizacionales encargados de gestionar las identidades digitales de los miembros de la red de Hyperledger Fabric. Es decir, mediante la infraestructura de llave publica emite certificados digitales de tipo certificado raíz a los miembros y certificados de inscripción usuarios de la red [35].

Canales (Channels), son subredes privadas que comunican a un consorcio de organizaciones (dos o más) dentro de la red empresarial. Estos canales se utilizan para ejecutar transacciones privadas y confidenciales de modo que intervengan solo las organizaciones autorizadas. Dentro del alcance de este canal se encuentra un conjunto de organizaciones, un libro mayor, un contrato inteligente, una aplicación que ejecuta el contrato inteligente y un nodo que proporciona el servicio de ordenamiento [36].

Los activos (Assets), son bienes, ya sean estos tangibles o intangibles, que poseen valor. El concepto de Asset es central en Hyperledger Fabric debido a que garantiza la inmutabilidad de transacciones sobre dichos activos. Son representados dentro una red de Hyperledger Fabric por una colección de datos de tipo clave - valor. Estos almacenan el estado final de los cambios aplicados a los Asset mediante transacciones gestionadas por los contratos inteligentes dentro del canal del espacio de nombres [30].

El libro Mayor (Ledger), es un registro inmutable que almacena de manera secuencial los estados resultantes de las transiciones por los que atraviesa el conjunto de Assets. Cada Ledger se encuentra asociado a un Chaincode dentro de un canal. Cada Peer preserva una copia del Ledger de cada canal en el que se encuentra, lo que facilita el proceso de auditoría y resolución de conflictos [37]. El Ledger está compuesto por dos elementos:

- **Estado actual (WorldState),** es una base de datos que presenta (como su nombre lo indica) el estado actual en el que se encuentran los registros almacenados. Contiene las últimas versiones de los registros activos de la base de datos [37].

- **Cadena de bloques (Blockchain),** es un log de transacciones almacena cada estado resultante de las transiciones como una secuencia de bloques. Y cada nuevo bloque creado se encuentra criptográficamente encadenado al anterior [37].

Los contratos inteligentes (Smartcontract), son códigos de programación mediante los que se define la lógica del negocio. Dentro de la terminología de Hyperledger Fabric se conoce a los contratos inteligentes como Chaincode [38]. La estructura de un contrato inteligente consta de tres secciones:

- **Estructura de datos del Asset,** define la estructura de datos de tipo clave - valor en los que se almacena información del Asset en el estado actual de la base de datos. A estos Assets se aplicarán las reglas de lectura o modificación que se definirán en la sección Invoke [38].
- **Init,** construye un conjunto inicial de datos que se presentan a la red y se almacenan en el Ledger de todos los nodos [38].
- **Invoke,** abarca todas las funciones o transacciones que se pueden aplicar a los Assets existentes en el estado actual *WorldState* de la base de datos. Esto incluye las transacciones básicas de una base de datos: crear, leer, modificar y eliminar los respectivos Assets. Existen transacciones propias de Blockchain como lo es la consulta del historial de cambios de un registro. Adicional a esto se pueden agregar funciones específicas del negocio [38].

1.4.3.2 Estructura de la red de Hyperledger Fabric

Para comprender la estructura de una red de Hyperledger Fabric se describirá la **Figura *3***; de este modo, el lector tendrá una visión general de los componentes que conforman una red de Hyperledger Fabric lo cual facilitará la comprensión de los conceptos explicados en la sección 1.4.3.1.

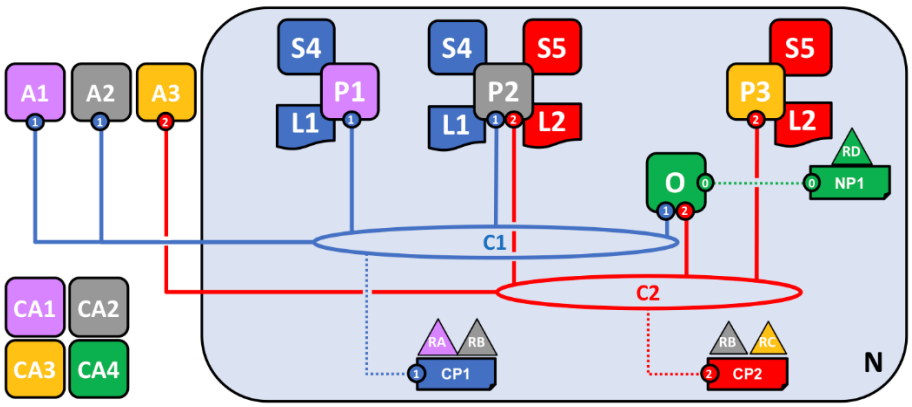

Figura 3: Estructura de una Red de Hyperledger Fabric *[35]*

En esta red ***N*** de Hyperledger Fabric existen cuatro organizaciones. Estas son: **RA**, **RB**, **RC**, **RD**. Cada organización está conformada de diferentes nodos. La organización **RA** está conformada de un nodo de Aplicación **A1**, un Peer **P1** y una autoridad de certificación **CA1**. Esta organización se puede identificar ya que sus elementos se encuentran en color rosado. La organización **RB** está conformada de un nodo de Aplicación **A2**, un Peer **P2** y una autoridad de certificación **CA2**. Esta organización se puede identificar ya que sus elementos se encuentran en color gris. La organización **RC** está conformada de un nodo de Aplicación **A3**, un Peer **P3** y una autoridad de certificación **CA3**. Esta organización se puede identificar ya que sus elementos se encuentran en color amarillo. La organización **RD** está conformada de un nodo de Ordenamiento **O** y una autoridad de certificación **CA4**. Esta organización se puede identificar ya que sus elementos se encuentran en color verde.

En esta red existen canales que interconectan a los nodos de las organizaciones. El canal C1 es un consorcio entre las organizaciones **RA** y **RB**, es decir, en el canal **C1** participan las organizaciones **RA** y **RB** con sus respectivos nodos de Aplicación **A1, A2** y Peers **P1, P2**. El canal utiliza las políticas de canal **CP1**. Los Peers **P1** y **P2** al ser miembros de este canal gestionan el Chaincode **S4** y el Ledger **L1**. El canal posee el servicio de ordenamiento mediante el nodo **O** de la organización **RD**, el cual implementa las políticas **NP1** en el canal.

De un modo similar, el canal **C2** es un consorcio entre las organizaciones **RB** y **RC**, es decir, en el canal **C2** participan las organizaciones **RB** y **RC** con sus respectivos nodos de Aplicación **A2, A3** y Peers **P2** y **P3**. El canal utiliza las políticas de canal **CP2**. Los Peers **P2** y **P3** al ser miembros de este canal gestionan el Chaincode **S5** y el Ledger **L2**. El canal posee el servicio de ordenamiento mediante el nodo **O** de la organización **RD**, el cual implementa las políticas **NP1** en el canal.

2 METODOLOGÍA

En este TIC utilizamos una metodología de desarrollo tradicional que consiste de las fases análisis, diseño, construcción del prototipo y pruebas. En la fase de análisis, en primer lugar, analizamos los requerimientos legales y normativos relacionados al control de gasto electoral. Con los resultados de dicho análisis, definimos los requerimientos tecnológicos del sistema de interconexión de datos en tres macro capas: infraestructura física, middleware y aplicación. En la fase de diseño, se tomó en cuenta los requerimientos tecnológicos y se definió el diseño de cada una de las mencionadas macro capas. Para validar el diseño, se construyó un prototipo basado en la red de prueba de Hyperledger

Fabric. Las pruebas de funcionalidad del prototipo para demostrar su correcto funcionamiento se describen en la sección 3.1.

2.1 Análisis

En esta fase se realiza el análisis de los requerimientos legales y normativos a partir de los cuales se extrae los requerimientos tecnológicos para las respectivas capas que comprenden el sistema de control de datos, definiendo así requerimientos para la infraestructura física, requerimientos del middleware y requerimientos de la aplicación.

2.1.1 Análisis de requerimientos legales y normativos

En esta sección se presenta el análisis los artículos del Código de la Democracia relacionados al control del gasto electoral (Registro Oficial N.º 134), del Reglamento para el Control y Fiscalización del Gasto Electoral (Registro Oficial N.º 345) y de la norma ISO 54001:2019; a partir de los cuales se extrajo los requerimientos para realizar el diseño y la implementación del del prototipo del mencionado sistema de control financiero.

Primero presentamos los textos de la ley, reglamento y norma y luego desarrollamos el análisis correspondiente. Se resaltó las partes de los artículos que contribuyen con información relevante para el desarrollo de este TIC.

2.1.1.1 CÓDIGO DE LA DEMOCRACIA

> Art. 211.1-" ***El Consejo Nacional Electoral desarrollará las herramientas tecnológicas e informáticas necesarias para implementar el Sistema Contable del Financiamiento a la Política****, mismo que será de acceso gratuito a todas* ***las organizaciones y sujetos políticos****, a fin de que* ***registren de manera obligatoria su Plan de Cuentas****.* ***El Consejo Nacional Electoral hará pública la información contable a través de su página oficial de internet y reglamentará la obligatoriedad de cumplimiento de los mecanismos de control del gasto electoral****. El control de gasto se ejecutará y reportará en lapsos continuos de máximo quince días."*. [1]

El CNE debe implementar un sistema contable para las organizaciones políticas en el cual reporten sus ingresos y egresos, con el fin de proporcionar a la ciudadanía la información contable para controlar los gastos de dichas organizaciones políticas. Este articulo implica la necesidad de contar con un sistema informático público para transparentar la información relacionada al gasto electoral. Las herramientas de este sistema facilitaran el registro y

posteriormente la publicación de los datos almacenados en este sistema contable con los que pueda ejecutar el control de gasto electoral.

> **Art. 211.2.-** ***"El Consejo Nacional Electoral se vinculará o desarrollará un sistema de interconexión de datos que permita recabar la información depositada en cualquier organismo público o privado, referente al financiamiento político, sin prejuicio de las limitaciones legales existentes relativas a la privacidad, propiedad intelectual y seguridad integral****."*** [1]

Es necesario automatizar el intercambio de datos para agilizar el control financiero de las organizaciones políticas por parte del CNE. De este modo, el CNE, a través del sistema de interconexión de datos (**SID)** puede acceder a la información que requiere acerca del financiamiento de las organizaciones políticas sin restricciones. Al requerir que se comparta información de otras instituciones implica que el **SID** debe ser distribuido y su control debe ser descentralizado.

> **Art. 211.2.** ***"El sistema de interconexión de datos también reportará incumplimientos a la normativa cuya competencia recaiga en el Servicio de Rentas Internas, autoridades de regulación y control del sistema financiero nacional, Autoridad Nacional de Análisis Financiero y Económico, Contraloría General del Estado, Fiscalía General del Estado y Tribunal Contencioso Electoral. El sistema implementado deberá tener un carácter desconcentrado, permitiendo a las diferentes dependencias administrativas del Consejo Nacional Electoral reportar de manera directa sus hallazgos."*** [1]

Las entidades de control participantes del mencionado **SID** a través de investigaciones, fiscalizaciones y auditorias deben emitir alertas a partir de sus hallazgos y compartirlas a través del mencionado sistema distribuido, de control descentralizado mediante el cual debe informar a las demás entidades de control. Los reportes mencionados en el texto anterior, deben ser en línea y de carácter preventivo, de manera que se presenten con regularidad a los organismos competentes.

Disposición General Décima Primera.

> *"**Todo aporte que supere los diez mil dólares** de los Estados Unidos de América, **se respaldará con el formulario de origen lícito de fondos, conforme a lo exigido dentro del sistema financiero, aprobado por las entidades competentes en coordinación con la Unidad de Análisis Financiero y Económico**."* [1]

Esto implica que el control se realice en línea, con la información de las instituciones directamente relacionadas a dicho control. Con la respectiva presentación y aprobación del formulario correspondiente para aprobar la realización de dicha transacción a favor de una organización política. Las entidades deben utilizar los datos del formulario presentado para realizar la investigación pertinente en sus propias bases de datos acerca de irregularidades del origen de los fondos.

Disposición transitoria Primera

> *"**El Consejo Nacional Electoral, Superintendencias de Bancos y Compañías y las respectivas entidades de control del Gobierno Central, deberán generar un Sistema de Interconexión de Datos** en un plazo de ciento veinte días..."* [1]

Las entidades de control requieren de un sistema que facilite el intercambio de información para agilizar sus procesos. Actualmente, el sistema de control que posee el Consejo Nacional Electoral (CNE) es centralizado y sus procesos no se encuentran automatizados. Para automatizar estos procesos es necesario implementar un sistema distribuido, de operación y control descentralizado para comunicar de manera eficaz a las principales entidades de control de nuestro país.

Disposición transitoria Segunda párrafo 1

> *"**El Consejo Nacional Electoral** en un plazo de tres años**, implementará el Sistema de Interconexión de Datos para control del financiamiento a la política**."* [1]

Este texto aclara y consolida lo dispuesto en los artículos y disposición transitoria antes mencionados. La función de este **SID** es el control del financiamiento a la política. El CNE debe ser la entidad que se ponga al frente del diseño e implementación de este Sistema de Interconexión de Datos para el control del financiamiento a la política (**SIDCFP)**, mediante el cual, puede automatizar los procesos de financiamiento de las organizaciones políticas durante las campañas electorales con el fin de aumentar la transparencia.

Disposición transitoria Segunda párrafo 2

> *"**Las organizaciones políticas registrarán los ingresos y aportes, de manera mensual y en línea**. **Este sistema para el control de financiamiento será de acceso gratuito y público**."* [1]

Las organizaciones políticas tienen el deber de presentar una rendición de cuentas periódica acerca de sus actividades que generen ingresos y del financiamiento que reciben

para realizar su campaña electoral. Por esta razón, este sistema debe implementar de una plataforma en línea para registro periódico de aportes y gastos.

2.1.1.2 REGLAMENTO PARA EL CONTROL Y FISCALIZACIÓN DEL GASTO ELECTORAL

Artículo 1.

> *"Objeto.- Controlar la propaganda y publicidad electoral,* ***fiscalizar el gasto electoral, en lo relativo al monto, origen y destino de los recursos utilizados*** *por los candidatos inscritos en un proceso electoral."* [2]

Actualmente, este control es manual y extemporáneo, Esta falta de automatización provoca que la fiscalización de un proceso electoral en nuestro país tarde un tiempo muy prolongado. De tal manera que, de las elecciones realizadas en año 2019, el proceso de fiscalización no concluye a pesar de que han transcurrido 4 años de su inicio ya que en el CNE solo se encuentran informes de hasta el 2018 [39].

Mediante este artículo, se puede extraer el funcionamiento del mencionado **SIDCFP.** En este artículo se describen los actores y transacciones que intervienen en este **SIDCFP**. Estos son:

Persona, empresa u organización que realice un aporte o contribución a una organización política (**origen**), el cual en secciones posteriores se lo conocerá como "Aportante"**;** Aportes o contribuciones receptadas por una organización política (**monto**), esta transacción en secciones posteriores se conocerá como "Aporte"; Proveedor que proporcione productos o servicios de publicidad a las organizaciones políticas, en secciones posteriores se conocerá como "Proveedor"; El pago entregado por la organización política a un proveedor por sus productos o servicios (**destino**), esta transacción se conocerá en secciones posteriores como "Pago".

Artículo 23

> *"Límite del gasto electoral.- El Pleno del* ***Consejo Nacional Electoral aprobará el límite máximo del gasto para cada proceso electoral****, para lo cual el área respectiva presentará el informe correspondiente."* [2]

Como parte del control financiero que se debe aplicar a las organizaciones políticas, el CNE debe definir un valor máximo de gastos que puede realizar dicha organización política durante la campaña electoral. El máximo de gastos de cada organización política indica también un máximo en la cantidad de ingresos y aportes que puede recibir dicha

organización política. El cumplimiento de estos debe verificarse en la implementación de la aplicación del **SIDCFP.**

Artículo 28

> *"Responsable del manejo económico.- El responsable del manejo económico* ***receptará, administrará y registrará los aportes en especie o numerario, extenderá y suscribirá los correspondientes comprobantes y presentará las cuentas de campaña electoral*** *al Consejo Nacional Electoral en los formularios establecidos para el efecto, responsabilidad que asumirá hasta que se dicte la resolución correspondiente."* [2]

Los comprobantes de recepción de contribuciones y de pagos obtenidos por una organización política durante el ciclo de vida de la campaña electoral deben ser presentados para su registro en este **SIDCFP**. Estos comprobantes deben ser registrados, de modo que, posteriormente puedan ser usados para realizar las respectivas fiscalizaciones de la campaña política.

Artículo 34

> *"Apertura del Registro Único de Contribuyentes y de la cuenta bancaria única electoral.- El responsable del manejo económico en un plazo de diez (10) días contados a partir del día siguiente a la notificación de la calificación de la candidatura obtendrá el* ***Registro Único de Contribuyentes (RUC) por dignidad y jurisdicción****, y abrirá la* ***Cuenta Corriente Bancaria Única Electoral por cada dignidad, binomio, lista y jurisdicción****; documentos que le acreditan y habilitan para receptar aportes, contribuciones y registrar gastos de la campaña electoral."* [2]

Dado que el RUC es un número único, mediante el cual, se puede identificar con exactitud a una organización, a este lo acompaña un nombre llamado Razón Social el cual es el nombre Legal con el que se encuentra registrada dicha organización. Por esta razón, el número de RUC, es un dato fundamental para elaborar el registro de una organización política. Además, el número de cuenta corriente y el nombre de la institución bancaria a la que dicha cuenta pertenece deberán formar parte de los campos que componen el mencionado registro de la organización política. Ya que, también se debe registrar las contribuciones receptadas y gastos realizados, estos valores se deben incluir en dicho registro de la organización política. Entidades de control como el SRI pueden encargarse de verificar que el número de RUC exista y pertenezca a la organización política que lo

registró. Del mismo modo la Superintendencia De Bancos o la UAFE deben validar que el número de cuenta pertenezca a la organización política y verificar la el saldo que posee antes de iniciar el proceso electoral.

Artículo 45

> *"Comprobante de recepción de contribuciones y aportes.- El responsable del manejo económico estará obligado a* ***presentar por cada aporte, el comprobante de recepción de contribuciones y aportes, que deberá contener el número secuencial para control interno****,* ***la identificación plena del aportante*****,** ***sus nombres y apellidos completos, número de cédula de ciudadanía, dirección, teléfono, correo electrónico, el nombre y número de la organización política, social o alianza, así como la dignidad, binomio, lista y jurisdicción a la cual corresponde dicho aporte.****"* [2]

Los mencionados comprobantes deben ser electrónicos, con las seguridades criptográficas correspondientes. Entidades de control como la UAFE o el SRI deben analizar los datos de los aportantes con el objetivo de emitir alertas acerca de los aportes que han realizado. Por tal razón se requiere obtener con exactitud los datos identificativos de los aportantes. Estos datos deben constar como los campos que componen el registro perteneciente a dicho aportante. En el comprobante constan los campos que se deben usar para la transacción de aporte, es decir, los datos del aportante y los datos de la organización política.

2.1.1.3 Norma ISO/TS 54001:2019

Norma ISO 54001: 2019 es una norma internacional que especifica los requisitos que debe cumplir un sistema de gestión de la calidad electoral. El documento de esta norma presenta una metodología de mejora continua PDCA para la implementación de la norma ISO 9001:2015 especificando la relación de los requisitos de los sistemas de gestión de la calidad que se deben aplicar para implementar un sistema de gestión de la calidad electoral.

En el anexo B de la mencionada norma ISO se especifican ocho procesos electorales, sus definiciones y las principales actividades que deben realizar las organizaciones electorales en cada uno de los procesos electorales como parte de un sistema electoral de calidad. En particular para este TIC, se analiza el proceso electoral B.7 para implementación del **SIDCFP**.

ANEXO B (Informativo) Procesos electorales B.7 Fiscalización del financiamiento de campañas electorales

"Este proceso analiza los mecanismos que regulan el marco legal y sus mecanismos de cumplimiento para el financiamiento de campañas electorales dentro de un proceso electoral. ***Las principales actividades para la fiscalización de la financiación de campañas electorales*** *van orientadas a verificar la elegibilidad para* ***recibir financiación****,* ***realizar el seguimiento de fuentes de financiación públicas y privadas, realizar el seguimiento de los gastos****, y el acceso a los medios de comunicación.*

La fiscalización de la financiación de campañas electorales tiene el objetivo de:

a) *disuadir la corrupción, la influencia indebida de intereses especiales, el uso indebido de recursos estatales, la compra de votos y otras formas de fraude electoral;*

b) ***promover la rendición de cuentas, la transparencia y la imparcialidad en el acceso a los recursos financieros disponibles para los candidatos y partidos****;*

c) *promover la competencia justa entre organizaciones políticas y candidatos;"*

d) ***limitar el gasto total en campañas electorales y actividades políticas****;*

e) ***impulsar la transparencia y acceso público a la información sobre la financiación de campañas electorales al establecer requisitos de informes financieros públicos****;*

f) *promover la igualdad de acceso para que los candidatos utilicen los medios de comunicación para los propósitos de la campaña."* [3]

El **SIDCFP** debe ser capaz de almacenar de manera segura, inmutable, los datos de los aportantes a las campañas políticas y de los proveedores de servicios. Estos datos recolectados deberán identificar de manera precisa, tanto a aportantes como a proveedores. Ya que pueden existir fuentes de financiamiento públicas y privadas los datos recolectados de estos pueden variar de modo que se identifique de que tipo de fuente de financiamiento se trata. De este modo, se puede realizar el seguimiento desde la fuente del financiamiento hasta los gastos de un partido político a través de transacciones de aportes recibidas y gastos realizados durante el ciclo de vida de la campaña electoral. Estos datos deben almacenarse de tal modo que faciliten la automatización de las actividades de fiscalización que realicen las respectivas entidades de control.

En esta sección de la norma ISO 54001:2019 se presenta una visión general de las funciones que debe cumplir este **SIDCFP**. Estos aspectos se especifican con una mayor extensión en los artículos del Código de la Democracia y en el Reglamento para el Control y Fiscalización del Gasto Electoral.

2.1.2 Requerimientos tecnológicos

A partir del análisis de las leyes y normas se obtuvo los requerimientos a implementarse en este mencionado sistema de control para el financiamiento a la política. A continuación, se presenta una formalización de los requerimientos organizados en tres macro capas infraestructura física, middleware, aplicación.

2.1.2.1 Requerimientos de Infraestructura física

Para la infraestructura física se consideraron dos aspectos fundamentales para la implementación del **SIDCFP,** estos son los requerimientos funcionales y requerimientos de seguridad. A pesar de que requerimientos como rendimiento son importantes, estos no son tomados en cuenta por el alcance de este TIC.

Requerimientos funcionales

Los requerimientos funcionales se organizan por el tipo de red que se requiere para implementar este **SIDCFP**.

Red distribuida

- Se requiere la implementación de un sistema de interconexión de datos en la que participen todas las entidades de control del gobierno central como miembros del sistema, es decir, un sistema de red distribuida.
- El objetivo de la creación del mencionado sistema de interconexión de datos es controlar el financiamiento y los gastos de las organizaciones políticas durante las campañas electorales.

Control descentralizado

- Se requiere que el sistema de interconexión de datos sea de control descentralizado, es decir, el control de este sistema no debe recaer sobre una única entidad central. Todas las entidades de control deben participar de acuerdo a sus roles y atribuciones dentro de la red para la aprobación de las transacciones que se realicen en el **SIDCFP**.

Requerimientos de seguridad

Los requerimientos de seguridad se organizan usando la triada CIA, para facilitar la comprensión de estos requerimientos.

Integridad

- Se requiere preservar la integridad de los datos registrados en el **SIDCFP**, es decir, dichos datos deben ser inmutables y no deben permitir manipulaciones no autorizadas, ya que la principal preocupación de las entidades de control. Para ello estas entidades de control deben autenticarse debidamente en la red.

Confidencialidad

- Se requiere que las entidades de control puedan acceder a los datos acerca del financiamiento y gastos de las organizaciones políticas registrados en el **SIDCFP**, respetando lo establecido en la norma sobre la confidencialidad, privacidad de datos personales, sigilo bancario, etc.
- Se requiere que la información del **SIDCFP**, sea de acceso público, con las limitaciones indicadas en el párrafo anterior.

Disponibilidad

- Se requiere que los datos registrados en el **SIDCFP** deben estar disponibles para su acceso ininterrumpido y en línea para las todas las entidades de control participantes. Con el acceso a los datos dichas entidades de control deben realizar sus propios procesos de auditorías e investigación durante el ciclo de vida de una campaña electoral.
- Los datos registrados deben estar disponibles para el acceso público con las limitaciones de las leyes de protección de datos privados.

Para cumplir los requerimientos de seguridad se requiere que el **SIDCFP** implemente los mecanismos necesarios para que las entidades de control se identifiquen de manera segura. El SIDCFP debe disponer de un sistema seguro de gestión de identidades, autenticación, autorización y auditoria.

2.1.2.2 Requerimientos del Middleware

El middleware debe proporcionar el fundamento para un sistema de interconexión de datos distribuido, con control descentralizado, que garantice la inmutabilidad de la información y transacciones y gestione la identidad de las entidades participantes en el **SIDCFP**.

Autenticación, autorización y auditoria

- El middleware debe facilitar la identificación y autenticación de los participantes del **SIDCFP** a través de infraestructuras de llave publica, proporcionando un sistema distribuido de control descentralizado permisionado para proteger la confidencialidad de los datos registrados.
- El **SIDCFP** debe definir y controlar los roles y autorizaciones de todos los actores del sistema.
- El **SIDCFP** debe disponer de mecanismos automáticos y seguros para la realización de auditorías.

Comunicaciones

- El middleware requiere utilizar comunicación segura para la implementación de todos los servicios (cadena de bloques, algoritmos de consenso, transacciones, autenticación, etc.).
- Para facilitar la implementación de los contratos inteligentes y aplicaciones, el middleware debe proporcionar facilidades de comunicación de alto nivel, que transparente los diferentes sistemas informáticos utilizados por las organizaciones participantes en el SIDCFP.

Inmutabilidad de información y transacciones por transparencia

- El middleware requiere usar tecnologías de contabilidad distribuidas, que utilicen cadenas de bloques para almacenar y proteger la integridad los registros contables del financiamiento de las organizaciones políticas.
- El middleware requiere implementar contratos inteligentes que implementen las políticas y las aplicaciones que gestionen el funcionamiento de las transacciones del **SIDCFP**, de modo que, automaticen el proceso de recolección de datos acerca del financiamiento y gastos de las organizaciones políticas.

Consenso

- Al ser el middleware de control descentralizado, se debe utilizar un algoritmo de consenso descentralizado, como Kafka, el cual se debe encargar de orquestar el orden, la aprobación y la verificación de las transacciones presentadas en el **SIDCFP.**

2.1.2.3 Requerimientos de la Aplicación

La aplicación del **SIDCFP** debe garantizar la integridad, confidencialidad y disponibilidad de los datos registrados, además, transparentar el financiamiento de las campañas electorales. Para cumplir con los requerimientos obtenidos de las leyes y normas, así como los objetivos buscados en el desarrollo de este **TIC**, se sugiere no implementar una plataforma web para que las propias organizaciones políticas registren la información tanto de los aportes recibidos como de los gastos realizados, tal como lo sugieren los artículos. 211.1 y Disposición transitoria Segunda del Código de la Democracia.

Para automatizar la recopilación de los datos acerca del financiamiento, se requiere que todas las transacciones relacionadas al control del gasto electoral sean bancarizadas. Por lo tanto, las instituciones bancarias deben participar directamente en el **SIDCFP**, a través de las cuales, se puede obtener una constancia de las transacciones realizadas en las respectivas cuentas corrientes de las organizaciones políticas. Por otra parte, para automatizar la recolección de datos acerca de los gastos realizados por las organizaciones políticas, se sugiere asignarle la responsabilidad al SRI, ya que esta institución está encargada de autorizar y emitir comprobantes de ventas y facturas. De modo que, registre los datos de los comprobantes emitidos a los números de RUC de las organizaciones políticas, y los datos de los proveedores que emitieron dicho comprobante o factura.

Estas automatizaciones reducen el tiempo de espera desde que se realizan tanto los aportes como los pagos hasta que estos sean registrados por las organizaciones políticas en el **SIDCFP**.

A partir del análisis realizado en la sección **Análisis de requerimientos legales y normativos**, y de las sugerencias presentadas en esta sección, se plantearon las principales transacciones que debe desarrollar la aplicación. Dichas transacciones se describen a continuación:

Registro de organizaciones políticas

El CNE es la entidad encargada de recopilar y organizar los datos de las organizaciones políticas. Los datos que se requieren de las organizaciones políticas son: el número de RUC de la organización política, el nombre o razón social asociada al RUC de la organización, el nombre de la institución bancaria y el número de cuenta corriente que abierta por la organización política para recibir aportes y realizar pagos de sus gastos.

Además, el CNE debe asignar un valor al límite máximo de gastos que puede realizar una organización política.

Con estos datos el CNE debe elaborar el conjunto inicial de datos que se almacenan en las bases de datos del sistema.

Registro de aportes

Se requiere almacenar un registro de los aportantes para conocer el origen del financiamiento obtenido por las organizaciones políticas. Dichos aportantes se registran en el momento que realicen un depósito o transferencia hacia la cuenta de una organización política a través de una institución bancaria. La institución bancaria debe solicitar los datos necesarios para identificar de manera precisa al aportante. Los datos con los que debe contar el registro del Aportante son: nombres y apellidos completos, número de cédula de ciudadanía, dirección, teléfono, correo electrónico. En caso de que el aportante sea una empresa u organización se debe considerar Razón Social en lugar de nombres y apellidos, así como número RUC en lugar de Cédula de ciudadanía.

Los registros de los aportantes sirven a las entidades de control para realizar procesos de fiscalización e investigación.

Registro de gastos

Se requiere almacenar un registro de los proveedores para conocer el destino de los montos utilizados por las organizaciones políticas. Dichos proveedores se registran en el momento que se genere una factura o comprobante de ventas a través del sistema de facturación SRI con el número de RUC de la organización política. Junto con dicho comprobante se debe generar una transacción bancaria desde la cuenta de la organización política hacia la cuenta del proveedor de servicio. La factura debe contener los datos identificativos del proveedor y debe detallar los datos de la organización política mientras que de la transacción bancaria se debe conservar el monto y el número de cuenta del proveedor.

Auditoria automatizada e independiente

Se requiere que las diferentes entidades de control puedan consultar los cambios por los que han atravesado los registros para realizar sus procesos de auditoría e investigación. Las entidades de control deberán acceder a dichos registros contables desde sus propios sistemas sin restricciones de privacidad y sin la necesidad de solicitarlos a otra entidad. Es decir, el sistema debe facilitar a las entidades de control la auditoria al proporcionar un registro inmutable de historial de cambios por los que atraviesan los registros con lo que se proporciona la transparencia que requieren las mencionadas entidades de control.

2.2 Diseño

El diseño del **SIDCFP** se encuentra estructurado del mismo modo que el análisis de requerimientos tecnológicos; es decir, se lo realizará en las mismas tres macro capas: infraestructura física, middleware y aplicación.

2.2.1 Diseño de la infraestructura física

En esta fase de diseño se discute la infraestructura tecnológica de procesamiento, almacenamiento y comunicación que requiere implementar cada entidad de control que participe en el **SIDCFP**.

2.2.1.1 Procesamiento

Cada entidad de control requiere implementar varios servidores (lógicos, no físicos) para implementar el middleware y la aplicación. Se requiere de un servidor de transacciones para implementar los nodos que conforman cada organización. Hyperledger Fabric usa un mecanismo de consenso de aprobación de transacciones mediante votación por parte de la mayoría de los participantes, por esta razón los servidores de transacciones no requieren de alta capacidad de procesamiento, pero si requieren de alta disponibilidad.

También debe implementarse un servidor para poner en marcha el servicio de ordenamiento. Ya que, este es un nodo fundamental para el funcionamiento de la red de Hyperledger Fabric debe implementarse de manera redundante. Un balanceador de carga para este servicio evitará caídas de dicho servicio, ocasionados por fallas de hardware y/o software.

Finalmente, todas las entidades de control requieren implementar un nodo para la autoridad de certificados, CA; para alojar los correspondientes certificados digitales, ya que Hyperledger Fabric requiere que cada participante de la red se encuentre debidamente identificado. Además, este nodo CA también debe encargarse de habilitar las comunicaciones seguras mediante TLS.

Todos los servidores participantes de la red, ya que se requiere comunicaciones protegidas mediante criptografía, deben estar basados en la arquitectura x86-64: la cual implementa *AES-NI (Advanced Encryption Standard New Instrucciones*): aceleración de cifrado por hardware. Este tipo de arquitectura facilita el uso de instrucciones criptográficas al ser ejecutadas por el hardware, de este modo, se obtiene un aumento en el rendimiento en comparación con el uso instrucciones criptográficas a nivel de software [40].

2.2.1.2 Almacenamiento

Para el almacenamiento se debe tomar en cuenta que los nodos Peer requieren almacenar dos bases de datos, estás son: cadena de bloques y base de datos del estado actual. La base de datos del estado actual posee una cantidad inicial de registros proporcional a la cantidad de organizaciones políticas en el país, ya que el control se debe realizar a nivel de candidatos y no solo a nivel de organización política. El crecimiento de la base de datos del estado actual se presenta con el registro de las transacciones de aportes y pagos de las organizaciones políticas. De modo similar, la cadena de bloques crece a lo largo del ciclo de vida de la campaña electoral y durante este periodo requiere de alta disponibilidad. Por tanto, el sistema de almacenamiento debe ser elástico y estar basado en configuraciones *RAID*, de modo que permita el crecimiento de la base de datos de estado actual y de la cadena de bloques a lo largo del tiempo y además proporcione alta disponibilidad.

2.2.1.3 Comunicación

Debido a que uno de los requerimientos es la creación de un sistema de interconexión para el control financiero a la política, es necesario discutir acerca del uso de una red de comunicaciones que interconecte a las diferentes entidades de control. A continuación, se presentan dos posibilidades y la respectiva valoración de las ventajas y desventajas que presenta implementación.

Red Privada física vs Red Privada Virtual

Existe la posibilidad de implementar un tendido de red físico, de uso exclusivo para el mencionado sistema de interconexión para las respectivas entidades de control del país. Dicha implementación para crear la mencionada red de interconexión trae como ventaja el aislamiento de la red, lo que proporciona seguridad en el canal de comunicaciones al usarse de modo exclusivo para interconectar a las entidades de control. Además, al usarse de manera exclusiva, todo el ancho de banda del canal se encontraría disponible para las transacciones del mencionado sistema. Esta posibilidad supone el costo de los materiales usados para construir la red, es decir, cableado estructurado, enrutadores, conmutadores, repetidores, etc. Además del costo de instalación y mantenimiento de la red física. También se debe tomar en cuenta los costos y el tiempo que tarda el diseño e implementación y pruebas para poner en marcha una red física.

Otra posibilidad es implementar una red privada virtual para comunicar a las entidades de control participantes en el **SIDCFP**. Mediante esta implementación se aprovechan las conexiones a Internet que actualmente poseen dichas entidades de control, con lo que se

evitan los costos de la implementación de un tendido físico. Esta implementación supone el costo de la adquisición del servicio asumiendo que todas las entidades de control usan el proveedor de servicios de Internet del estado CNT. Ya que, el tráfico de esta red privada virtual viajara a través de la infraestructura de red de cada proveedor de internet, es vital establecer conexiones seguras mediante protocolos de seguridad que aseguren la integridad y confidencialidad del tráfico de datos.

A continuación, en la **Figura *4*** se observa a las entidades de control acceder a internet a través de un Proveedor de Servicios de Internet (ISP). Sobre estas conexiones, en líneas entrecortadas, se encuentra una red privada virtual que interconecta a todas las entidades de control.

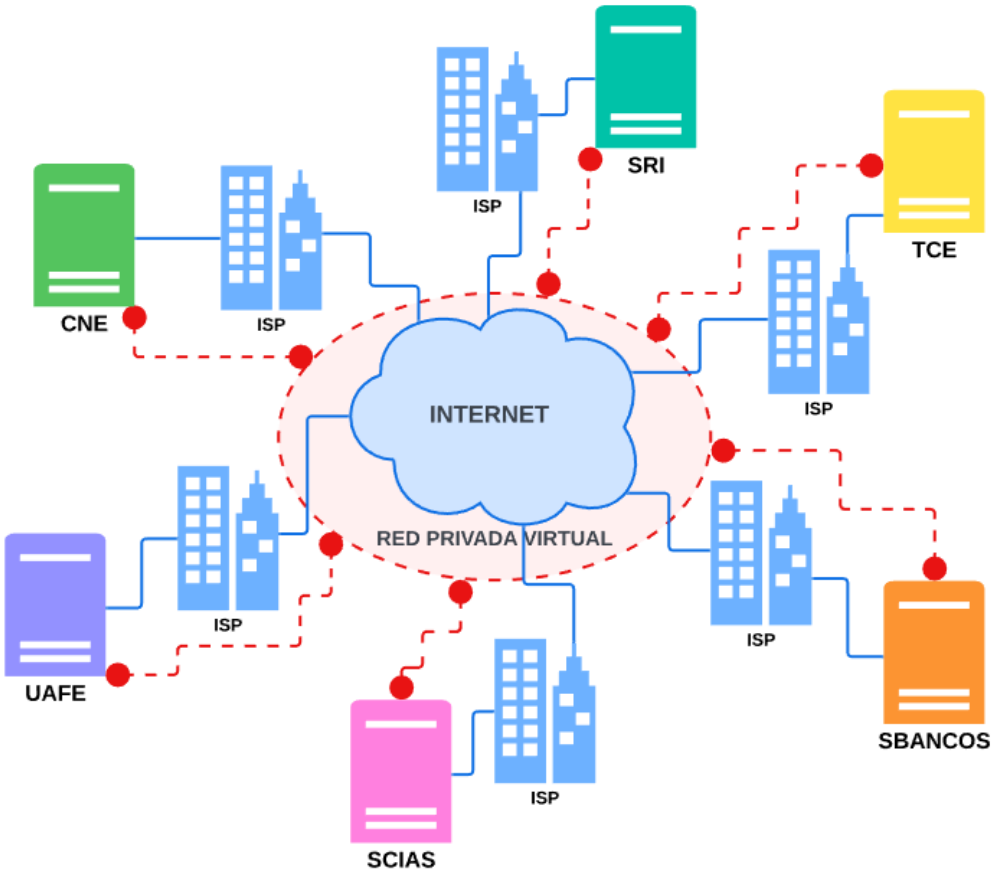

Figura 4: Diseño de Red Privada Virtual

Estructura de direcciones IP (privadas) y puertos TCP utilizados

Para la comunicación en esta red privada virtual deben definirse direcciones IP privadas para cada entidad de control del SIDCFP. Para estas direcciones IP privadas se puede utilizar una red privada de clase B; por ejemplo, 172.16.0.0/16. De este modo, a cada

institución se le asignaría una subred de dicha red de clase B, como se puede observar en la **Tabla 1**.

Tabla 1: Asignación de subredes a las entidades de control

	Subred	**Institución**
1	172.16.1.0/24	Consejo Nacional Electoral
2	172.16.2.0/24	Superintendencia de Compañías
3	172.16.3.0/24	Superintendencia de Bancos y Seguros
4	172.16.4.0/24	Comisión de Control Cívico de la Corrupción
5	172.16.5.0/24	Unidad de Análisis Financiero y Económico
6	172.16.6.0/24	Consejo de la Judicatura
7	172.16.7.0/24	Tribunal Contencioso Electoral
8	172.16.8.0/24	Servicio de Rentas Internas
9	172.16.9.0/24	Contraloría General del Estado
10	172.16.10.0/24	Procuraduría General del Estado
11	172.16.11.0/24	Fiscalía General del Estado

La subred 172.16.100.0/24 debe ser utilizada para las redes punto a punto entre las diferentes entidades de control.

A continuación, en la **Tabla 2** se presenta una distribución de direcciones IP dentro de la subred para los elementos de la red. En estas direcciones IP las entidades de control deben alojar los servidores necesarios que requiere el Middleware para su funcionamiento.

Tabla 2: IP para los elementos de la red

Servicio	**Dirección IP**
Router	.1
Orderer	.5
Peer	.10
CA	.15

Los puertos que utiliza Hyperledger Fabric son 7051 para el nodo Peer, 4369 para la base de datos, 7050 para el Orderer y 7054 para la CA. Estos puertos deben ser tomados en cuenta para permitir el tráfico TCP al momento de implementar los cortafuegos en la red.

2.2.2 Diseño del Middleware

En esta sección se presentan las configuraciones que deben considerarse para implementar una red de Hyperledger Fabric.

2.2.2.1 Diseño de la PKI

Para establecer confianza entre las PKI y sus CAs, se debe utilizar la cadena de confianza x509. Cada organización define su PKI mediante su propia CA. Para eso se sugiere adquirir un certificado digital exclusivo para este **SIDCFP,** debe ser provisto por una Autoridad Certificadora de confianza reconocida en el país, por ejemplo, el Banco Central del Ecuador, es una Autoridad certificadora que proporciona certificados digitales para los Entidades de Control del Gobierno Central.

El certificado digital adquirido por cada Entidad de Control, llave pública y privada, debe utilizarse para crear sus propios certificados raíz. Mediante los certificados raíz creados por cada Entidad de Control deben firmarse los certificados digitales de para cada nodo que implemente dichas entidades de control como organización participante del **SIDCFP.**

2.2.2.2 Configuración de Canales

Un canal es suficiente para comunicar a los nodos Peer que poseen las entidades de control. Con este canal se asegura la privacidad y aislamiento de las transacciones considerando que todos los Peers que interactúan en el canal están autorizados para el manejo de los datos que componen los Assets. En el caso de requerir un mayor aislamiento de datos se puede habilitar el uso de recolección de datos privados, mediante la cual, se puede ocultar los datos de los Assets a los participantes del canal.

2.2.2.3 Configuración de organizaciones (entidades de control)

Cada entidad de control debe configurar los elementos que conforman su organización, estos son Peer, Orderer, CA, bases de datos.

Peer, cada entidad de control que intervenga en la aprobación de las transacciones cada entidad deberá implementar un nodo de tipo Peer para gestionar las transacciones mediante el Chaincode y el Ledger.

Orderer, Inicialmente es suficiente la implementación de un nodo de tipo Ordenador para el Canal de la red y la entidad que se puede encargar de implementar este servicio es el CNE, de modo que, será encargado de distribuir y ordenar las transacciones ocurrentes en

el canal. Además, debe implementar las políticas de ordenamiento y algoritmos de consenso que utilizarán los participantes para aprobar las transacciones. Posteriormente para conseguir una mayor descentralización en el **SIDCFP**, cada entidad de control deberá implementar su propio servicio de ordenamiento con un algoritmo de consenso especializado como Raft.

CA, las entidades de control requieren implementar un nodo para la CA, el cual tendrá la tarea de compartir los certificados digitales de su respectiva organización, para implementar el correspondiente PKI. Además, dada la necesidad de proveer seguridad al canal de comunicaciones las organizaciones deben configurar sus nodos CA para que las comunicaciones entre los Peers que conforman la red las realicen mediante el protocolo TLS.

Bases de datos

Hyperledger Fabric ofrece la posibilidad de utilizar LevelDB como base de datos predeterminada, tanto para la cadena de bloques como para la base de datos del estado actual. Además, para la base de datos del estado actual, ofrece la posibilidad de utilizar CouchDB en el caso de requerir soluciones más sofisticadas. Todas las entidades deben usar el mismo tipo de base de datos para evitar que existan conflictos de compatibilidad.

2.2.2.4 Arquitectura de la Red

En la **Figura *5*** se presenta un diseño de la red en términos de Hyperledger Fabric. Esta red consta de un consorcio de todas las entidades de control del gobierno central, tales como, CNE, Superintendencia de Compañías, UAFE, SRI, etc. Dicho consorcio de organizaciones se encuentra interconectado mediante el canal C1. El canal se encuentra gestionado por el nodo ordenador CNE-Orderer que debe ser implementado por el CNE ya que es la entidad que debe ponerse al frente con la implementación del **SIDCFP**. Cada entidad de control implementa un nodo Peer con el que participa en el canal y cada entidad debe poner en marcha una aplicación para poder gestionar mediante su respectivo Peer el Chaincode y el Ledger.

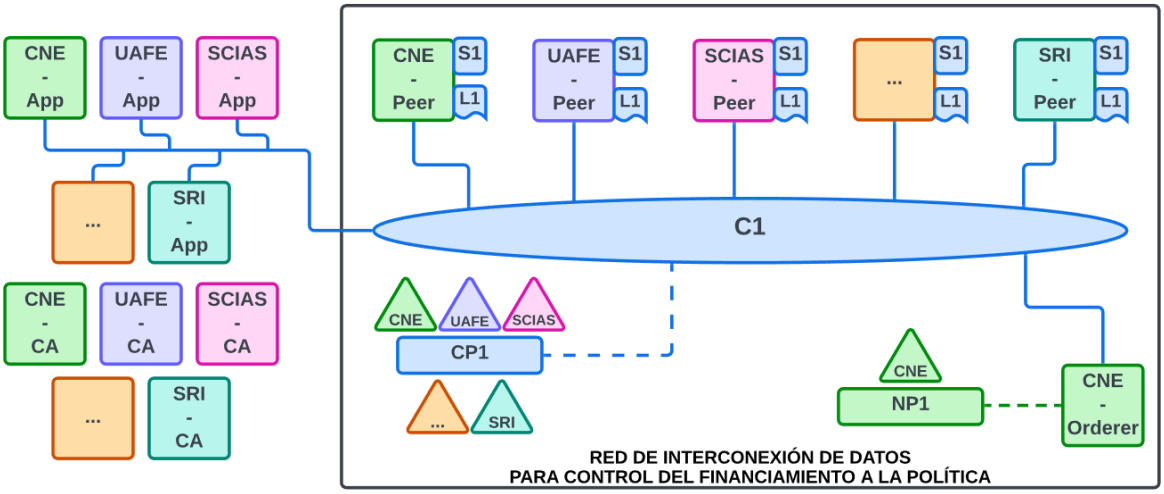

Figura 5: Diseño de Arquitectura del SIDCFP en Hyperledger Fabric

Las aplicaciones de las entidades de control gestionan sus respectivos Peer participantes en la figura 5 se presentan los nodos Peer etiquetados con el nombre de la entidad de control un guion medio y el identificador adicionalmente la palabra Peer para identificarlos. Cada gestiona su copia de Ledger L1 y el Chaincode definido para el canal S1. En la parte externa de la red junto a las aplicaciones se encuentran los respectivos nodos de CA de cada entidad de control participante la red.

2.2.2.5 Flujo de las transacciones

Las transacciones realizadas por los participantes de la red siguen un flujo desde su ejecución hasta su aprobación y almacenamiento en los Ledger del sistema.

En la **Figura 6** se puede observar el flujo por el que atraviesan las transacciones de aporte y pago.

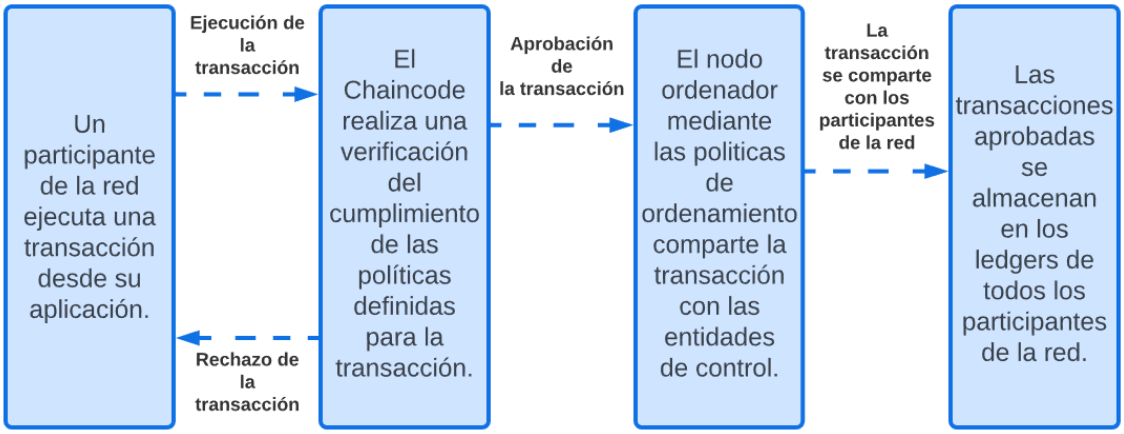

Figura 6: Flujo de aprobación de transacciones

El flujo de transacciones es el mismo para los registros de actores (organización política, aportante, proveedor) y para las transacciones (aporte y pago). Inician con la ejecución de una transacción por parte un participante, seguido por la comprobación del cumplimento de las políticas del Chaincode para luego ser compartidas a la red por el nodo ordenador. Finalmente se almacena el respectivo registro de dicha transacción en todos los Ledger participantes.

2.2.3 Diseño de la aplicación

Esta sección presenta una solución a los requerimientos definidos en la sección Requerimientos de la Aplicación. Se definen las funciones que debe implementar el **SIDCFP** mediante el Chaincode y la interfaz de usuario.

2.2.3.1 Chaincode

El Chaincode está compuesto de las siguientes secciones: Estructura de datos, Init, Invoke. El diseño de dichas secciones se presenta a continuación.

Estructura de datos

Mediante el análisis de los artículos realizado en la sección 2.1.1, se encontraron cinco estructuras de datos que se deben desarrollar en el **SIDCFP**. Estos se dividen en actores y transacciones. Los actores Organización Política, Aportante, Proveedor y sus respectivos campos se presentan en la **Figura 7**.

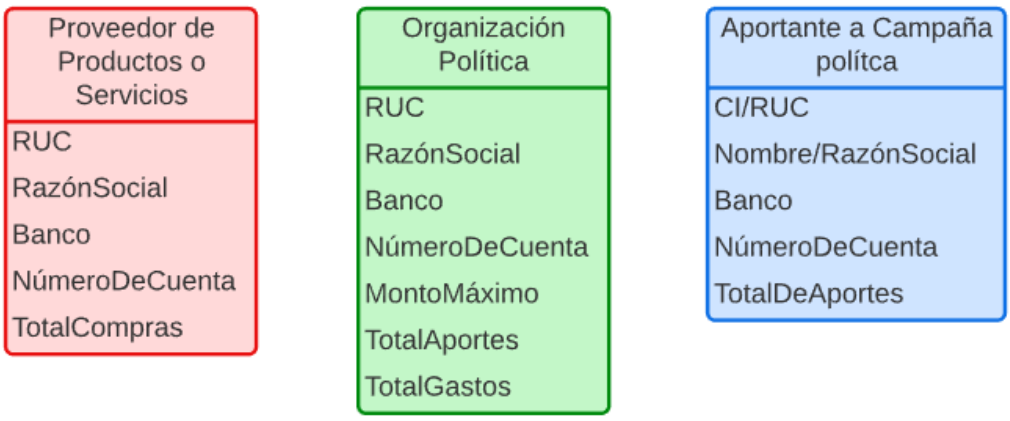

Figura 7: Estructura de datos, Assets de Actores

Cada estructura de datos consta de los campos que fueron definidos para realizar las transacciones durante la fase de analisis de requrimientos de la aplicación. Estos campos

se utilizan para crear los registros de los activos lo cual se describirá en mayor detalle en la sección 2.2.3.2.

Además, se definieron dos tipos de transacciones que se pueden realizar mediante el uso del Chaincode y la aplicación en esta red. Estas son pagos y aportes, sus campos se muestran en la **Figura *8***.

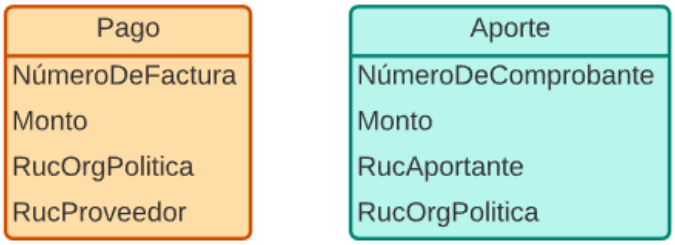

Figura 8: Estructura de datos de transacciones de aporte y pago

Las estructuras de datos de las transacciones de pago y aportes se componen de los campos indispensables para preservar un registro adicional como constancia de la transaccion realizada. Dichos registros poseen un identificador del tipo de transaccion realizada, el monto de la transaccion realizada, el identificador del Actor que representa el origen del monto de la transaccion, y el identificador del Actor que representa el destino del monto de la transaccion.

A partir de estas estructuras de datos se definió el Activo (Asset) que contiene el valor que se desea proteger. En el caso de este **SIDCFP**, el valor del Activo son los montos obtenidos mediante aportes por las organizaciones politicas y los gastos realizados por compras de productos o servicios por dichas organizaciones políticas.

2.2.3.2 Algoritmos

En esta sección se presenta una formalización de los registros que implementa el **SIDCFP**.

Init

- **Registro de activos (Assets) Organizaciones Políticas**

Esta transacción crea el conjunto inicial de registros de todas las organizaciones políticas. Cada registro consta de los siguientes datos: RUC de la Organización Política, Razón Social de la Organización Política, institución bancaria, número de cuenta bancaria, monto máximo que puede recibir, total de aportes recibidos y total de gastos realizados. El campo de tipo identificador de este registro es el RUC de la organización política.

El CNE es el encargado de recopilar los datos de las diferentes organizaciones políticas y de ejecutar esta transacción para presentar los registros a las entidades de control participantes del **SIDCFP** y solicitar la aprobación dichos registros para su respectiva inicialización en los Ledger de las mencionadas entidades de control del gobierno central.

Invoke

- **Registro de Assets de Aportante**

Esta transacción crea el registro de un aportante con los siguientes datos: Número de cédula o RUC del aportante, Nombre o Razón Social del aportante, institución bancaria, número de cuenta bancaria y total de aportes realizados. El campo de tipo identificador de este registro es el Número de cédula o RUC del aportante. El contrato inteligente verifica que el Número de cédula o RUC del aportante no se encuentre registrado. En caso de no estar registrado, se crea el registro para completar la transacción de manera exitosa. En caso contrario presenta un mensaje de error “el Número de cédula o RUC del aportante ya se encuentra registrado” y la transacción finaliza sin realizar cambios en los registros.

Las instituciones bancarias deben encargarse de realizar este registro en el momento que se realice un aporte hacia la cuenta de una organización política. La transacción se completa al realizar también el registro del aporte por parte de dicha institución bancaria.

- **Registro de Aportes**

Esta transacción crea el registro de un aporte con los siguientes datos: Número de comprobante de depósito o transferencia del aporte, RUC o CI del aportante, RUC de la organización política que recibe el aporte, el monto recibido por la organización política. El campo de tipo identificador este registro es el número de comprobante de depósito o transferencia del aporte. El contrato inteligente verifica que la cantidad máxima que puede recibir una organización política durante la campaña electoral no sea superada con la suma entre el total de aportes y el aporte que se desea registrar. En caso de no superar la cantidad máxima, se crea el registro y se adiciona el aporte al total de aportes recibidos por la organización política para completar la transacción de manera exitosa. En caso contrario la transacción debe finalizar sin realizar cambios en los registros.

- **Registro de Assets de Proveedores**

Esta transacción crea el registro de un proveedor con los siguientes datos: RUC del proveedor, Razón Social del proveedor, institución bancaria, número de cuenta bancaria y total de pagos recibidos. El campo de tipo identificador de este registro es el RUC del

proveedor. El contrato inteligente verifica que el RUC del proveedor no se encuentre registrado. En caso de no estar registrado, se crea el registro para completar la transacción de manera exitosa. En caso contrario se presenta un mensaje de error “el RUC de proveedor ya se encuentra registrado” y la transacción finaliza sin realizar cambios en los registros.

El SRI debe encargarse de realizar este registro en el momento que se genere una factura o comprobante electrónico por la adquisición de productos o servicios por parte de una organización política. La transacción se completa al realizar también el registro del pago por parte de dicha institución bancaria.

- **Registro de Pagos**

Esta transacción crea el registro de un pago con los siguientes datos: Número de factura emitida por el proveedor, el RUC de la organización política que adquiere dichos productos o servicios, el RUC del proveedor y el monto pagado al proveedor. El campo de tipo identificador este registro es el número de factura. El contrato inteligente verifica que el número de factura no se encuentre registrado. a continuación, debe verificar la existencia de fondos para cubrir el pago de la factura. En el caso de no encontrarse registrado el número de factura y existir los fondos necesarios, se crea el registro del pago y se adiciona el monto al total de gastos realizados por la organización política para completar la transacción de manera exitosa. En caso contrario la transacción finaliza sin registrar el pago y sin realizar cambios en los registros.

- **Consultar Registro**

Esta consulta presenta como resultado el estado actual de un registro, ya sea este de organizaciones políticas, aportantes o proveedores, así como de transacciones como aporte o pago, usando como parámetro de búsqueda el campo de tipo identificador del respectivo registro. En el caso de que el registro buscado no exista la consulta presenta un mensaje de tipo “Registro no encontrado”.

- **Consultar todos los registros**

Esta función muestra todos los Asset registrados en el **SIDCFP**. Como resultado de esta consulta se presentan todos los Assets existentes en el estado actual del Ledger. Esta consulta debe estar disponible para su ejecución para todas las entidades de control a través de sus respectivas aplicaciones. Un subconjunto de estos Assets debe ser publicados a través de las páginas web del CNE para su acceso público con las respectivas limitaciones de las leyes de protección de datos existentes en nuestro país.

- **Eliminación de Registros**

Esta transacción elimina un registro del estado actual usando como parámetro de búsqueda el campo de tipo identificador del respectivo registro para ello el contrato inteligente debe verificar la existencia de dicho registro. En el caso de que el registro buscado no exista la consulta presenta un mensaje de tipo “Registro no encontrado” y la transacción finaliza sin realizar cambios en los registros.

- **Consulta de Historial de un Registro**

Esta consulta presenta como resultado el historial de un registro utilizando como parámetro de búsqueda el campo de tipo identificador del respectivo registro. El contrato inteligente debe verificar la existencia de dicho registro. En el caso de que el registro buscado no exista la consulta presenta un mensaje de tipo “Registro no encontrado”.

Esta consulta debe estar disponible para las entidades de control mediante la cual pueden realizar sus respectivos procesos de auditorías.

2.2.3.3 Interfaz de Usuario.

Para facilitar el uso de las funciones definidas es preciso que cada entidad de control implemente su propia interfaz para interactuar con el sistema. Mediante dicha interfaz de usuario las mencionadas entidades de control pueden acceder a las funciones que les conciernen. En el prototipo, presentamos un ejemplo muy simple de dicha interfaz.

2.3 Prototipo

En esta sección se presenta la implementación del prototipo del **SIDCFP**. Para realizar su implementación se utilizó una máquina virtual y contenedores para virtualizar los servicios de la red distribuida necesarios para ejecutar la red de prueba de Hyperledger Fabric dentro de un mismo computador físico. La implementación de este prototipo sigue la misma estructura utilizada durante las fases de análisis y diseño.

2.3.1 Infraestructura física

Fabric-Samples posee una red de prueba de Hyperledger Fabric, diseñada para ejecutar aplicaciones para probar el funcionamiento de los contratos virtuales. En esta sección se describe la infraestructura utilizada y las configuraciones realizadas para implementar la red de prueba mencionada anteriormente.

2.3.1.1 Procesamiento

Debido a que, Fabric Samples ofrece la posibilidad de ejecutar sus servicios virtualizados mediante contenedores de Docker, se puede implementar en un único servidor todos los servicios requeridos de Hyperledger Fabric como lo son Peers, Orderer y CAs.

Para hacer uso de Fabric Samples se instaló, en una máquina virtual de Oracle Virtual Box, el sistema operativo Ubuntu Server 22.04 para una arquitectura x86-64, con 1024 MB de memoria RAM, un procesador asignado hasta un 100% y un adaptador de red de tipo puente.

Fabric Samples requiere de una serie de paquetes que deben ser instalados como prerrequisitos para su correcto funcionamiento. La instalación paso a paso de dichos prerrequisitos y de Fabric Samples se encuentra en el ANEXO I. Instalación de prerrequisitos.

2.3.1.2 Almacenamiento

Para la implementación del prototipo se consideraron los siguientes requerimientos de almacenamiento:

- Sistema operativo
- Gestor de contenedores
- Prerrequisitos
- Red de prueba
- Datos de prueba

Se consideró que 20GB es una cantidad de almacenamiento suficiente para poner en marcha todos los requisitos considerados.

2.3.1.3 Comunicación

La red de este prototipo es una red TCP/IP, basada en la interfaz localhost. Los nodos del **SIDCFP** utilizan los puertos TCP descritos en la sección 2.2.1.3. Los Peer, Orderer, CAs de las organizaciones se encuentran implementados en contenedores de Docker que simulan los nodos de las 3 organizaciones que participan en este prototipo y se comunican a través del adaptador de puente de virtual box.

2.3.2 Middleware

En la red de prueba, Fabric-Samples, existen las facilidades para poner en marcha una red que consta de tres organizaciones. Dichas organizaciones forman un consorcio por lo que cada una implementa su respectivo Peer (P1, P2, P3) y CA (CNE-CA, UAFE-CA, SCIAS). Así mismo se implementa un Orderer. La arquitectura de dicha red se presenta a continuación en la **Figura 9**.

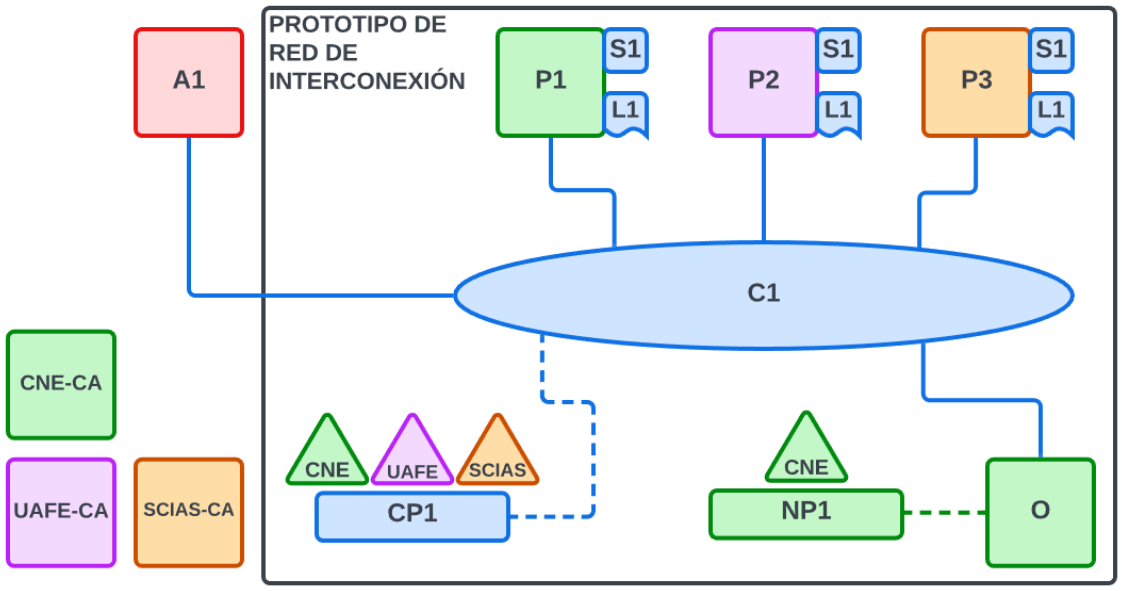

Figura 9: Arquitectura del Prototipo en Fabric-Samples

La red presentada en el prototipo es la misma red del diseño, de un modo simplificado para conservar la funcionalidad y la arquitectura presentada en la sección 2.2.2.

2.3.2.1 Contenedores

A partir de las configuraciones mostradas en la sección 2.2.2. (***Diseño del Middleware***) se crearon los contenedores de Docker para implementar los Peer, CA y Orderer que se comunicarán en el canal a través de la aplicación del SIDCFP.

En la **Figura *10*** se presenta un resumen del resultado producido por el comando "*docker ps –a"*. En esta figura se puede observar los puertos TCP que utiliza cada contenedor de los nodos descritos en la sección 2.3.2, los nombres de dichos contenedores y las imágenes de que se utilizaros para crear dichos contenedores. Dicho comando fue ejecutado, cuando todos los nodos y servicios del SIDCFP estaban operacionales.

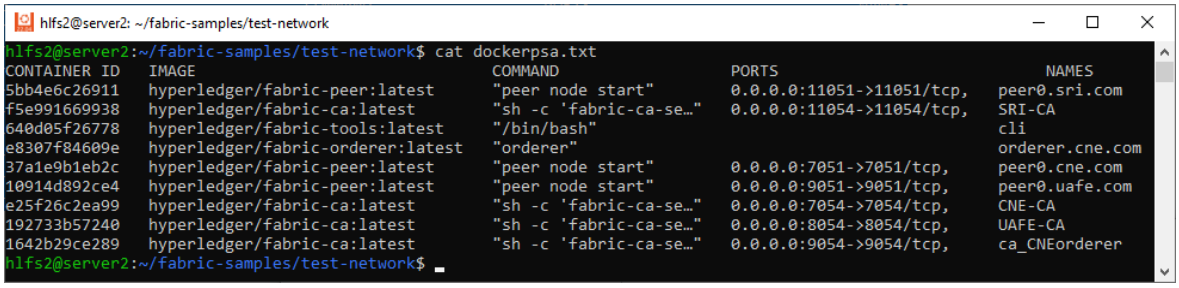

Figura 10: Contenedores de nodos de la red

2.3.2.2 Seguridad de la red

A pesar de que se trata de una red de localhost, la seguridad de las comunicaciones entre los componentes de la red se encuentra protegida mediante TLS para demostrar el uso de la misma en este ambiente de prueba. Para habilitar las comunicaciones mediante TLS en el momento de inicializar la red de debe especificar el uso de Autoridades de certificación. Para ello se utiliza el comando:

- `./network.sh up createChannel -ca`

En la Figura 11 se observa la creación e inicialización de los contenedores de las CA de las organizaciones.

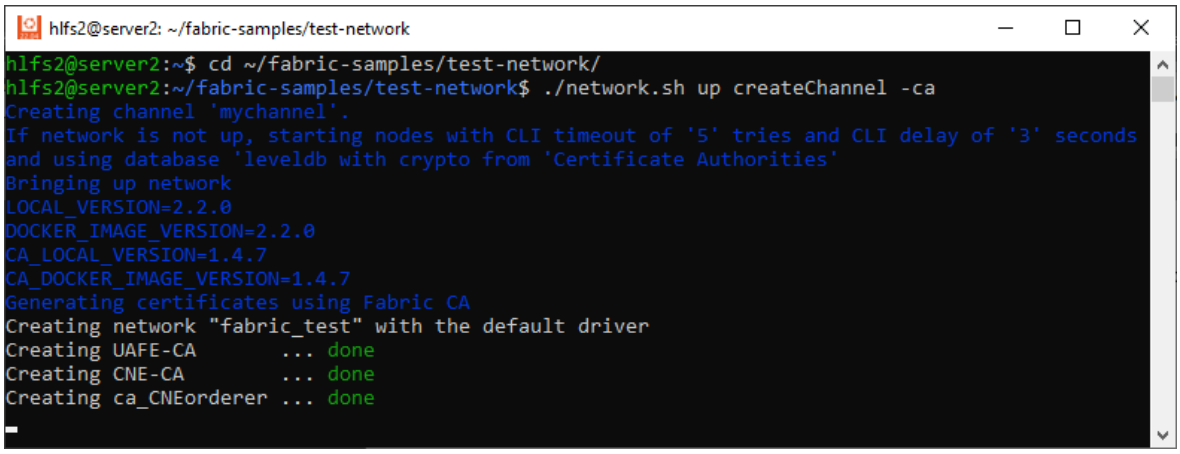

Figura 11: Inicialización de la red y creación de CAs

2.3.3 Aplicación

En esta sección de la implementación del prototipo, se definió la estructura de datos del activo, a partir de la cual, fueron definidas las estructuras de datos dentro del chaincode y sus respectivas funciones. Por otra parte, se definieron también la interfaz de usuario y las funciones que presenta el menú de dicha interfaz.

2.3.3.1 Estructura de datos

La estructura de datos se redujo a un único Activo, esto se realizó debido a que los actores poseen muchos campos en común y a que solo puede existir un Activo por contrato inteligente. Para distinguir estos activos similares se introduce el campo DocType, el cual, contiene una cadena de texto que identifica el tipo de Activo almacenado en el prototipo. En la **Tabla 3** se muestra la interpretación de los campos que aplica para cada tipo de actor, es decir, Organización Política, Aportante, Proveedor y para las transacciones Aporte y Pago.

Tabla 3: Simplificación de Assets de Actores y transacciones

Campos del Asset	**Actores**			**Transacciones**	
	Organización política	**Aportante**	**Proveedor**	**Aporte**	**Pago**
DocType	ORG-POL	APORTANTE	PROVEEDOR	APORTE	PAGO
RUC	RUC	RUC/CI	RUC	Identificador del aporte	Identificador del pago
Razón Social	Razón Social	Razón Social / Nombre	Razón Social	RUC de la organización política	RUC de la organización política
Banco	Identificador de institución Bancaria	Identificador de institución Bancaria	Identificador de institución Bancaria	0	0
Número de cuenta	Número de cuenta	Número de cuenta	Número de cuenta	RUC/CI del aportante	RUC del proveedor
Monto Máximo	Cantidad máxima que puede recibir durante la campaña electoral	0	0	0	0
Total de Aportes	Monto total de Aportes recibidos	Cantidad total de aportes realizados	0	Monto recibido por la organización política	0
Total de Gastos	Monto total de gastos realizados	0	Monto total recibido como pago por productos o servicios	0	Monto pagado

Los campos representados por el valor “0” no poseen un significado para los respectivos documentos y serán representados por dicho valor en el prototipo del sistema.

Los tipos de datos que se utilizaron para la creación del Asset se describen en la **Tabla 4**:

Tabla 4: Campos y tipos de datos que conforman el Asset.

Campo	Tipo de dato
DocType	string
RUC	string
Razón social	string
Banco	int
Número de cuenta	string
Monto máximo	float64
TotalAportes	float64
TotalGastos	float64

2.3.3.2 Chaincode

El Chaincode se adaptó usando las funciones ya implementadas en el archivo fabric-samples/asset-transfer-basic/chaincode-go/chaincode/smartcontract.go.

El código completo del Chaincode el lector lo puede encontrar en el ANEXO II. Código fuente Chaincode. En este archivo se define la estructura de datos del activo y las transacciones inmutables del SID. A continuación, definimos la estructura de datos de los activos y las funciones que implementas dichas transacciones.

- **Asset**

La implementación del Activo dentro del Chaincode consta de la definición de los campos, tipo de dato y su respectiva definición en formato json. La sección de código de la implementación de la estructura de datos del Activo se encuentra en el ANEXO II. Código fuente Chaincode, desde la línea 19 hasta la línea 28. Con excepción de los campos DocType que es un distintivo para el tipo de Activo y RUC que es el identificador del Activo, todos los campos deben definirse en orden alfabético para evitar que durante la serialización y des serialización varie el orden de los campos creando conflictos en los registros de los Activos.

- **Historial**

Para recuperar el historial de transacciones realizadas sobre un Activo se definió una estructura para organizar estos datos y facilitar su presentación al usuario mediante la transacción ***HistorialdeActivo*** que se definirá en la sección Invoke de transacciones. La sección de código de la implementación de la estructura de datos para la consulta del

historial se encuentra en el ANEXO II. Código fuente Chaincode, desde la línea 31 hasta la línea 36.

Transacciones

Las transacciones definidas en la sección Algoritmos, de Diseño de la Aplicación se implementan en esta sección usando la misma estructura Init e Invoke.

Init

A continuación, se implementa la función inicial que se ejecuta al iniciar la aplicación del **SIDCFP**.

- **Función InicializarLibro**

La implementación de la función *InicializarLibro* consta de dos secciones. En la primera sección, se define el conjunto inicial de datos a partir de la estructura del Activo definida. En este conjunto de datos iniciales consta de seis datos de prueba definidos mediante una lista en formato json, con los campos de tipo clave – valor que corresponden al estado inicial de cada registro. En la segunda sección, se crean los Activos de la lista en el estado actual del Libro mayor.

La sección de código de la implementación de la función *InicializarLibro* se encuentra en el ANEXO II. Código fuente Chaincode, desde la línea 41 hasta la línea 64.

En el **SIDCFP**, esta transacción debe contener los Activos de todas las organizaciones políticas registradas para participar en una campaña electoral. La cantidad de Activos que deben ser registrados dependerá de la granularidad del control del financiamiento, es decir, la cantidad de Activo puede variar desde la cantidad de organizaciones políticas hasta la cantidad de candidatos existentes en todas las organizaciones políticas.

Invoke

En esta sección se implementan las transacciones y consultas disponibles para su ejecución a través de la aplicación del SIDCFP.

- **Función CrearActivo**

La función *CrearActivo* crea un registro a partir de los datos que recibe de la aplicación. Esta función consta de tres secciones. En la primera sección comprueba que el Activo no se encuentre registrado usando para ello el identificador de los datos recolectados por la aplicación. En la segunda sección se organizan los datos recibidos siguiendo la estructura de datos del Activo previamente definido. En la tercera sección se serializan estos datos

usando la función *json.Marshal* y se los agrega al estado actual del Ledger. La sección de código de la implementación de la función *CrearActivo* se encuentra en el ANEXO II. Código fuente Chaincode, desde la línea 67 hasta la línea 92. Esta función la utiliza la aplicación para registrar los Activos de Aportante, Aporte, Proveedor y Pago en el estado actual.

- **Función ConsultarActivo**

La función *ConsultarActivo* consulta un Activo a partir del identificador obtenido por la aplicación. Esta función consta de tres secciones. En la primera sección se comprueba el acceso al estado actual del Libro mayor. Seguido de una comprobación de la existencia del registro. En la segunda sección se comprueba la existencia del Activo. En la tercera sección se solicita el registro el cual debe ser des serializado, mediante la función *json.Unmarshal*, para poder ser leído por la aplicación. La sección de código de la implementación de la función *ConsultarActivo* se encuentra en el ANEXO II. Código fuente Chaincode, desde la línea 95 hasta la línea 111. La aplicación usa esta función para consultar todos los tipos de Activos registrados en el estado actual del **SIDCFP**.

- **Función EliminarActivo**

La función *EliminarActivo* elimina un registro del estado actual del Ledger usando para ello el identificador obtenido por la aplicación. Para ello comprueba la existencia del registro. Y posteriormente lo elimina del estado actual. La sección de código de la implementación de la función *EliminarActivo* se encuentra en el ANEXO II. Código fuente Chaincode, desde la línea 114 hasta la línea 124. Los Activos eliminados por la aplicación siguen registrados en los diferentes bloques de la cadena de bloques del **SIDCFP**.

Esta función se implementó de manera didáctica para observar su funcionamiento. A pesar de que sea una función muy útil con respecto a una base de datos, las respectivas entidades de control u organizaciones políticas no requieren eliminar Activos, por esta razón esta función no debe estar disponible para su uso en ninguna aplicación de este **SIDCFP**.

- **Función ActivoExiste**

La función *ActivoExiste* comprueba la existencia de un registro del estado actual del Libro mayor usando para ello el identificador obtenido por la aplicación. La sección de código de la implementación de la función *ActivoExiste* se encuentra en el ANEXO II. Código fuente Chaincode, desde la línea 127 hasta la línea 134.

- **Función TransferirActivoAporte**

La función *TransferirActivoAporte* realiza la transferencia del valor de un aporte a una Organización Política. Para ello recibe el RUC de la Organización Política y el valor del aporte. Esta transacción consta de tres secciones. En la primera sección obtiene los datos de la Organización Política que recibe el aporte. En la segunda sección se comprueba que la suma entre el total de aportes recibidos y el aporte que se desea registrar no supere al monto máximo que la organización política puede recibir. En la tercera sección se asigna el nuevo valor total de aportes recibidos con lo que se actualiza el Activos de la Organización Política. La sección de código de la implementación de la función *TransferirActivoAporte* se encuentra en el ANEXO II. Código fuente ChaincodeANEXO II. Código fuente Chaincode, desde la línea 137 hasta la línea 155.

- **Función TransferirActivoAportante**

La función *TransferirActivoAportante* realiza la transferencia del valor de un aporte a un aportante. Para ello recibe el identificador del Aportante y el valor del aporte. Esta transacción consta de tres secciones. En la primera sección se obtiene los datos del Aportante. En la segunda sección se adiciona el valor del aporte al total de aportes realizados por el Aportante. En la tercera sección se asigna el nuevo valor al total de aportes recibidos y se actualiza el Activo del Aportante. La sección de código de la implementación de la función *TransferirActivoAportante* se encuentra en el ANEXO II. Código fuente Chaincode, desde la línea 158 hasta la línea 171.

- **Función TransferirActivoPago**

La función *TransferirActivoPago* realiza la transferencia del valor de un pago de una Organización Política a un Proveedor. Para ello recibe el RUC de la Organización Política y el valor del aporte. Esta transacción consta de tres secciones. En la primera sección obtiene los datos de la Organización Política que realiza el pago. En la segunda sección se comprueba que la suma entre el total de gastos realizados y el pago que se desea registrar no supere al total de aportes recibidos de la organización política. En la tercera sección se asigna el nuevo valor al total de gastos con lo que se actualiza el Activo de la Organización Política. La sección de código de la implementación de la función *TransferirActivoPago* se encuentra en el ANEXO II. Código fuente Chaincode, desde la línea 174 hasta la línea 192.

- **Función TransferirActivoProveedor**

La función *TransferirActivoProveedor* realiza la transferencia del valor de un pago a un Proveedor. Para ello recibe el identificador del Proveedor y el valor del aporte. Esta transacción consta de tres secciones. En la primera sección se obtiene los datos del Proveedor. En la segunda sección se adiciona el valor del pago al total de pagos recibidos por el Proveedor. En la tercera sección se asigna el nuevo valor al total de pagos recibidos y se actualiza el Activo del Proveedor. La sección de código de la implementación de la función *TransferirActivoProveedor* se encuentra en el ANEXO II. Código fuente Chaincode, desde la línea 195 hasta la línea 208.

- **Función ConsultarTodoslosActivos**

La función *ConsultarTodoslosActivos* consulta todos los Activos existentes en la base de datos del estado actual del SIDCFP. Esta función está compuesta de tres secciones. En la primera sección se consulta la base de datos del estado actual. En la segunda sección se define un arreglo de Activo el cual se llena mediante un iterador que des serializa los Activos obtenidos del estado actual mediante la función *json.Unmarshal*. En la tercera sección se envía el arreglo de Activos a la aplicación. La sección de código de la implementación de la función *ConsultarTodoslosActivos* se encuentra en el ANEXO II. Código fuente Chaincode, desde la línea 211 hasta la línea 234.

- **Función HistorialdeActivo**

La función *HistorialdeActivo* consulta el identificador de un Activo en la base de datos de la cadena de bloques y devuelve el historial de cambios por los que ha atravesado un Activo. Esta función consta de varias secciones. En la primera sección se consulta la base de datos de la cadena de bloques. En la segunda sección se define un arreglo a partir de la estructura de datos de historial. El arreglo del historial se llena mediante un iterador que des serializa los Activo recibidos mediante la función *json.Unmarshal* organizando los datos recibidos de la fecha, Activo y el estado del Activo. En la tercera sección se envía el arreglo de historial a la aplicación. La sección de código de la implementación de la función *HistorialdeActivo* se encuentra en el ANEXO II. Código fuente Chaincode, desde la línea 238 hasta la línea 279.

2.3.3.3 Interfaz de usuario

Para poder verificar el funcionamiento del Chaincode, se decidió implementar una interfaz de usuario que facilite el uso de las funciones implementadas en el Chaincode. La interfaz de usuario se adaptó usando las funciones ya implementadas en el archivo fabric-

samples/asset-transfer-basic/application-go/assetTransfer.go. Esta interfaz muestra al usuario un menú de las funciones definidas durante el diseño de la aplicación.

En la **Figura 12** se presenta la estructura de la aplicación para facilitar la descripción de las funciones implementadas para la aplicación. El código completo de la implementación de dichas estructuras de datos e interfaz está en el ANEXO III. Código fuente de la Aplicación.

```
APLICACIÓN
 main()
{
InitLedger()
   menu()
   {
        RegistrarAportante();
        RegistrarProveedor();
        RegistrarAporte();
        RegistrarPago();
        ConsultarRegistro();
        EliminarRegistro();
        ConsultarHistorial();
        MostrarTodosLosRegistros();
   }
}
```

Figura 12: Estructura de funciones de la Aplicación

Las funciones de la aplicación interactúan con la base de datos del estado actual del **SIDCFP,** con excepción de la función *HistorialdeActivo* que recorre la base de datos de la cadena de bloques. A continuación, se presenta la descripción de las funciones implementadas en la aplicación.

Init

- **Función crear conjunto inicial de datos**

Al iniciar la aplicación se ejecuta la función main de la aplicación. Esta realiza una llamada a la función *InicializarLibro* implementada en el Chaincode. El resultado de esta ejecución es la creación del conjunto de organizaciones políticas definidas en el Chaincode. Esta función debe estar disponible solo para el CNE, ya que, es la entidad encargada de organizar los datos de las organizaciones políticas con esto inicializar los datos en el libro mayor de todas las entidades de control del gobierno central.

En la aplicación de este prototipo se agregó un mensaje al ejecutar la llamada a la función *InicializarLibro* y otro para confirmar la creación correcta de este conjunto inicial de datos. La sección de código de la implementación de la función *crear conjunto inicial de datos* se

encuentra en el ANEXO III. Código fuente de la Aplicación, desde la línea 61 hasta la línea 67.

Invoke

Para interactuar con el Chaincode en la aplicación se implementó ocho funciones las cuales se describen a continuación:

- **Función Registrar Aportante**

La función registrar Aportante solicita al usuario los datos para la creación de un registro de aportante. Estos datos son organizados por la aplicación con los cuales puede llamar a la función *CrearActivo* implementada en el Chaincode. El resultado de esta ejecución es la creación del Activo del Aportante con los datos solicitados. La sección de código de la implementación de la función Registrar Aportante se encuentra en el ANEXO III. Código fuente de la Aplicación, desde la línea 86 hasta la línea 101.

- **Función Registrar Proveedor**

La función Registrar Proveedor solicita al usuario los datos para la creación de un registro de Proveedor. Estos datos son organizados por la aplicación con los cuales puede llamar a la función *CrearActivo* implementada en el Chaincode. El resultado de esta ejecución es la creación del Activo del Proveedor con los datos solicitados. La sección de código de la implementación de la función Registrar Proveedor se encuentra en el ANEXO III. Código fuente de la Aplicación, desde la línea 102 hasta la línea 120.

- **Función Registrar Aporte**

La función Registrar Aporte solicita al usuario los datos para la creación de un registro de Aporte. Estos datos son organizados por la aplicación con los cuales la aplicación llama a la función *TransferirActivoAporte* implementada en el Chaincode con lo que se verifica que el valor del total de aportes sumado el aporte, no supere el límite máximo de aportes que puede recibir la organización política. Si la llamada a la función *TransferirActivoAporte* se realiza correctamente, se llama a la función *CrearActivo* para crear el Activo del aporte. La sección de código de la implementación de la función Registrar Aporte se encuentra en el ANEXO III. Código fuente de la Aplicación, desde la línea 121 hasta la línea 153.

- **Función Registrar pago**

La función Registrar Pago solicita al usuario los datos para la creación de un registro de Pago. Estos datos son organizados por la aplicación con los cuales la aplicación llama a la función *TransferirActivoPago* implementada en el Chaincode con lo que se verifica que el

valor del total de gastos sumado el pago, no supere al total de aportes ha recibido la organización política. Si la llamada a la función *TransferirActivoPago* se realiza correctamente, se llama a la función *CrearActivo* para crear el Activo del Pago. La sección de código de la implementación de la función Registrar Pago se encuentra en el ANEXO III. Código fuente de la Aplicación, desde la línea 154 hasta la línea 186.

- **Función Consultar Registro**

La función Consultar Registro solicita al usuario el identificador del registro que se desea consultar. Con dicho identificador se realiza una llamada a la función *ConsultarActivo* implementada en el Chaincode. El resultado de la ejecución de esta consulta es la presentación de la información del Activo consultado presentándolo al usuario en formato json. La sección de código de la implementación de la función Consultar Registro se encuentra en el ANEXO III. Código fuente de la Aplicación, desde la línea 187 hasta la línea 204.

- **Función Eliminar Registro**

La función Eliminar Registro solicita al usuario el identificador del registro que se desea eliminar. Con dicho identificador se realiza una llamada a la función *EliminarActivo* implementada en el Chaincode. Como resultado de la ejecución de esta función se elimina el Activo del estado actual, es decir, los registros de dicho Activo existentes en la cadena de bloques se mantienen. La sección de código de la implementación de la función Eliminar Registro se encuentra en el ANEXO III. Código fuente de la Aplicación, desde la línea 205 hasta la línea 215.

- **Función Consultar Historial**

La función Consultar Historial solicita al usuario el identificador del registro del cual se requiere consultar su historial de cambios. Con dicho identificador se realiza una llamada a la función *HistorialdeActivo* implementada en el Chaincode. El resultado de la ejecución de esta consulta es la presentación de la historial de cambios del Activo consultado presentándolo al usuario como una sucesión de Activos en formato Json. Cabe recalcar que se puede consultar el historial de un registro eliminado ya que mediante esa función se consulta el estado del Activo en todas las cadenas de bloques almacenadas en el Libro mayor.

La sección de código de la implementación de la función Consultar Historial se encuentra en el ANEXO III. Código fuente de la Aplicación, desde la línea 216 hasta la línea 227.

- **Función Mostrar todos los registros**

Esta función muestra todos los Activos registrados en el SIDCFP, no solicita datos al usuario. Como resultado de esta consulta se presentan todos los Activos existentes en el estado actual del Libro mayor. Estos Activos se presentan al usuario en formato json. La sección de código de la implementación de la función Consultar Registro se encuentra en el ANEXO III. Código fuente de la Aplicación, desde la línea 228 hasta la línea 236.

3 PRUEBAS, RESULTADOS, CONCLUSIONES Y RECOMENDACIONES

Esta sección presenta las pruebas realizadas al prototipo para validar el diseño realizado en el presente TIC; así como el correcto funcionamiento tanto del *chaincode*, como de la aplicación implementadas. Esta sección, también, presenta las conclusiones obtenidas a partir de la realización de este TIC y las recomendaciones que se proponen para mejorar el diseño y funcionamiento del sistema de interconexión de datos para el control financiero a la política.

3.1 Pruebas

Se realizaron pruebas individuales de todas las funciones implementadas en la aplicación del prototipo del **SIDCFP**. A continuación, se presenta la ejecución de las funciones de la aplicación para observar la funcionalidad y los resultados de las transacciones y consultas implementadas en el prototipo. La ejecución y resultados de dichas pruebas se presenta a continuación.

3.1.1 Pruebas del funcionamiento del prototipo

Las pruebas funcionales realizadas al prototipo se estructuran con base en el menú diseñado para el prototipo del **SIDCFP**. En las pruebas se ejecutaron todas las opciones del menú para verificar su correcto funcionamiento.

3.1.1.1 Ejecución de la aplicación

Para ejecutar la aplicación el usuario debe ubicarse en el directorio fabric-samples/asset-transfer-basic/application-go/. A continuación, se ejecuta la aplicación mediante el siguiente comando:

- `go run assetTransfer.go`

En la **Figura 13** se presentan los resultados de la ejecución de dicho comando.

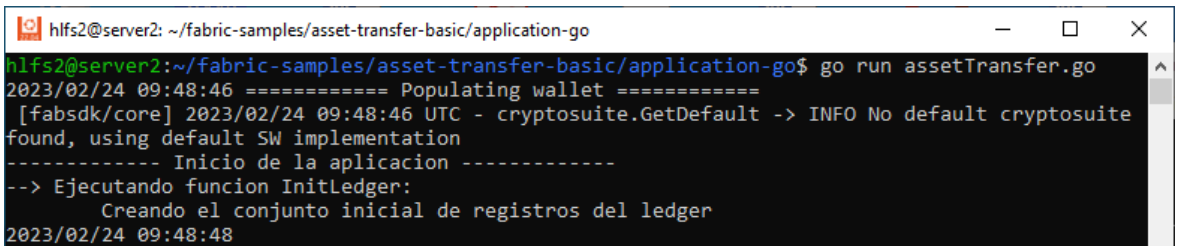

Figura 13: Ejecución de la aplicación del prototipo

Se observa el mensaje de creación de registros iniciales lo cual indica que la ejecución de la función InicializarLibro, y por lo tanto la inserción de los registros iniciales en el libro mayor, fue exitosa.

3.1.1.2 Menú de la aplicación

En la figura se presenta al usuario el menú con las ocho funciones disponibles que fueron implementadas en el prototipo de la aplicación. Estas funciones fueron definidas en la sección 2.2.3.2. Además, se presenta una función para finalizar la ejecución de la aplicación. El menú de la aplicación se presenta a continuación en la **Figura 14**.

Figura 14: Menú de la aplicación

3.1.1.3 Mostrar todos los registros

Para verificar que la función InicializarLibro del *chaincode* fue ejecutada correctamente, utilizamos la opción 8 del menú (*Mostrar todos los registros*). Al realizar esta consulta se presenta una lista en formato json de los seis activos registrados mediante la mencionada

función InicializarLibro, correspondientes a las organizaciones políticas de prueba definidos en dicha función.

La ejecución y los resultados de dicha función se puede observar a continuación en la **Figura 15**.

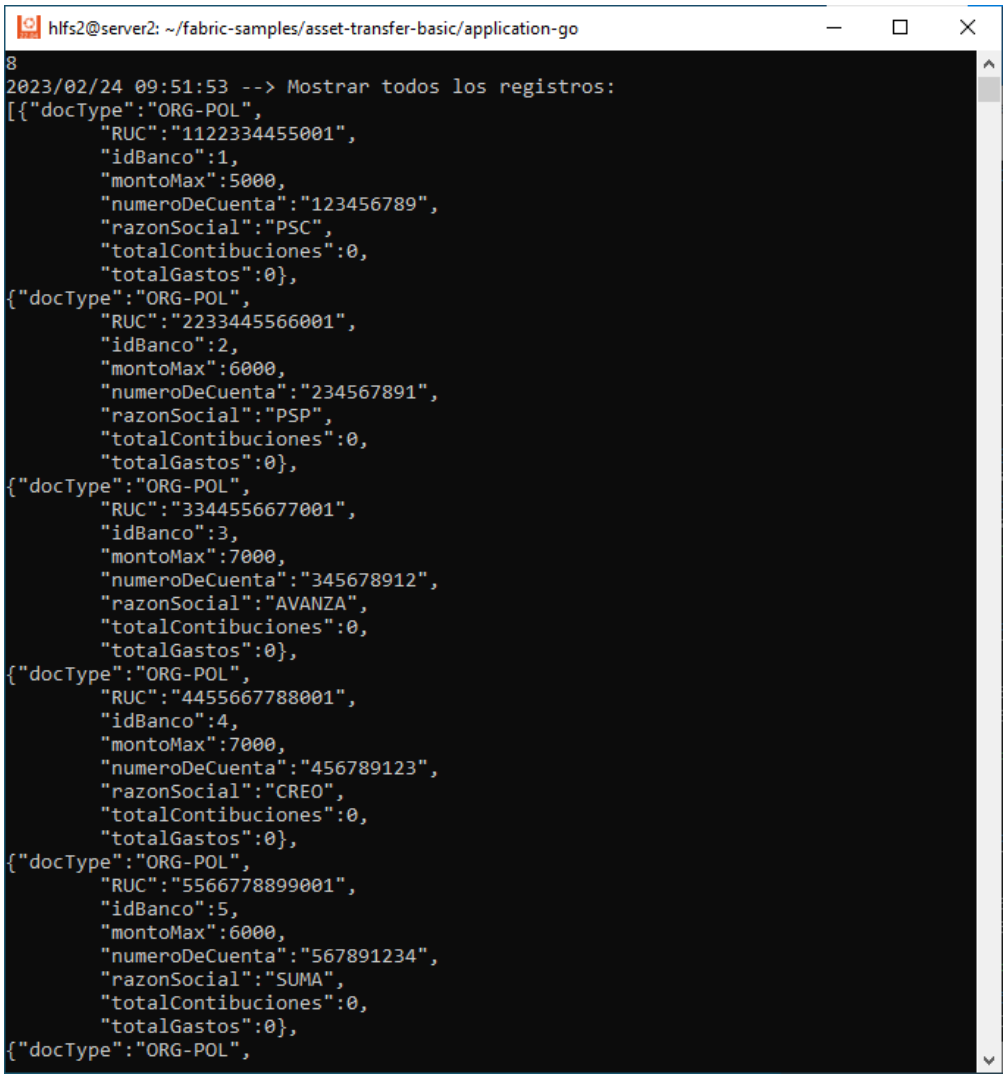

Figura 15: Mostrar todos los registros

Como resultado de la ejecución de dicha función se presenta al usuario los registros insertados tanto en la base de datos del libro mayor como en la base de datos del estado actual de dicho libro mayor. Y finalmente se presenta nuevamente el menú al usuario.

3.1.1.4 Registrar Aportante

A continuación, se realizará el registro de un aportante usando la función Registrar aportante. Para componer dicho registro se utilizó los siguientes datos:

- **RUC:** 1234567890001

- **Razón Social:** Aportante1
- **Banco:** 5
- **Número de cuenta:** 789456123

La ejecución de la función se realiza ingresando la opción 1 de la aplicación. La ejecución y resultado de dicha función se observa en la **Figura 16**.

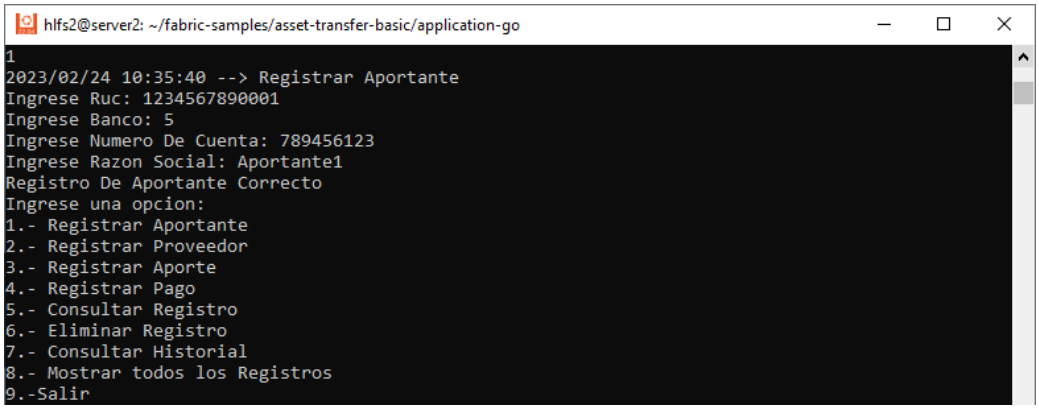

Figura 16: Registro de un aportante

Como resultado de la ejecución de dicho registro se observa el mensaje “Registro de Aportante Correcto”, lo cual indica que se realizó exitosamente el registro del Aportante. Finalmente se presenta nuevamente el menú al usuario.

3.1.1.5 Consultar Registro de aportante

Para verificar el registro del aportante se utilizó la Función Consultar Registro. Dicha función solicita al usuario el identificador del registro, para ello se utiliza el número de RUC del registro del aportante previamente creado.

- **RUC:** 1234567890001

La ejecución de dicha función se realiza ingresando la opción 5 de la aplicación. La ejecución y resultado de la presente consulta se observa a continuación en la **Figura 17**.

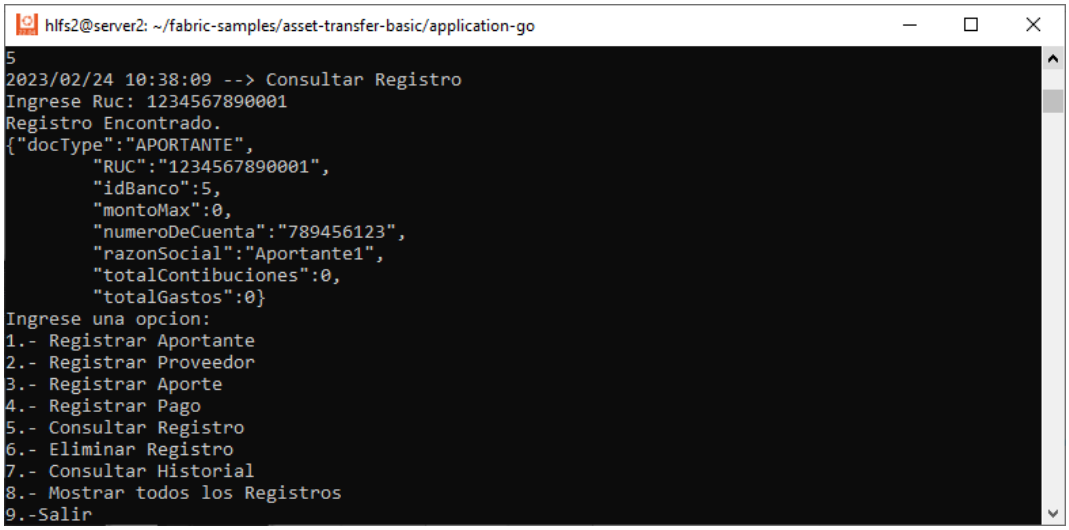

Figura 17: Consulta del registro de un aportante

Como resultado de la ejecución de dicha consulta se observa el registro en formato json del registro del aportante creado. Finalmente se presenta nuevamente el menú al usuario.

3.1.1.6 Registrar Aporte

A continuación, se realizará el registro de un aporte usando la función Registrar aporte. Para componer dicho registro se utilizó los siguientes datos:

- **RUC del Aportante:** 1234567890001
- **RUC de la Organización Política:** 1122334455001
- **Monto del Aporte:** 1000
- **Identificador del aporte:** AP-01

Para esta prueba se utilizó como identificador del aporte un identificador corto para facilitar su posterior consulta. En la implementación real del **SIDCPF** debe utilizarse como identificador el número de depósito o transferencia bancaria, tal como se definió en el diseño de las transacciones.

La ejecución de dicha función se realiza ingresando la opción 3 de la aplicación. La ejecución y resultado de dicho registro se observa en la **Figura 18**.

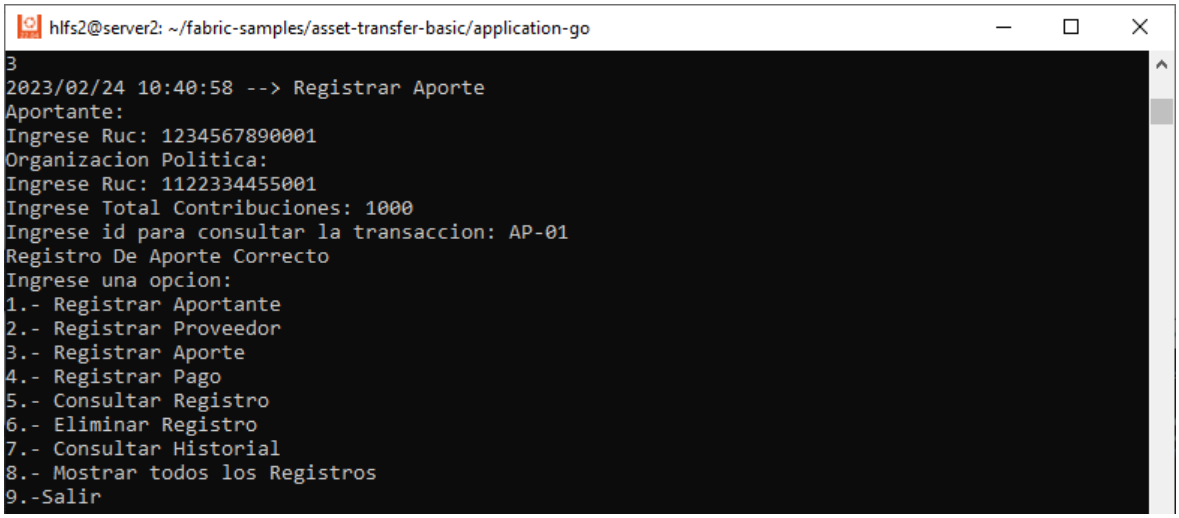

Figura 18: Registro de un aporte

Como resultado de la ejecución de dicho registro se observa el mensaje “Registro de Aporte Correcto”, lo cual indica que se realizó exitosamente el registro del Aporte. Finalmente se presenta nuevamente el menú al usuario.

3.1.1.7 Consultar Registro de aporte

Para verificar el registro del aporte se utilizó la Función Consultar Registro. Dicha función solicita al usuario el identificador del registro, para ello se utiliza el identificador del registro del aporte previamente creado.

- **Identificador del aporte:** AP-01

La ejecución de dicha función se realiza ingresando la opción 5 de la aplicación. La ejecución y resultado de la presente consulta se observa a continuación en la **Figura 19**.

Figura 19: Consulta del registro de un aporte

Como resultado de la ejecución de dicha consulta se observa el registro en formato json del registro del aporte creado. Finalmente se presenta nuevamente el menú al usuario.

3.1.1.8 Registrar Proveedor

A continuación, se realizará el registro de un proveedor usando la función Registrar proveedor. Para componer dicho registro se utilizó los siguientes datos:

- **RUC:** 9876543210001
- **Razón Social:** Proveedor1
- **Banco:** 6
- **Número de cuenta:** 456789123

La ejecución de la función se realiza ingresando la opción 3 de la aplicación. La ejecución y resultado de dicha función se observa en la **Figura 20**.

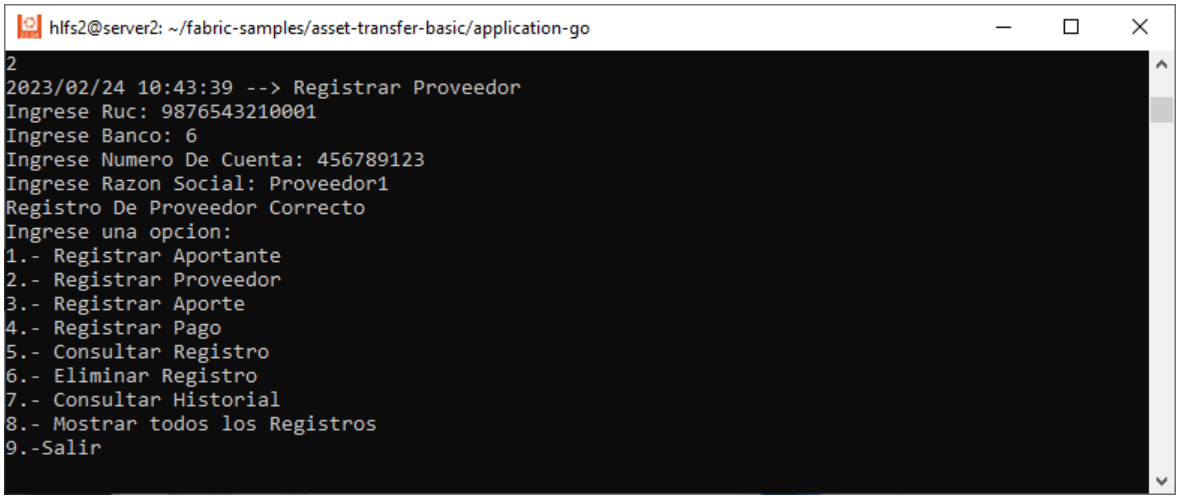

Figura 20: Registro de un proveedor

Como resultado de la ejecución de dicho registro se observa el mensaje “Registro de Proveedor Correcto”, lo cual indica que se realizó exitosamente el registro del Proveedor. Finalmente se presenta nuevamente el menú al usuario.

3.1.1.9 Consultar Registro de proveedor

Para verificar el registro del aporte se utilizó la Función Consultar Registro. Dicha función solicita al usuario el identificador del registro, para ello se utiliza el identificador del registro del proveedor previamente creado.

- **RUC:** 9876543210001

La ejecución de dicha función se realiza ingresando la opción 5 de la aplicación. La ejecución y resultado de la presente consulta se observa a continuación en la **Figura 21**

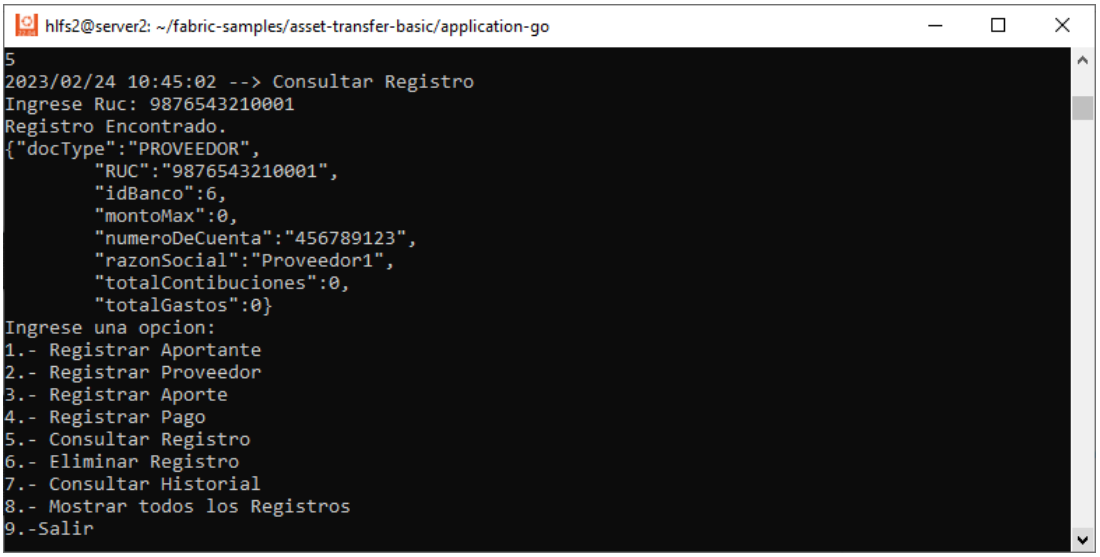

Figura 21: Consulta del registro de un proveedor

Como resultado de la ejecución de dicha consulta se observa el registro en formato json del registro del aporte creado. Finalmente se presenta nuevamente el menú al usuario.

3.1.1.10 Registrar Pago

A continuación, se realizará el registro de un proveedor usando la función Registrar pago. Para componer dicho registro se utilizó los siguientes datos:

- **RUC de la Organización Política:** 1122334455001
- **RUC del Aportante:** 9876543210001
- **Monto del pago:** 500
- **Identificador del pago:** PG-01

Para esta prueba se utilizó como identificador del aporte un identificador corto para facilitar su posterior consulta. En la implementación real del **SIDCPF** debe utilizarse como identificador el número de comprobante o factura, tal como se definió en el diseño de las transacciones.

La ejecución de la función se realiza ingresando la opción 4 de la aplicación. La ejecución y resultado de dicha función se observa en la **Figura 22**.

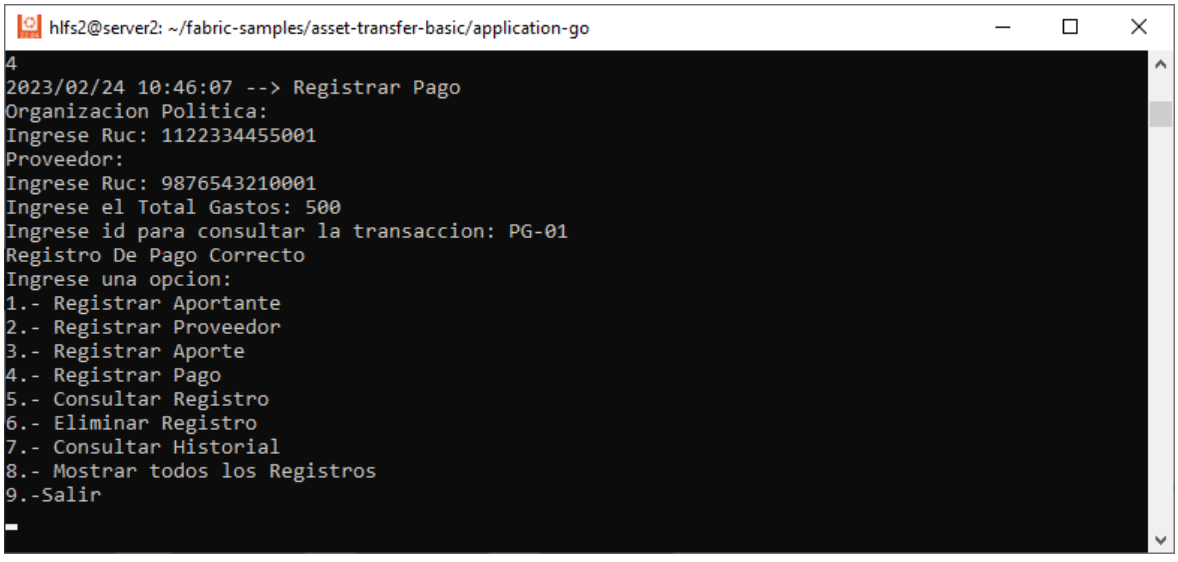

Figura 22: Registro de un pago

Como resultado de la ejecución de dicho registro se observa el mensaje “Registro de Pago Correcto”, lo cual indica que se realizó exitosamente el registro del Pago. Finalmente se presenta nuevamente el menú al usuario.

3.1.1.11 Consultar Registro de pago

Para verificar el registro del aportante se utilizó la Función Consultar Registro. Dicha función solicita al usuario el identificador del registro, para ello se utiliza el número de RUC del registro del aportante previamente creado.

- **Identificador del aporte:** PG-01

La ejecución de dicha función se realiza ingresando la opción 5 de la aplicación. La ejecución y resultado de la presente consulta se observa a continuación en la **Figura 23**.

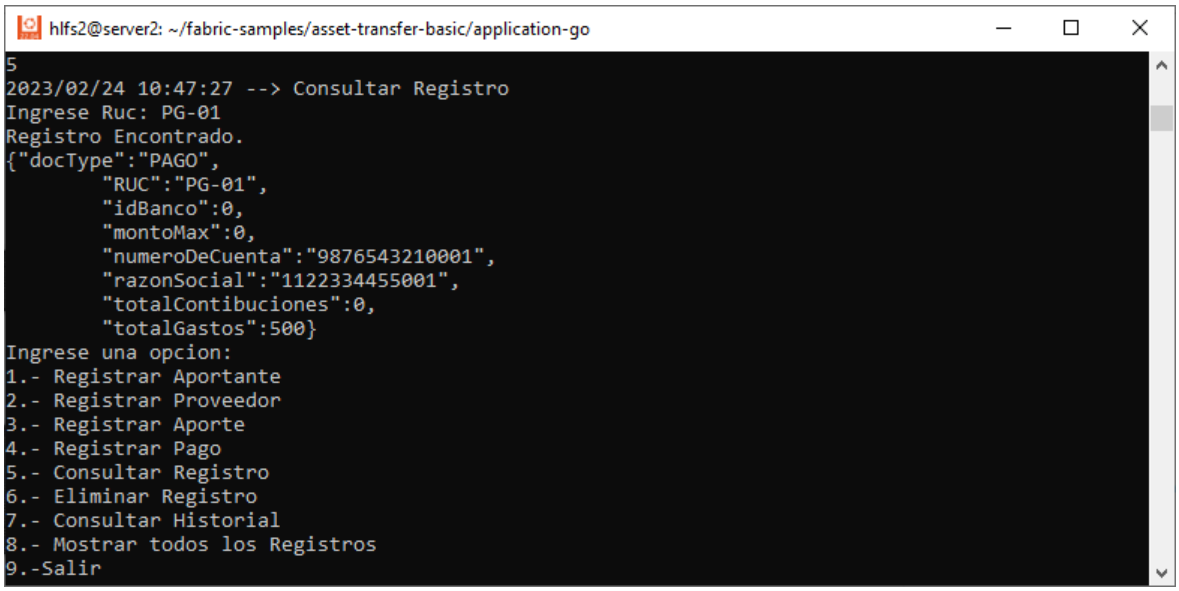

Figura 23: Consulta del registro de un pago

Como resultado de la ejecución de dicha consulta se observa el registro en formato json del registro del pago creado. Finalmente se presenta nuevamente el menú al usuario.

3.1.1.12 Consultar Registro de organización política

Para verificar el registro del aportante se utilizó la Función Consultar Registro. Dicha función solicita al usuario el identificador del registro, para ello se utiliza el número de RUC del registro del aportante previamente creado.

- **RUC:** 1122334455001

La ejecución de dicha función se realiza ingresando el valor 5. La ejecución y resultado de la presente consulta se observa a continuación en la **Figura 24**.

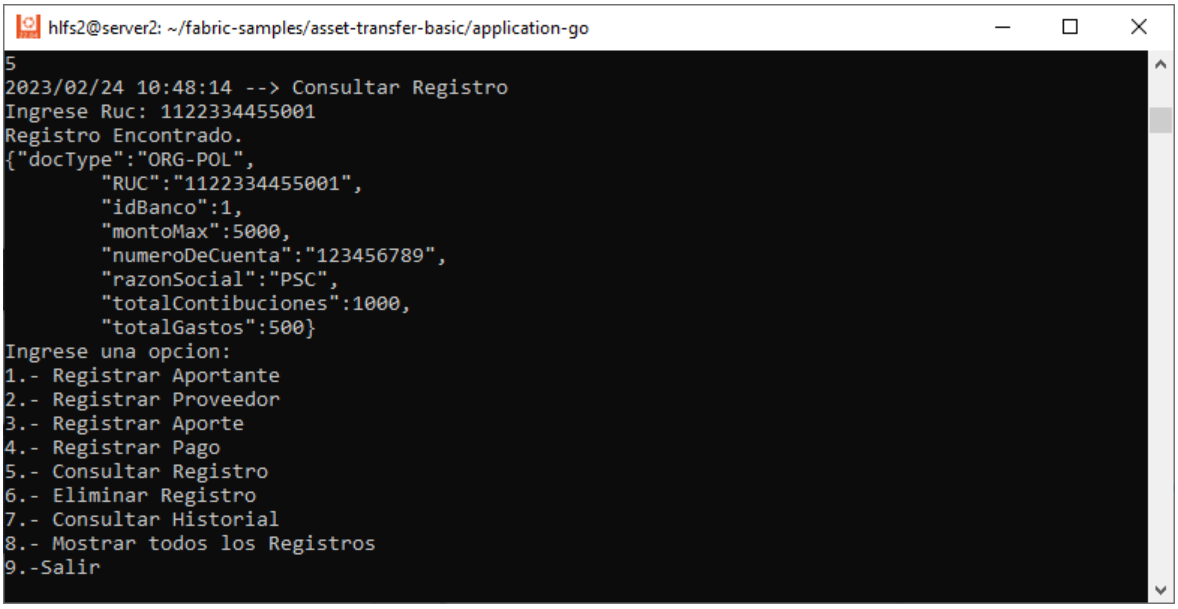

Figura 24: Consulta del registro de una organización política

Como resultado de la ejecución de dicha consulta se observa el registro de la organización política en el que se presenta el monto del aporte y monto del pago realizado. Finalmente se presenta nuevamente el menú al usuario.

3.1.1.13 Eliminar registro de organización política

Para eliminar el registro de la organización se utilizó la función Eliminar Registro. Dicha función solicita al usuario el identificador del registro, para ello se utiliza el número de RUC del registro de la organización política.

- **RUC de la Organización Política:** 1122334455001

La ejecución de dicha función se realiza ingresando la opción 6 de la aplicación. La ejecución y resultado de la presente consulta se observa a continuación en la **Figura 25**.

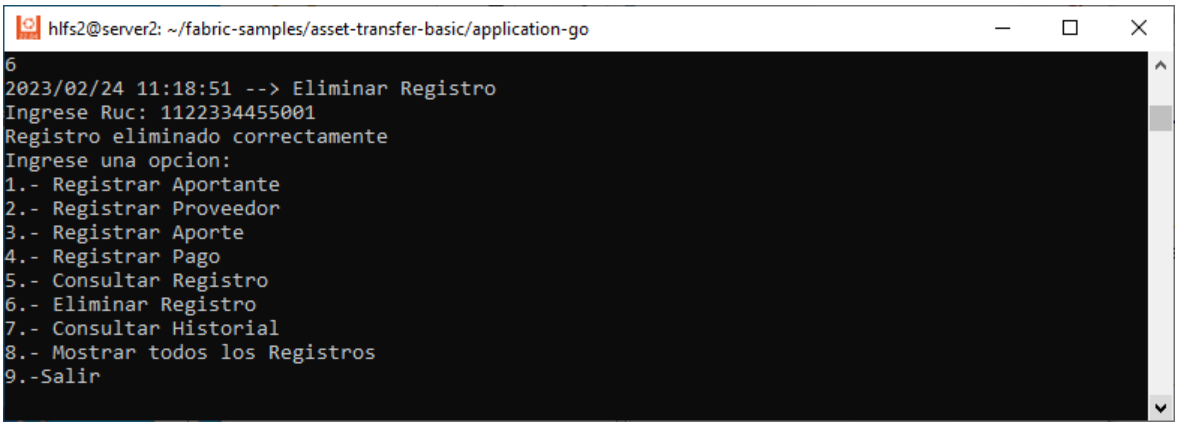

Figura 25: Eliminación del registro de una organización política

Como resultado de la ejecución de dicho registro se observa el mensaje “Registro eliminado correctamente”, lo cual indica que se realizó exitosamente la eliminación de la organización política del estado actual del libro mayor. Como se explicó en secciones anteriores esta función no elimina el registro, si no cambia su estado a “eliminado”. Finalmente se presenta nuevamente el menú al usuario.

3.1.1.14 Consultar Historial de transacciones de organización política

Para ello se utiliza el número de ruc del Asset de la organización política.

- **RUC de la Organización Política:** 1122334455001

La ejecución de dicha función se realiza ingresando la opción 7 de la aplicación. La ejecución y resultado de la presente consulta se observa a continuación en la **Figura 26**.

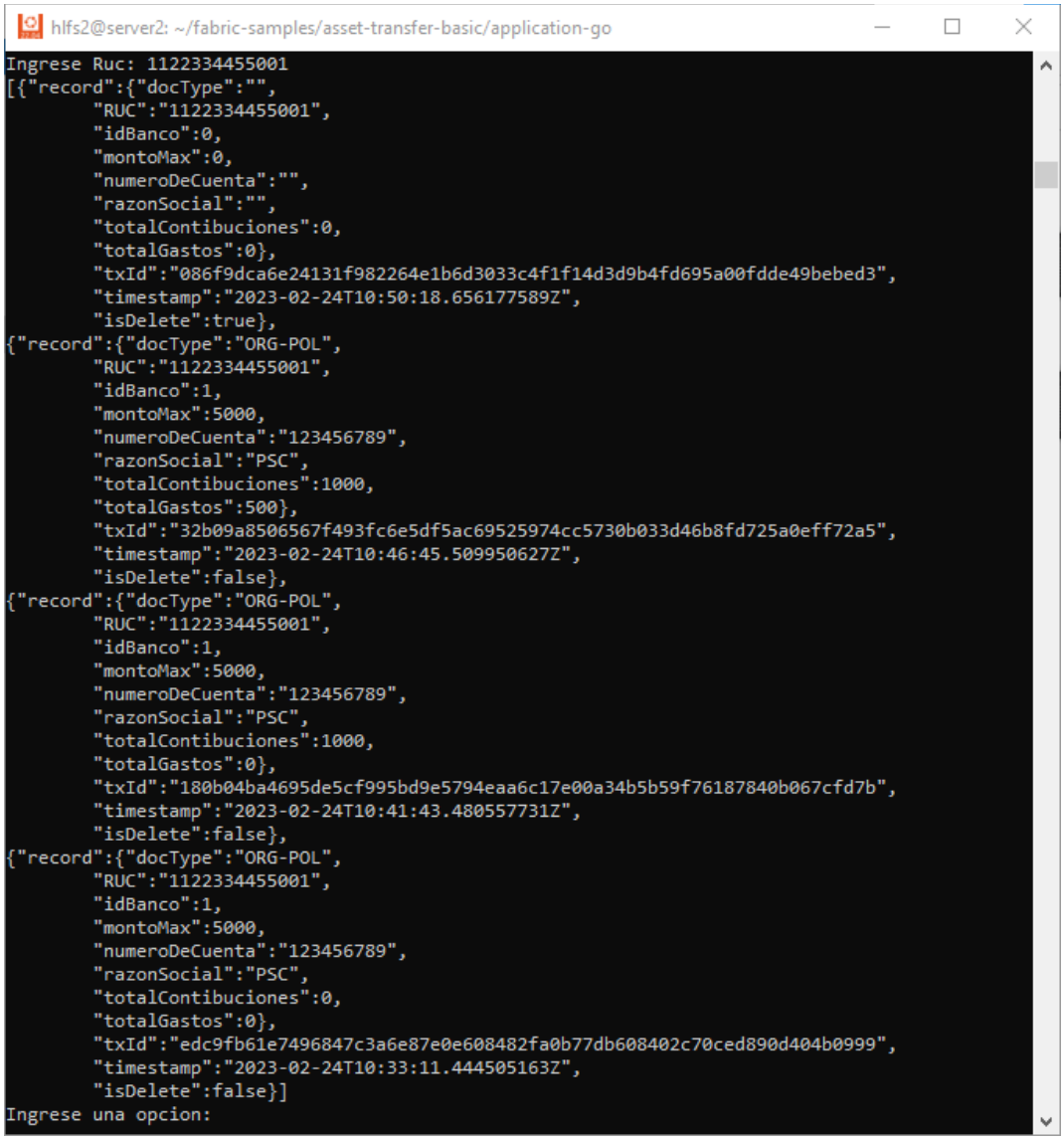

Figura 26: Consulta del historial de un activo de organización política

Como resultado se puede observar que existen 4 registros: estos corresponden a la creación del Activo mediante la función inicialización del Libro, registro del aporte, registro del pago y finalmente eliminación del registro, se presentan en orden desde el registro más reciente hasta el más antiguo. Estos registros se encontraron recorriendo la cadena de bloques, de modo que, el usuario puede verificar el historial de cambios por los que ha atravesado un Activo durante el ciclo de vida de la campaña electoral.

3.1.1.15 Salir de la aplicación

Finalmente se utilizó la función Salir para finalizar la ejecución de la aplicación. La ejecución de dicha función se realiza ingresando la opción 9 de la aplicación. La ejecución y resultado de la presente función se observa a continuación en la **Figura 27**.

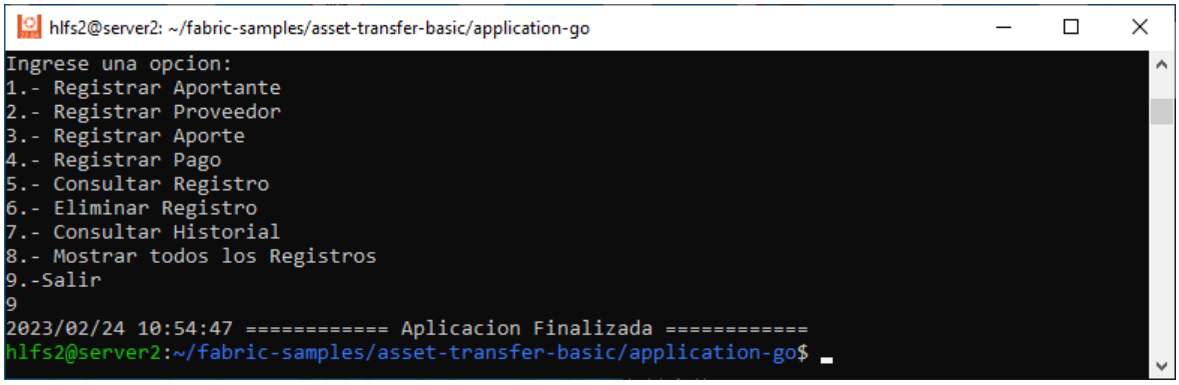

Figura 27: Finalización de la aplicación

3.2 Resultados

En este TIC logramos cumplir los objetivos. Se diseño un sistema de interconexión de datos que responde a los requerimientos legales y normativos y a los requerimientos tecnológicos. El prototipo permite validar dicho diseño. A través de las pruebas descritas en la sección Pruebas, se demuestra el funcionamiento correcto del prototipo. El lector puede probar el funcionamiento del prototipo realizando el tutorial de instalación de los prerrequisitos que se encuentra en el ANEXO I. Instalación de prerrequisitos y descargar el Chaincode y la aplicación del repositorio de GitHub ubicado en el ANEXO IV. Repositorio de código fuente.

3.3 Conclusiones

En esta sección se presentan las conclusiones obtenidas de la elaboración del este trabajo de integración curricular. A continuación, se presentan las conclusiones a partir de los respectivos objetivos planteados:

- **Objetivo 1:** *Transparentar el control del gasto electoral durante las campañas políticas*. En este TIC, se demuestra que es posible transparentar el control de gasto electoral al utilizar tecnologías de contabilidad distribuidas y de cadenas de bloques en una red permisionada, como la plataforma Hyperledger Fabric, misma que fue utilizada para el desarrolló este TIC. Con esta implementación se puede realizar el seguimiento desde el origen del financiamiento de las organizaciones políticas hasta los gastos realizados por las mismas durante el ciclo de vida de la campaña electoral. Además, con la automatización de la recolección de datos para la creación de los registros inmutables de las organizaciones políticas y la implementación de transacciones como historial de un registro se facilita el trabajo de auditoría a las entidades de control y a las partes interesadas en la integridad del financiamiento de las campañas electorales.
- **Objetivo 2:** *Proporcionar una estructura descentralizada para el control del gasto electoral*. Se proporciona una estructura para descentralizar el control del proceso de gestión del gasto electoral. Mediante esta estructura de control descentralizada pueden participar las entidades de control del gobierno central en el control del financiamiento y gastos de las organizaciones políticas. El sistema SIDCFP permite que cada entidad de control obtenga acceso inmediato a los datos del sistema en sus propias infraestructuras tecnológicas mediante los cuales debe realizar sus procesos de auditoría e investigación.
- **Objetivo 3:** *Automatizar los procesos de control del gasto electoral*. Se logra automatizar los procesos de control de gasto electoral mediante el diseño de las transacciones definidas en la sección **Diseño de la aplicación** y su posterior implementación en el Chaincode y en la Aplicación del prototipo del SIDCFP.
- **Objetivo 4:** *Diseñar el sistema de interconexión de datos para control del financiamiento a la política*, de acuerdo con la *disposición transitoria segunda del Código de la Democracia*. Se presenta un diseño para el **SIDCFP** que responde a los requerimientos obtenidos a partir del análisis de las respectivas leyes y normativas vigentes en el país.
- **Objetivo 5:** *Validar el mencionado diseño mediante la construcción y pruebas de un prototipo*. Se demostró la validez del diseño propuesto mediante la construcción, ejecución y pruebas del prototipo con lo que se confirma que el sistema permisionado de cadena de bloques de la plataforma Hyperledger Fabric, son las tecnologías y herramientas indicadas para solucionar los problemas de falta de integridad del control del financiamiento y gasto electoral.

3.4 Recomendaciones

- Se recomienda tomar en cuenta la automatización sugerida en este TIC de evitar la implementación de una plataforma web para recopilar la información del financiamiento de las organizaciones políticas planteada en la sección 2.1.2.3. Automatizar el proceso de recopilación de la información acerca del financiamiento y gastos de las organizaciones políticas es vital para conseguir transparencia e integridad del financiamiento de las campañas electorales. Dicha automatización es uno de los requerimientos considerados en la construcción de este **SIDCFP**.
- En el diseño de la aplicación se propone una transacción para la eliminación de los Activos, que, aunque solo los elimina del estado actual, estos permanecen en las cadenas de bloques de las transacciones. Se recomienda que esta transacción no se encuentre disponible para ninguna entidad de control participante o que esta no sea implementada en el **SIDCFP**. Tal como se explicó en la sección 2.2.3.2, dicha transacción fue considerada para fines didácticos, únicamente.
- Existen proveedores de soluciones de Hyperledger Fabric en la nube, mediante las que se puede facilitar la implementación de la infraestructura tecnológica, así como del middleware de este **SIDCFP**. A continuación, se presentan dos proveedores de servicios de Hyperledger Fabric en la nube. IBM posee una plataforma en la nube basada en Hyperledger Fabric IBM Blockchain Platform ofrece herramientas y soporte completo, beneficios que no se pueden obtener al utilizar Blockchain de código abierto gratuito. Del mismo modo, AWS Amazon Web Services provee Hyperledger Fabric como servicio en la nube mediante el cual proporciona la infraestructura necesaria acompañada de tutoriales para levantar y configurar el servicio, desde los prerrequisitos y creación de la red hasta la ejecución del Chaincode en un canal con múltiples participantes.

4 REFERENCIAS BIBLIOGRÁFICAS

[1] Asamblea Nacional de la Republica del Ecuador, LEY ORGÁNICA REFORMATORIA A LA LEY ORGÁNICA ELECTORAL Y DE ORGANIZACIONES POLÍTICAS, CÓDIGO DE LA DEMOCRACIA. Suplemento – Registro Oficial Nº 134, Quito: Registro Oficial, Órgano de la Republica del Ecuador, 2020.

[2] CNE Consejo Nacional Electoral, REGLAMENTO PARA EL CONTROL Y FISCALIZACIÓN DEL GASTO ELECTORAL, Quito: Registro Oficial, Órgano de la República del Ecuador, 2020.

[3] ISO, «Sistemas de gestión de la calidad — Requisitos específicos para la aplicación de la Norma ISO 9001:2015 a organizaciones electorales en todos los niveles de gobierno,» ISO/TS 54001:2019(es).

[4] Hyperledger, «Hyperledger Fabric – Hyperledger Foundation,» Hyperledger Foundation, 26 Junio 2022. [En línea]. Available: https://www.hyperledger.org/use/fabric. [Último acceso: 27 12 2022].

[5] Hyperledger Fabric, «Using the Fabric test network,» 4 Diciembre 2019. [En línea]. Available: https://hyperledger-fabric.readthedocs.io/es/latest/test_network.html. [Último acceso: 9 Noviembre 2022].

[6] github, «hyperledger/fabric-samples,» github, 16 Diciembre 2019. [En línea]. Available: https://github.com/hyperledger/fabric-samples. [Último acceso: 8 Noviembre 2022].

[7] G. Coulouris, J. Dollimore y T. Kindberg, Sistemas distribuidos, Conceptos y diseño, Madrid: Pearson Educación S. A, 2001.

[8] Kriste, «Algoritmo de consenso,» 9 Mayo 2022. [En línea]. Available: https://tech-dir.net/algoritmo-de-consenso/#:~:text=Un%20algoritmo%20de%20consenso%20es%20un%20proceso%20en,una%20red%20que%20involucra%20m%C3%BAltiples%20nodos%20no%20confiables. [Último acceso: 17 Febrero 2023].

[9] I. Gallardo, P. Bazan y P. Venosa, «Análisis del anonimato aplicado a criptomonedas,» 18 Octubre 2019. [En línea]. Available: https://core.ac.uk/download/pdf/296434137.pdf. [Último acceso: 10 Febrero 2023].

[10] J. Maldonado, «¿Qué es Prueba de trabajo / Proof of Work (PoW)?,» 6 Julio 2018. [En línea]. Available: https://academy.bit2me.com/que-es-proof-of-work-pow/. [Último acceso: 30 Enero 2023].

[11] M. Jiménez, «Distribución de Datos Basada en Blockchain: Nivel Control,» Junio 2022. [En línea]. Available: https://oa.upm.es/71306/1/TFG_SERGIO_PARDO_DE_LA_BORBOLLA.pdf. [Último acceso: 5 Enero 2023].

[12] «Data Integrity: Detecting and Responding to Ransomware and Other Destructive Events,» 1 Octubre 2020. [En línea]. Available:

https://www.nccoe.nist.gov/publication/1800-26/VolA/index.html. [Último acceso: 30 Enero 2023].

[13] A. López, «Todo sobre criptografía: Algoritmos de clave simétrica y asimétrica,» 5 Diciembre 2022 . [En línea]. Available: https://www.redeszone.net/tutoriales/seguridad/criptografia-algoritmos-clave-simetrica-asimetrica/. [Último acceso: 15 Febrero 2023].

[14] H. Sorenson, «An Introduction to OpenSSL, Part Three: PKI- Public Key Infrastructure,» 2001 Septiembre 19. [En línea]. Available: https://community.broadcom.com/symantecenterprise/communities/community-home/librarydocuments/viewdocument?DocumentKey=ca689f42-70a8-41f6-8a00-1ff13812bf08&CommunityKey=1ecf5f55-9545-44d6-b0f4-4e4a7f5f5e68&tab=librarydocuments. [Último acceso: 16 Febrero 2023].

[15] appviewx, «Public-Key-Infrastrutcure-2,» appviewx, Diciembre 2020. [En línea]. Available: https://www.appviewx.com/wp-content/uploads/2020/12/Public-Key-Infrastrutcure-2.png. [Último acceso: 9 Enero 2023].

[16] IBM, «Alta disponibilidad,» 14 Abril 2021. [En línea]. Available: https://www.ibm.com/docs/es/i/7.3?topic=availability-high. [Último acceso: 1 Enero 2023].

[17] J. L. P. Luján, «Revisión de los Sistemas de Comunicaciones más empleados en Control Distribuido,» 10 Noviembre 2009. [En línea]. Available: https://riunet.upv.es/bitstream/handle/10251/6408/Comunicaciones%20en%20los%20sistemas%20distribuidos.pdf. [Último acceso: 15 Enero 2023].

[18] IBM, «Protocolos TCP/IP,» 12 Abril 2021. [En línea]. Available: https://www.ibm.com/docs/es/aix/7.2?topic=protocol-tcpip-protocols. [Último acceso: 18 Enero 2023].

[19] IBM, «Protocolo eXternal Data Representation,» 12 Abril 2021. [En línea]. Available: https://www.ibm.com/docs/es/aix/7.2?topic=system-external-data-representation-protocol. [Último acceso: 2 Enero 2023].

[20] IBM, «API REST,» 6 Abril 2021. [En línea]. Available: https://www.ibm.com/mx-es/cloud/learn/rest-apis. [Último acceso: 28 Enero 2023].

[21] Microsoft Learn, «gRPC,» 14 Febrero 2023. [En línea]. Available: https://learn.microsoft.com/es-es/dotnet/architecture/cloud-native/grpc. [Último acceso: 20 Febrero 2023].

[22] Protocol Buffers Documentation, «Overview | Protocol Buffers Documentation,» 28 Noviembre 2022. [En línea]. Available: https://protobuf.dev/overview/. [Último acceso: 20 Febrero 2023].

[23] S. Frankel, NIST, S. Krishnan y Ericsson, «IP Security (IPsec) and Internet Key Exchange (IKE) Document Roadmap,» Febrero 2011. [En línea]. Available: https://www.rfc-editor.org/rfc/rfc6071#section-2. [Último acceso: 10 Enero 2023].

[24] E. Rescorla, «The Transport Layer Security (TLS) Protocol Version 1.3,» Agosto 2018. [En línea]. Available: https://www.rfc-editor.org/rfc/pdfrfc/rfc8446.txt.pdf. [Último acceso: 27 Diciembre 2022].

[25] B. Farahani, F. Firouzi y M. Luecking, «The convergence of IoT and distributed ledger technologies (DLT): Opportunities, challenges, and solutions.,» Journal of Network and Computer Applications, vol. 177, nº 102936, 2021.

[26] Association for Development of Financial, «Distributed ledger technology regulatory framework,» 1 Agosto 2019. [En línea]. Available: https://www.itu.int/en/ITU-T/focusgroups/dlt/Documents/d41.pdf. [Último acceso: 3 Enero 2023].

[27] TIBCO Software, «¿Qué es la arquitectura basada en eventos?,» 5 Marzo 2020. [En línea]. Available: https://www.tibco.com/es/reference-center/what-is-event-driven-architecture. [Último acceso: 15 Febrero 2023].

[28] IBM, «¿Qué es la tecnología de blockchain?,» IBM, 26 Marzo 2021. [En línea]. Available: https://www.ibm.com/co-es/topics/what-is-blockchain. [Último acceso: 10 Noviembre 2022].

[29] Hyperledger Fabric, «Open, Proven, Enterprise-grade DLT,» Marzo 2020. [En línea]. Available: https://www.hyperledger.org/wp-content/uploads/2020/03/hyperledger_fabric_whitepaper.pdf. [Último acceso: 15 Noviembre 2022].

[30] Hyperledger Fabric, «Modelo de Hyperledger Fabric,» 8 Octubre 2020. [En línea]. Available: https://hyperledger-fabric.readthedocs.io/es/latest/fabric_model.html. [Último acceso: 26 Noviembre 2022].

[31] IBM, «¿Qué es Hyperledger Fabric?,» IBM, 25 Marzo 2021. [En línea]. Available: https://www.ibm.com/es-es/topics/hyperledger. [Último acceso: 20 Noviembre 2022].

[32] En Hyperledger Fabric, «Glosario,» 30 Octubre 2020. [En línea]. Available: https://hyperledger-fabric.readthedocs.io/es/latest/glossary.html. [Último acceso: 23 Diciembre 2022].

[33] Hyperledger Fabric, «Peers,» 8 Octubre 2020. [En línea]. Available: https://hyperledger-fabric.readthedocs.io/es/latest/peers/peers.html#peers. [Último acceso: 17 Diciembre 2022].

[34] Hyperledger Fabric, «The Ordering Service,» 29 Marzo 2019. [En línea]. Available: https://hyperledger-fabric.readthedocs.io/en/release-2.5/orderer/ordering_service.html. [Último acceso: 1 Diciembre 2022].

[35] Hyperledger Fabric, «Blockchain network,» 22 Agosto 2018. [En línea]. Available: https://hyperledger-fabric.readthedocs.io/en/release-2.2/network/network.html. [Último acceso: 19 Noviembre 2022].

[36] Hyperledger Fabric, «Channels,» 18 Octubre 2020. [En línea]. Available: https://hyperledger-fabric.readthedocs.io/en/latest/channels.html. [Último acceso: 13 Diciembre 2022].

[37] Hyperledger Fabric, «Ledger,» 14 Octubre 2020. [En línea]. Available: https://hyperledger-fabric.readthedocs.io/es/latest/ledger.html. [Último acceso: 3 Diciembre 2022].

[38] Hyperledger Fabric, «Smart Contracts and Chaincode,» 27 Enero 2019. [En línea]. Available: https://hyperledger-fabric.readthedocs.io/en/release-2.5/smartcontract/smartcontract.html. [Último acceso: 22 Diciembre 2022].

[39] Grupo faro, «Financiamiento de la política: Vía a la transparencia,» 5 Enero 2021. [En línea]. Available: https://elecciones2021.ecuador-decide.org/wp-content/uploads/2021/02/Factsheet-Financiamiento-de-la-política.pdf. [Último acceso: 20 Febrero 2023].

[40] Intel, «Intel® Advanced Encryption Standard (AES) New Instructions Set,» Mayo 2010. [En línea]. Available: https://www.intel.com/content/dam/doc/white-paper/advanced-encryption-standard-new-instructions-set-paper.pdf. [Último acceso: 11 Enero 2023].

5 ANEXOS

ANEXO I. Instalación de prerrequisitos

ANEXO I. Instalación de prerrequisitos.pdf

ANEXO II. Código fuente Chaincode

ANEXO II. Código fuente Chaincode.pdf

ANEXO III. Código fuente de la Aplicación

ANEXO III. Código fuente de la aplicación.pdf

ANEXO IV. Repositorio de código fuente

https://github.com/VictorMaiguashca/SIDCFP.git