UNIVERSIDAD TECNOLÓGICA NACIONAL

Facultad Regional Córdoba
Ingeniería en Sistemas de Información

Kratia

Cátedra: Proyecto Final
Curso: 5k3

Autores:

- Wismer Axel
- Morardo Diego
- Bossio Mateo

Profesores:

- Destefanis María Laura
- Quinteros Sergio

Kratia

Elecciones 2021
Lista 5K3 G11

Candidato a mejor producto

Kratia es un sistema de votación electrónica basado en **blockchain**

¿Por qué Blockchain?

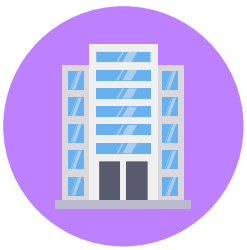
La tecnología blockchain nos permite el registro seguro e inmutable de los votos, logrando así una elección 100% segura y transparente

Creá tus convocatorias a elecciones completamente personalizadas con nuestro módulo de administración

Votá de forma segura y transparente gracias a nuestro algoritmo. Así se consigue el secreto, la integridad y la verificación del voto

Consultá los resultados de la elección de forma inmediata. Utilizá nuestro módulo de Auditoria para corroborar la información registrada

Aplicable a...



Empresas

ONGs

Gobiernos

UTN
FACULTAD REGIONAL CÓRDOBA
INGENIERÍA EN SISTEMAS DE INFORMACIÓN

PROYECTO FINAL 2021

DOCENTES
Ing. Quinteros, Sergio
Ing. Destefanis Ma. Laura

AUTORES
Bossio, Mateo 75428
Morardo, Diego 75527
Wismer, Axel 75930

GitHub · Jira · gRPC · HYPERLEDGER · BLOCKCHAIN · node · docker

2021

Sistema de Votación Electrónico basado en Blockchain

Bossio, Mateo; Morardo, Diego; Wismer, Axel
Universidad Tecnológica Nacional, Facultad Regional Córdoba

Abstract
Kratia fue creado como un sistema de votación electrónico basado en la tecnología Blockchain para llevar a cabo elecciones en pequeñas y grandes organizaciones, ONGs y gobiernos de forma segura y transparente.
Este sistema permitió a usuarios validados contra un padrón previamente cargado votar a sus representantes en elecciones en las que participan múltiples candidatos que ocupan un cargo particular y conforman una lista.
Se trabajó con una red basada en Blockchain utilizando Hyperledger Fabric, haciendo uso de contenedores Docker para hospedar los nodos de la misma, una API programada con Express que actúa como pasarela y un sitio web que implementa React para manejar las solicitudes de los usuarios. Además, se agregó el servicio de Explorer para analizar las transacciones ejecutadas y registradas en la blockchain.

Palabras Clave
Kratia, votación, electrónica, e-government, blockchain, Hyperledger, Fabric, Producto.

Introducción

Kratia es una alternativa a las herramientas utilizadas actualmente para llevar a cabo elecciones. Los sistemas de votación tradicionales conllevan altos costos derivados de cuestiones de seguridad, cantidad de personal, tiempos extensos y representan un gran impacto negativo para el medio ambiente, principalmente por las excesivas impresiones de papel y emisiones de gases dañinos.

Los sistemas electrónicos solucionan gran parte de estas problemáticas, pero surge el inconveniente de las vulnerabilidades de seguridad que estos presentan, por lo que existen barreras de confianza que deben superarse.

Kratia combina las ventajas de un sistema electrónico con la seguridad y fiabilidad de un sistema tradicional. La tecnología Blockchain permite que el sistema se mantenga seguro ya que los votos son inmutables y totalmente secretos.

En resumen, Kratia es un sistema de votación electrónico transparente, descentralizado, seguro, de bajo costo y sustentable; destinado a cualquier tipo de organizaciones, ya sean empresas privadas, ONGs o gobiernos.

Elementos del Trabajo y metodología

Para la gestión del proyecto se siguieron los estándares definidos por el Project Management Institute[1] que contiene las mejores prácticas en esta área. En cuanto a metodología de trabajo se optó por trabajar con Scrum[2], se definieron 5 Sprints de 1 mes de duración cada uno, dedicando el primero a tareas de investigación de las tecnologías elegidas. A lo largo del proyecto se trabajó junto a un Product Owner que entregó un feedback en las reuniones de revisión y planificación de cada sprint. En cuanto a la administración del proyecto se utilizó la herramienta Jira[3] para hacer el seguimiento de las actividades. Se optó por Github como herramienta de versionado de código durante el avance del proyecto. En el proceso de desarrollo se aplicó Test Driven Development[4] para guiar la codificación por casos de prueba diseñados con anterioridad. En cuanto a las tecnologías se utilizó HyperLedger Fabric[5] (HF) para crear una red de nodos en donde se almacenen los datos de forma distribuida y empleando la técnica de encriptación característica de Blockchain[6]. Es importante aclarar que HF utiliza contenedores de Docker[7] para hospedar estos distintos nodos de la red. También se

diseñó un framework llamado King para agilizar el desarrollo y abstraer la complejidad de las interacciones entre los objetos creados y su almacenamiento en la blockchain. Se creó una plataforma web utilizando React[8] para permitir a los usuarios interactuar con los registros del sistema, invocando las funciones definidas en los contratos desplegados en los distintos canales de la red que representan las funcionalidades del negocio abordado. También se creó una API utilizando el framework Express[9] para comunicar estos dos componentes del sistema y para abstraer las diferencias tecnológicas de los mismos, proporcionando gran flexibilidad ante los cambios. Por último, se implementó la herramienta Explorer[10] para analizar cada transacción registrada en la red con fines de auditoría.

Arquitectura

Se decidió utilizar una arquitectura cliente-servidor con 3 clientes, cada uno compuesto por un servidor React y una API que permita su comunicación con la blockchain. Dentro de esta se utilizaron 2 canales, el canal de usuarios y de votación, los cuales serán a futuro administrados por diferentes grupos de organizaciones. Esto permite mantener el secreto del voto, mediante la separación física de los datos críticos (figura 1).

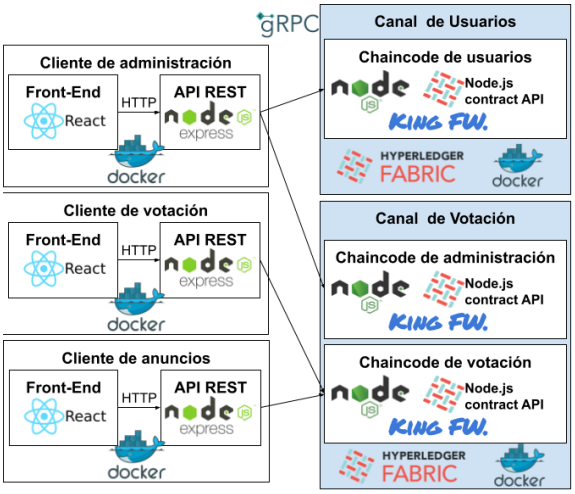


Figura 1.

El canal de usuarios se encarga del manejo del padrón, los usuarios y sus permisos mientras que el canal de votación maneja la votación y todas las actividades administrativas previas que deben realizarse.

La arquitectura del sistema se planteó de forma que sea posible mantener el secreto del voto y su integridad incluso en el caso de que múltiples organizaciones o servidores se vean comprometidas dado que nunca se encontrará toda la información de un voto en el mismo lugar y gracias al funcionamiento de Fabric ninguna organización puede registrar una transacción inválida en la blockchain.

Votación

Se identificó como la transacción principal del sistema. Es realizada por un votante que envía una petición al cliente de votación, el cual se comunica con el canal de usuarios para obtener el alias de usuario, este alias es un identificador único y efímero que acredita al votante como válido mientras le permite ocultar su identidad al registrar el voto. El cliente de votación se comunica con el canal de votación para registrar el voto. Este recibe el alias y el voto, verifica el alias en el canal de usuarios, verifica el voto, lo registra junto con un Hash(producto de una función criptográfica unidireccional)[11] del alias y retorna un comprobante generado como el Hash del alias y el voto. Finalmente, el cliente de votación comprueba que el comprobante de votación no se haya modificado durante la transacción. Al finalizar la votación se eliminan las claves de encriptación y los alias generados para eliminar toda relación entre el votante y su voto (figura 2).

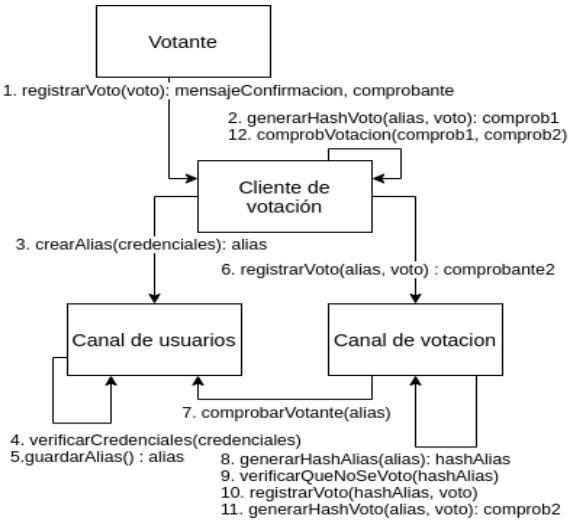


Figura 2.

Framework

Para el desarrollo de los smart contracts se creó el Framework King el cual se basa en el popular framework Django[12] e incorpora las buenas prácticas de IBM en el desarrollo de contratos inteligentes que se encuentran en los ejemplos de Fabric. King está programado en Node js y sigue un modelo Model View con las siguientes clases (figura 3).

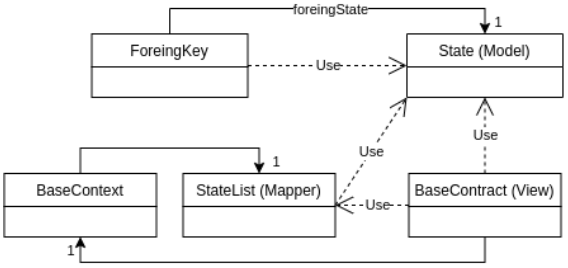


Figura 3.

State: Se encarga de la serialización y deserialización de objetos para su guardado en la blockchain y para la comunicación con las aplicaciones de usuario.

StateList: Mapea los estados representados como objetos en memoria a la blockchain y maneja su guardado, consulta y eliminación.

BaseContract: Clase abstracta que hereda de la clase Contract del SDK de Fabric, que permite gestionar los estados en la blockchain. Todas las transacciones se realizan en batch para mayor eficiencia.

BaseContext: Utilizada por los contratos para el manejo de transacciones.

ForeingKey: Clase concreta que representa una referencia de un estado a otro. Se utiliza como atributo de un estado y contiene la clave y clase del atributo a referenciar. Las relaciones son bidireccionales y se representan como una lista de referencias desde el estado referenciado.

En Kratia cada contrato contempla la gestión de un objeto del dominio junto con las funcionalidades particulares requeridas para ese objeto y múltiples contratos relacionados se incluyen en la misma chaincode[13]. Cada objeto del dominio se representa con una clase que hereda de "State".

Para ilustrar el uso del framework se muestra la chaincode AdminChaincode, la cual contiene dos contratos del proyecto cuyos modelos se encuentran relacionados (figura 4).

La relación "use" en la figura 3 y figura 4 representa una referencia a una clase y permite realizar una evaluación perezosa[14] para la validación dinámica de tipos e instanciación de objetos.

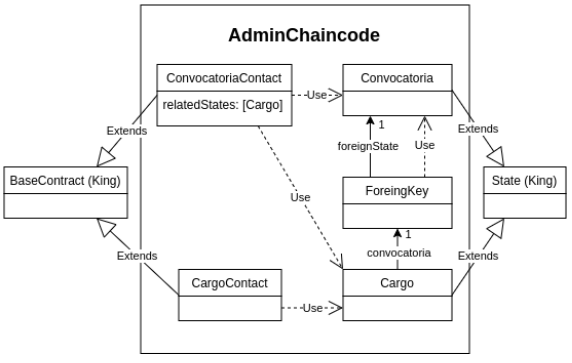


Figura 4.

Discusión

El mundo se encuentra en constante avance y crecimiento, las nuevas tecnologías han cambiado casi todos los aspectos de nuestra vida, sin embargo, la forma en que funcionan los sistemas democráticos y en

particular los sistemas de votación no han evolucionado al mismo ritmo. Sistemas seguros de votación electrónica podrían permitir a futuro una mayor participación democrática de los integrantes de todo tipo de organizaciones y de las personas en sus gobiernos. Además, esta tecnología podría desempeñar un papel fundamental en los gobiernos democráticos que emplean formas de democracia directa[15] y democracia semidirecta[16], abaratando los costos y limitaciones físicas de los sistemas actuales y facilitando los medios para la participación colectiva en la toma de decisiones.

Desde un punto de vista económico y considerando la creciente demanda de sistemas para la toma de decisión empresarial con foco en los aspectos de seguridad y el surgimiento de Blockchain como una tecnología aplicable a negocios, Kratia se encuentra en un nicho que no ha sido explotado hasta el momento. Junto a otros productos de software similares, Kratia se destaca por utilizar una red de Blockchain permisionada, que solo permite a entidades registradas participar de la misma y a su vez estas cuentan con una entidad certificadora que genera las credenciales para que sus usuarios interactúen con el sistema. Otra ventaja es la modularidad que presenta a la hora de adaptarse a distintos dominios, ya que la red es altamente configurable y el framework diseñado permite crear las entidades de negocio con facilidad, de esta forma ofrece una gran flexibilidad para satisfacer requerimientos específicos de distintos clientes.

Conclusión

Finalizado el desarrollo de Kratia se obtiene un sistema de votación electrónico que permite a sus usuarios participar del proceso de elección de sus representantes a través de una plataforma web intuitiva y segura. Gracias al desarrollo del framework King el equipo cuenta con una herramienta para la implementación futura de nuevas reglas de negocio que se pudieran identificar. Cabe destacar que, al utilizar la metodología elegida de Scrum, al final de cada Sprint se completaron entregables de valor para los stakeholders del proyecto. Por último, las tareas de investigación realizadas permitieron al grupo aprender sobre la tecnología de Blockchain, herramientas como Hyperledger Fabric para implementarlas, diseño de sistemas web y API con React y Express y las mejores prácticas de gestión de proyectos a través de Jira y las guías pautadas por el PMI.

Agradecimientos

A nuestros familiares, amigos, y a los docentes que nos acompañaron a lo largo de la carrera y durante este proyecto.

Referencias

[1]https://pmi.org.ar/
[2]https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Spanish-European.pdf
[3]https://www.atlassian.com/es/software/jira
[4]https://es.wikipedia.org/wiki/Desarrollo_guiado_por_pruebas
[5]https://hyperledger-fabric.readthedocs.io/en/release-2.2/
[6]https://www.ibm.com/ar-es/topics/what-is-blockchain
[7]https://docs.docker.com/
[8]https://es.reactjs.org/docs/getting-started.html
[9]https://expressjs.com/es/
[10]https://blockchain-explorer.readthedocs.io/en/master/introduction.html
[11]https://es.wikipedia.org/wiki/Funci%C3%B3n_hash
[12]https://www.djangoproject.com/
[13]https://hyperledger-fabric.readthedocs.io/en/v1.0.0-beta/chaincode.html
[14]https://es.wikipedia.org/wiki/Evaluaci%C3%B3n_perezosa
[15]https://es.wikipedia.org/wiki/Democracia_directa
[16]https://es.wikipedia.org/wiki/Democracia_semidirecta

Datos de Contacto:

UTN-FRC Ingeniería en Sistemas de la Información
Morardo, Diego - diegomorardo@gmail.com
Wismer, Axel - axelwismer@gmail.com
Bossio, Mateo - mateofbossio@gmail.com

PLANILLA PARA CATALOGAR EL PROYECTO FINAL

AÑO	2021	CURSO Y NRO. DE GRUPO	5K3 - G 11

NOMBRE DEL SISTEMA / PROYECTO
KRATIA
CATEGORÍA
Producto

HERRAMIENTAS Y TECNOLOGÍAS UTILIZADAS

ÁMBITO DE APLICACIÓN	NOMBRE Y VERSIÓN
ENTORNO DE DESARROLLO	• Visual Studio Code 1.59
REPOSITORIOS Y VERSIONADO	• Github
PROGRAMACIÓN	• Javascript ES6 • HTML 5 • CSS 3
BASE DE DATOS	• Blockchain / Couch DB 3.1.1
COMUNICACIÓN INTERNA	• Google Meet • Correo • Discord • Whatsapp
FRAMEWORKS	• NodeJS 14.17.3 • Hyperledger Fabric 2.3.2 • React 17.0.2 • Express 4.17.1 • Fabric Explorer 1.1.8
PRUEBAS DE SISTEMA	• Mocha 5.2.0 • Chai 4.1.2 • ESlint 4.19.1
GESTIÓN DEL PROYECTO	• Jira
DOCUMENTACIÓN	• Google Drive
MODELOS	• diagrams.net • Canva

	• **Google Draws**
DESPLIEGUE	• **Docker** • **Shellscript**