

DAPP (APLICACIÓN DESCENTRALIZADA) EN ETHEREUM TESTNETS PARA EL REGISTRO DE DATOS GENERADOS EN UN ENTORNO DE INTERNET DE LAS COSAS (IOT)

DAPP (DECENTRALIZED APPLICATION) ON ETHEREUM TESTNETS FOR LOGGING DATA GENERATED IN AN INTERNET OF THINGS (IOT) ENVIRONMENT.

Guadalupe-Gallardo, Kevin Luciano ^a, Soto-Ortiz, Saul Isai ^a, Martínez-Calva, Citlali Azucena ^a

^aTecnológico Nacional de México/ ITS del Occidente del Estado de Hidalgo, Tecnologías de la información y comunicaciones., Mixquiahuala de Juárez, México. 19011516@itsoeh.edu.mx*, ssoto@itsoeh.edu.mx, cmartinez@itsoeh.edu.mx

RESUMEN. En este proyecto se ha llevado a cabo el desarrollo de una aplicación descentralizada o "DAPP" por sus siglas en inglés, junto con un prototipo orientado al uso del Internet de las Cosas usando el sensor DHT11, los cuales trabajan juntos para almacenar datos en una blockchain proporcionada por Ganache, de forma que los datos sean accesibles, inmutables y seguros, ya que los modelos existentes de Internet de las Cosas (IoT) tienen una pobre seguridad informática lo cual hace que sus datos se vean comprometidos, afectando directamente la integridad, disponibilidad y confidencialidad, por lo que se pretende acabar con ese problema, además se busca ser independiente de un sistema centralizado. En primer lugar, se ha realizado un estudio sobre las diferentes tecnologías Blockchain aplicables a entornos IoT y los diferentes dispositivos y placas de desarrollo. La interfaz de usuario generado en React, permite visualizar los datos recibidos por el prototipo IoT, los cuales interactúan con un contrato inteligente desplegado en una blockchain, para la comunicación e interacción con la red de la cadena de bloques. Finalmente, se implementó en la tarjeta de desarrollo ESP32 y un sensor de temperatura y humedad DHT11 junto con el software que se encargaría de la emisión de datos directamente con el Blockchain. Durante las pruebas del proyecto se demostró la integración entre ambas tecnologías, la de Internet de las Cosas y Blockchain, lo cual mantuvo la integridad de los datos recolectados por los sensores al ser registrados en el Blockchain, lo cual abrió la oportunidad para dar confiabilidad a soluciones a nivel industrial de Internet de las Cosas.

Palabras clave: Descentralizado, Blockchain, IoT.

ABSTRACT. This project has carried out the development of a decentralized application or "DAPP" for its acronym in English, along with a prototype oriented to the use of the Internet of Things using the DHT11 sensor, which work together to store data in a blockchain provided by Ganache, so that the data are accessible, immutable and secure, since existing Internet of Things (IoT) models have poor IT security which makes their data compromised, directly affecting integrity, availability and confidentiality, so it is intended to end that problem, also it is sought to be independent of a centralized system. Firstly, a study has been carried out on the different Blockchain technologies applicable to IoT environments and the different devices and development boards. The user interface generated in React, allows to visualize the data received by the IoT prototype, which interacts with a smart contract deployed in a blockchain, for communication and interaction with the blockchain network. Finally, it was implemented on the ESP32 development board and a DHT11 temperature and humidity sensor along with the software that would be responsible for the issuance of data directly with the Blockchain. During the project tests, the integration between both technologies, the Internet of Things and Blockchain, was demonstrated, which maintained the integrity of the data collected by the sensors when registered in the Blockchain, which opened the opportunity to give reliability to solutions at industrial level Internet of Things.

Key words: Decentralized, Blockchain, IoT.

INTRODUCCIÓN

Las cadenas de bloques están cada vez más integradas en la sociedad. Su gran potencial y su revolucionaria tecnología se abren cada vez más a la gente. Su modelo seguro, descentralizado y público le permite operar con independencia de las autoridades bancarias u otras instituciones. Una de las criptomonedas con mayor potencial que utiliza la

tecnología blockchain es Ethereum. El objetivo de Ethereum es crear una blockchain programable que cambie el modelo de internet, donde los programas almacenados en ella sirvan de columna vertebral para futuras aplicaciones descentralizadas (DAPS).

Estos programas se conocen como contratos inteligentes. La existencia de contratos inteligentes

permite la autoejecución de tareas que responden a una condición preprogramada en el contrato, es decir, en un acuerdo previamente registrado entre partes donde se cumpla una condición existente en el contrato, se ejecutará la cláusula correspondiente a dicha condición.

La Internet de los objetos (IoT) implica la creciente prevalencia de objetos y entidades con identificadores únicos y la capacidad de transferir automáticamente datos a través de una red. Gran parte del aumento de la comunicación del IoT procede de los dispositivos informáticos integrados y los sistemas de sensores utilizados en las comunicaciones industriales, las redes de energía inteligente, la automatización de viviendas y edificios, la comunicación entre vehículos y los dispositivos informáticos portátiles.¹

Esta propuesta de implementación corresponde al desarrollo tecnológico para resolver los principales problemas de seguridad a los que se enfrentan los dispositivos IoT a la hora de enviar información a servidores centralizados, para ello se pretende utilizar la red Ethereum Blockchain y una aplicación descentralizada para la recepción de datos y su almacenamiento en un servidor descentralizado, con el fin de garantizar la inmutabilidad, seguridad y accesibilidad.

METODOLOGÍA

1. Identificación de requerimientos

El proyecto tiene como objetivo el registro de datos generados en un entorno de Internet de las cosas, por lo cual se necesita dispositivos basados en IoT tales como las placas de desarrollo Esp32, el microordenador Raspberry. Destacando que estos componentes son de suma necesidad, ya que se busca asegurar la seguridad de la información y sus tres pilares fundamentales, integridad, confidencialidad y disponibilidad, los cuales se aseguran a través de la seguridad proporcionada por blockchain y el protocolo MQTT; recordando que necesitaremos una computadora con una capacidad de procesamiento suficiente para llevar a cabo la instalación de los programas necesarios tales como, Arduino IDE, Python, Visual Studio, Ganache y Truffle entre otros, los cuales serán necesarios para realizar el desarrollo de nuestro dispositivo².

2. Herramientas de diseño

Primeramente, todo el desarrollo e implementación se realizó en el sistema operativo Microsoft Windows 11, fue necesario contar con un sistema completamente limpio y dedicado para la instalación de herramientas, de esta manera se prevendrá cualquier incompatibilidad entre paquetes de instalación. Para la creación del contrato inteligente se empleó la herramienta Truffle, y para eso es necesario instalarla por medio de un comando npm asumiendo que ya se cuenta con el entorno de desarrollo Node.js instalado en el sistema operativo, dicha herramienta requiere que el sistema cuente con el lenguaje de programación Python y C++ desde visual studio correctamente instalado y configurado, de igual manera se realizó una interfaz de usuario sencilla con HTML Y CSS, se utilizaron las librerías de Bootstrap para dar diseño a los componentes y textos, de igual manera se utilizó JavaScript nativo para el desarrollo del backend de la interfaz³.

3. Implementación del prototipo

Para el desarrollo del Smart Contract se debe de crear un espacio de trabajo con Truffle en una ruta específica, en este caso se usó el escritorio, para ello se ejecuta el siguiente comando, "truffle init" seguido del nombre de la carpeta donde se guardará los archivos de Truffle y el Smart Contract⁴.

Se generará una carpeta con nombre proporcionado y dentro de ella tres carpetas y un archivo "truffle-config.js", para comenzar en la carpeta contracts generaremos dentro de ella nuestro contrato inteligente. A continuación, se muestra la estructura del contrato inteligente haciendo énfasis en las partes más destacadas: Es importante que al inicio del código se especifique con qué versión de Solidity se va a compilar nuestro Smart Contract, en este caso se utilizó la versión 0.8.6⁵. También necesariamente se debe de disponer de una licencia que se agrega al inicio del código en forma de comentario.

Después especificamos las variables que se van a tomar en cuenta para guardar los datos del sensor y la temperatura, principalmente se especifica la variable que guardará la temperatura en una variable de tipo "uint" dado que va a ser un número, seguidamente estructuramos las variables que conformarán los datos del sensor. La variable id de

tipo uint256 necesaria para cualquier registro. La variable NoSerie de tipo String tomando en cuenta el número de serie perteneciente al sensor combinando letras y números. La variable Name de tipo string haciendo referencia al nombre del sensor.

La variable “active” de tipo bool encargada de cambiar entre “true o false” de acuerdo al estado en el que se encuentra el sensor (activo/Inactivo), y por último la variable “addAT” de tipo uint256 encargada de guardar la fecha en la que se agregó el sensor.

Es importante conocer completamente la función encargada de agregar un sensor de acuerdo a la estructura descrita anteriormente, recordando que es esencial conocer estos datos para poder desarrollar una estructura específica a los datos del sensor en el backend de la interfaz. La función se muestra en el siguiente fragmento de código:

```
function AddSensor(string memory _NoSerie, string memory _Name) public
{
    SensorCounter++;
    sensors[SensorCounter] = Sensor(
        SensorCounter,
        _NoSerie,
        _Name,
        true,
        block.timestamp
    );
    emit SensorAdd(
        SensorCounter,
        _NoSerie,
        _Name,
        true,
        block.timestamp
    );
}
```

Figura 1. Fragmento de código.

Esta función permite agregar un sensor a la red Blockchain y emitir un evento cuando el sensor es agregado, gracias a esto se puede consumir los datos de un sensor directamente en la interfaz de la Dapp. Por otro lado, se encuentran las funciones encargadas de consultar el estado del sensor y para la inserción y consulta del dato de la temperatura.

Una vez creado dicho contrato ya se tendrá el contrato inteligente terminado, solo hace falta desplegarlo, pero antes es necesario contar con Ether, ya que el despliegue es una transacción dentro de la red de Blockchain y se necesita pagar con Ether, en este caso se hizo uso de una red proporcionada por Ganache donde se obtendrá Ether de prueba para testear las transacciones⁶.

4. Pruebas y mejoras del prototipo

La programación de la placa Esp32 consta en dos partes, la primera consiste en programar el sensor de temperatura para que lea el valor de la temperatura actual y guarde el dato en una variable. La segunda es incorporar el protocolo de comunicación MQTT para controlar la salida del dato de la temperatura y publicar el mensaje en el servidor bróker⁷.

Como tal no existen tecnologías similares al proyecto propuesto, lo más cercano a lo propuesto es la plataforma Stacks blockchain, la cual es una plataforma de código abierto para permitir contratos inteligentes, DeFi, NFT y aplicaciones para Bitcoin.

En primer lugar, el sensor es digital y no hay que realizar cálculos para transformar voltaje en medidas, cosa que sí hay que realizar en los analógicos y aunque las medidas no son muy precisas, no es algo que cause preocupación para el prototipado.

En las diversas pruebas que se hicieron con el prototipo se reflejó la gran precisión con la que se cuenta debido a que observamos que ante cualquier variación del sensor lo refleja en tiempo real.

RESULTADOS Y DISCUSIÓN

En este paso mostraremos los resultados obtenidos al seleccionar un sensor, para evidenciar el correcto funcionamiento se mostrará un ejemplo seleccionando el sensor con el nombre “Sensor DS18B20 DALLAS”.

Después de dar en el botón “Seleccionar” automáticamente la página se actualiza con los datos del sensor.

Como podemos observar en la tabla de información muestra los datos del sensor seleccionado, y de igual manera, en el apartado “Datos recopilados del sensor” muestra el dato que se está mandando, como podemos observar en la siguiente tabla.



Figura 2. Información del sensor

Ahora se mostrará los resultados obtenidos con el prototipo operando de acuerdo a lo establecido en la programación del mismo, En la Figura 3 se muestra el prototipo conectado a una computadora para posteriormente mostrar los registros generados.

En el monitor serie de la Esp32 muestra los siguientes registros haciendo referencia de esta conectado a la red wifi y al servidor MQTT Broker, en la siguiente imagen podemos observar los registros del prototipo.

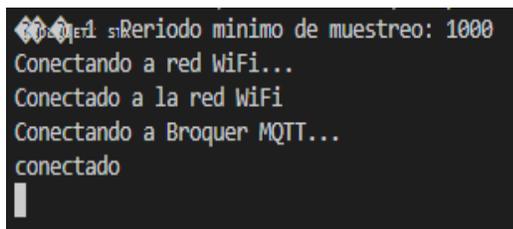


Figura 3. Registros prototipo

Para asegurarse de que se está registrando la temperatura correspondiente al sensor, accedemos al Dashboard del servidor MQTT Broker, los resultados se muestran a continuación.

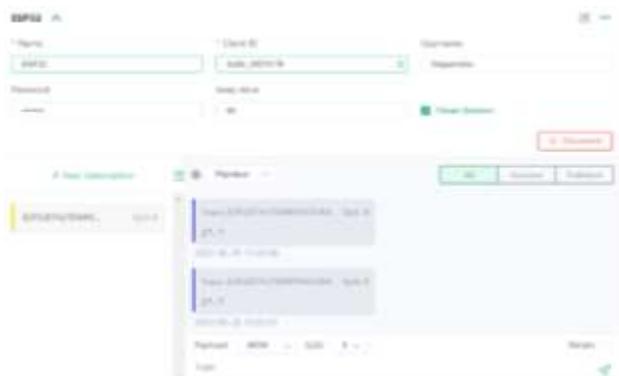


Figura 4. Resultados MQTT del sensor

Actualmente, el sensor está monitoreando una temperatura de 27.7 grados centígrados, misma que se muestra en la tabla “Datos recopilados del Sensor” de la interfaz de usuario de la Dapp. Finalmente, se concluye que el prototipo hace lo establecido de acuerdo a los objetivos establecidos en el proyecto.

CONCLUSIONES

Blockchain proporciona una forma bastante segura de almacenar datos y realizar transacciones. Se prevé que el mercado mundial de soluciones relacionadas con Blockchain aumente de 1500 millones a 15900 millones para 2023. Además, debido a que tecnología como internet de las cosas (IoT) continúa expandiendo su uso y aceptación, adopta la tecnología Blockchain.

Según las previsiones, para 2030, se utilizarán aproximadamente cincuenta mil millones de dispositivos IoT en todo el mundo. Esto es una noticia positiva para la tecnología Blockchain y una mayor demanda de desarrolladores relacionados con Blockchain.

El desarrollo de este proyecto es un gran inicio para introducirse a estas nuevas tecnologías, y se continuará con su desarrollo aportando nuevas mejoras y generando nuevos proyectos con el mismo enfoque. Finalmente, podemos concluir que el desarrollo de esta aplicación descentralizada, cumple con los objetivos propuestos para el desarrollo de este proyecto, los cuales fueron la creación de una web app, la cual fue creada exitosamente mediante ganache, destacando que se logró exitosamente el envío de datos de manera segura en un entorno IoT con Blockchain. De igual manera el despliegue del prototipo fue lo que se esperaba, en proyectos futuros se espera crear un PCB para integrar el mismo microcontrolador utilizado (ESP32) y que tenga la facilidad de conectar distintos tipos de sensores a una escala mayor que la obtenida.

AGRADECIMIENTOS Y/O RECONOCIMIENTOS

Agradecemos al Programa Educativo de Ingeniería en Tecnologías de la Información y Comunicaciones del Instituto Tecnológico Superior del Occidente del Estado de Hidalgo por permitir la participación en su línea de investigación de Redes y Telecomunicaciones a Organizaciones por el

desarrollo de nuestro proyecto, así como a los compañeros que forman parte del servicio social y residencia que apoyaron con sugerencias para la mejora del prototipo.

REFERENCIAS

1. Equisoft, "La cadena de bloques (blockchain). Una tecnología disruptiva con el poder de revolucionar el sector financiero." Marzo 2017.
2. M. Echebarria, "Contratos electronicos autoejecutables (smart contract) y pagos con tecnología blockchain" Revista de estudios europeos, no 70, pp. 69-97, 2017.
3. B. Asolo, "Blockchain Soft Fork & Hard Fork Explained", Septiembre 2017. [En línea]. Disponible en: <https://www.mycryptopedia.com/hard-fork-soft-fork-explained>
4. Arash, "Go-Ethereum GETH Documentation", Diciembre 2018. [En línea]. Disponible en: <https://github.com/ethereum/go-ethereum/>
5. Patel, Keyur., y M Patel, Sunil. (Mayo de 2016). Internet of Things-IOT: Definition, Characteristics, Architecture, Enabling Technologies, Application & Future Challenges. Maharaja Sayajirao University of Baroda. Research Gate. https://www.researchgate.net/publication/330425585_Internet_of_Things-IOT_Definition_Characteristics_Architecture_Enabling_Technologies_Application_Future_Challenges
6. Chandrashekhar, Kavya. (20 de septiembre de 2016). Internet of Things (IoT) Characteristics. LinkedIn. <https://www.linkedin.com/pulse/internet-things-iot-characteristics-kavyashree-g-c/>
7. Manuel José Gonçalves. ¿Qué es Angular y para qué sirve? - Blog de Hiberus Tecnología. Blog de Hiberus Tecnología. Published October 13, 2021. Accessed May 31, 2022. <https://www.hiberus.com/crecemos-contigo/que-es-angular-y-para-que-sirve/>