Nodo experimental de registro e inmutabilidad de variables ambientales

Ciro Edgardo Romero[1], **Alejandro Matías Elustondo**[2], **Reyna Karine Der Boghosian**[3], **Moisés Carlos Fontela**[4]

[1234] C&S Informática S.A. Departamento de Investigación, Desarrollo e Innovación

[4] Universidad de Buenos Aires. Facultad de Ingeniería.

[3] Universidad de la República Oriental del Uruguay. Facultad de Ingeniería

[2] Co Fundador de XDK2MAM

Buenos Aires, Argentina

{cromero, aelustondo, rderboghosian, cfontela}@cys.com.ar

Resumen. En el nuevo contexto de la industria 4.0, o cuarta revolución industrial, los datos se convirtieron en un activo de gran valor para las empresas. Toda la información recopilada y guardada por los diferentes elementos de un sistema informático son un valor en sí mismo.

En el mercado de hoy, el valor de la información está ligada a su procedencia. Generar los datos de manera segura sirve para dar información detallada sobre los pasos clave en la producción de un producto. El problema de tener tantos agentes funcionando en simultáneo es la confianza que se debe tener sobre los componentes y las personas involucradas. Utilizando tecnología blockchain se logra fiabilidad aumentando la confianza del consumidor de dichos datos, ya que no podrían ser modificados por nadie. El objetivo de este artículo es mostrar el resultado de la implementación de un sistema capaz de recolectar variables ambientales, procesarlas, enviarlas a una base datos y certificar que la información es confiable e inmodificable.

Palabras claves: Micropython, Blockchain, IoT, Internet de las cosas, BFA, Smart Contract.

1 INTRODUCCIÓN

La utilización de sensores es indispensable en la automatización de industrias de procesos y manufacturados, incluida la robótica, la ingeniería experimental, el control ambiental e incluso los equipos de gestión de datos, alejados de las aplicaciones industriales [1]. Un sistema eficiente de manufactura requiere una base de datos que contenga la información actualizada, detallada y precisa para que los individuos en la organización, o el propio sistema los recuperen o modifiquen según sea necesario. Un sistema de adquisición de datos recaba los datos en forma automática y provee información de las variables que cambien en el sistema. Los componentes del sistema de adquisición de datos incluyen microprocesadores, transductores y convertidores analógicos a digitales (ADC). Los sistemas de adquisición de datos también tienen la capacidad de analizar los datos y transferirlos a otras computadoras para su análisis estadístico, presentación y predicción de demanda de productos [2].

Una blockchain es una estructura de datos en la que la información contenida se agrupa en conjuntos, llamados "bloques" a los que se les añade meta información relativa al bloque anterior, produciendo así una vinculación en una línea temporal, de manera que, gracias a técnicas criptográficas, la información contenida en un bloque solo puede ser editada modificando todos los bloques posteriores [3]. Esta propiedad permite que pueda ser integrada en una aplicación de tal manera que la estructura de datos contenga un histórico irrefutable de información [4].

En el área de Investigación, Desarrollo e Innovación de C&S hemos decidido desarrollar un dispositivo de sensado de presión y temperatura, intercomunicado con una blockchain semi permisionada de acceso público, a través de un servicio de integración, para que pueda ser instalado en un ambiente acondicionado de tal forma que se pueda realizar un registro constante de los cambios de las variables ambientales, se puede establecer un control confiable dentro del ambiente. Al poseer información que se considera inmutada, se puede garantizar que las condiciones siempre se mantuvieron óptimas; y en el caso de un cambio o de una alteración, se deja constancia de cuáles fueron y en qué momento.

El artículo se compone por secciones descriptivas del desarrollo de cada elemento, integración de los diferentes componentes, pruebas de funcionamiento y conclusiones. La sección de Arquitectura del Sistema menciona los elementos que componen el mismo describiendo su función particular. La sección de implementación de hardware detalla el desarrollo del dispositivo de captura de datos ambientales, describiendo cada uno de sus elementos y las conexiones internas para su funcionamiento, así como cada una de las herramientas usadas durante su desarrollo. La sección de implementación de software, se divide en el software del dispositivo (firmware) y del servicio de integración, describiendo las tecnologías de lenguaje utilizados. En la sección de ejecución del programa se explica el funcionamiento íntegro del sistema, detallando el rol de cada elemento y explicando el paso a paso del flujo normal de funcionamiento. En las secciones de caso de prueba, conclusión y trabajo futuro vemos un caso hipotético donde se pudo ver el funcionamiento real del sistema, recopilado de información y análisis de datos para explicar las deducciones obtenidas. Se finaliza detallando los pasos a seguir basado en toda la evidencia recolectada y expuesta.

2 ANTECEDENTES

Actualmente, se utiliza el Internet de las Cosas para monitorear partes que componen determinados procesos, conectando los sensores y los dispositivos a un sistema de control global del proceso de fabricación, enriqueciendo la información integrada [5]. Hay una serie de productos y servicios en el mercado de hoy cuyo valor está ligado, en mayor o menor medida, a su procedencia. Utilizando un enfoque de bloques de código de programación, vinculados entre sí de manera segura, se puede rastrear y dar información detallada sobre los pasos clave en la producción de un producto. Los registradores actúan como gestores de confianza para verificar la identidad y las credenciales de los otros participantes nombrados [5]. Los posibles riesgos de seguridad que traen aparejados la interconexión de los dispositivos son diversos y cambian a una velocidad constante, lo que hace que el enfoque tradicional de centrar los componentes más importantes de un sistema y protegerlo, ya no sea prudente [6]. El problema de tener tantos agentes funcionando en simultáneo es la confianza que se debe tener, no solo sobre los componentes, sino también sobre las personas involucradas; lo que serían intermediarios dentro del proceso. Como la infraestructura carece de la seguridad suficiente, se tiene que tratar con los intermediarios [7] confiando plenamente en que los individuos, intermediarios u otras entidades actúen con integridad.

Por lo expuesto anteriormente, desde el área de Investigación, Desarrollo e Innovación concebimos la idea de tener un Nodo de registro e inmutabilidad de variables ambientales, sin intermediarios más que el sistema mismo y los agentes que lo componen. Al almacenar los datos utilizando tecnología como blockchain, aseguramos la inmutabilidad de los mismos asi como también su integridad; logrando confianza y garantías sobre la información recolectada.

3 ARQUITECTURA DEL SISTEMA

El sistema cuenta con tres partes para su funcionamiento.

- **Code Jelly**: dispositivo recolector de datos ambientales que cuenta con un módulo sensor, módulo FTDI, conexión wifi y batería autónoma.

- **Single Board Computer (SBC)**: pequeña computadora con un servicio de integración entre el dispositivo de recolección de variables y la base de datos [8].

- **Blockchain Federal Argentina (BFA)**: Es una plataforma descentralizada que funciona bajo Prueba de Trabajo y permite a cualquier desarrollador crear y publicar aplicaciones distribuidas para ejecutar. Blockchain Federal Argentina toma el software de Ethereum, utilizando Prueba de Autoridad, sin criptomoneda asociada [9].

En el desarrollo del sistema se destaca el uso de herramientas *Open Source*, tanto en Hardware como en Software. Las herramientas *Open Source*, o de código abierto, están diseñadas de manera que sea accesible al público: todos pueden ver, modificar y distribuir el código de la forma que consideren conveniente. Además, suele ser más económico y flexible, ya que las encargadas de su desarrollo son las comunidades y no un solo autor o una sola empresa [10]. Por esta practicidad y conveniencia, decidimos utilizar varias herramientas las cuales se describen a lo largo de todo el presente artículo.

Se implementó una red privada provista por la empresa C&S como contexto en el cual se probaron los diferentes casos de uso asociados al proyecto. Restringiendo al público el acceso a la red, como requisito mínimo de seguridad en esta primera etapa.

4 IMPLEMENTACIÓN DEL HARDWARE

La unidad de control está compuesta por el microcontrolador ESP32, de la empresa Espressif [11], una placa de desarrollo fácil de obtener en el país, dado su bajo costo y versatilidad. La misma placa tiene incorporado una antena WIFI y conexión Bluetooth, lo que hace un entorno de desarrollo integral muy versátil.

El desarrollo de los esquemas donde se reflejan la integración la conexión de los módulos, con el microprocesador y los misceláneos del proyecto, se empleó el software de diseño EasyEDA, una herramienta de uso gratuito que permite a diseñar, simular, compartir y discutir esquemas, simulaciones y placas de circuito impreso [12]. Con la misma herramienta, se realizó el desarrollo de la PCB [13] como parte de un plan de mejora para optimizar la implementación del dispositivo.

La conexión entre el microprocesador y sus componentes se realizó a través de un cableado simple, punto a punto, siguiendo el mapa de pines de la placa DOIT Esp32 DevKit v1 [14]. Se utiliza cable bus para conectar la placa con el módulo sensor de presión y temperatura BMP180

[15]. El Módulo FTDI [16] se conecta punto a punto con la placa de desarrollo para establecer una comunicación UART donde el usuario puede configurar la red a la que se va a conectar; así como desconectarse de la red conectada para cambiarla. Se utiliza una batería de 9V, pasando por un interruptor NA, para alimentar el dispositivo, aprovechando el regulador de voltaje LDO incorporado en la placa de desarrollo, desde el pin de alimentación externa. Para visualizar los estados de ejecución del programa, se agrega un led RGB, conectado directamente a la placa de desarrollo con 3 cables resistores más un cable común de retorno de 2 mm. En la figura 1 se muestra la relación entre los diferentes módulos del dispositivo colector de variables.

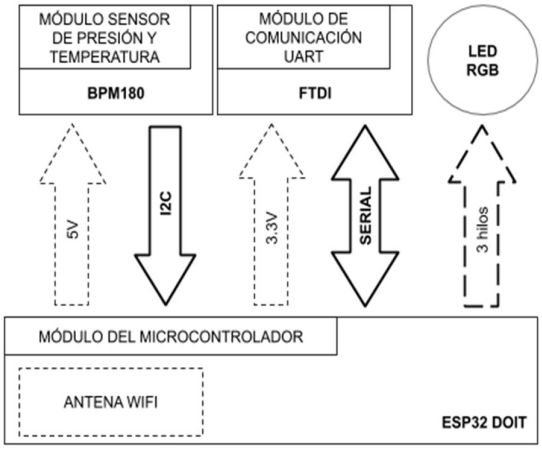

Figura 1: Módulos del dispositivo

Todo el montaje se realizó sobre un gabinete impreso utilizando tecnología de impresión 3D, diseñado con la plataforma Tinkercad; un programa gratuito de modelado 3D [17].

5 IMPLEMENTACIÓN DEL SOFTWARE

5.1 Desarrollo de Firmware

Se utilizó como como lenguaje de programación MicroPython; una implementación software del lenguaje de programación Python 3, escrita en C, y que está optimizada para poder ejecutarse en un microcontrolador. MicroPython es un compilador completo del lenguaje Python y un motor e intérprete en tiempo de ejecución, que funciona en el hardware del microcontrolador [18]. La gran ventaja de este lenguaje es su capacidad de compilación y ejecución en tiempo real, lo que permite hacer un trabajo más ágil cuando se utilizan metodologías de desarrollo colaborativo; cargando el proyecto en un software de control de versiones y ejecutando en el intérprete instalado en el microcontrolador.

5.2 Desarrollo de servicio de Integración

El servicio de integración está compuesto por una API, un Nodo de Ethereum (Geth) y el de un smart contract, o contrato inteligente como sería su traducción literal. Un contrato inteligente es

esencialmente un acuerdo escrito en código de programación y entregado por un blockchain. [19].

Estos smart contract son la mejor opción para interactuar, agrupar y almacenar un conjunto de datos en redes descentralizadas. El desarrollo del smat contract es de naturaleza genérica lo que permite que pueda ser desplegado en cualquier red de Ethereum, es por esto, que puede ser desplegado en la red Blockchain Federal Argentina.

Las tecnologías empleadas fueron NodeJS [20] para la API aprovechando la facilidad de acceso a la documentación, Geth [21] para el Nodo por ser la implementación core de Ethereum por lo cual recibe constantes actualizaciones y mantenimiento al ser un proyecto open source y finalmente Solidity [22] para la programación del smart contract, lenguaje de programación por defecto de Ethereum. En la figura 2 se muestra la relación entre los diferentes elementos del sistema.

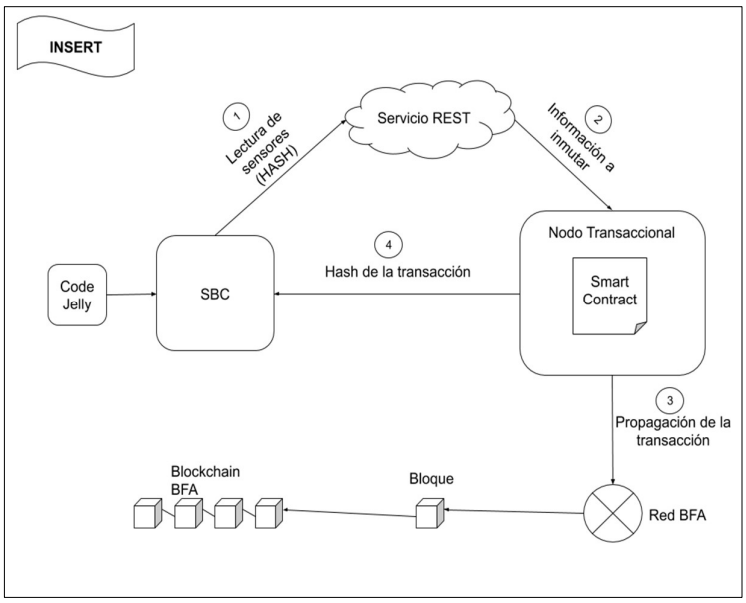

Figura 2: ciclo completo de inmutabilidad

La pieza fundamental del servicio de integración es la lógica que contiene el smart contract, este se encarga de agrupar todos los hashes representativos a cada medición del dispositivo para que luego, de ser necesario, se pueda verificar la inmutabilidad de dichas mediciones.

6 COMPORTAMIENTO DEL SISTEMA

Cuando inicia el dispositivo el gestor de arranque desde la ROM inicializa la configuración y los servicios de la placa [23]. En su primera ejecución, el dispositivo activa la antena WIFI y se configuró para funcionar en modo AP [24] y transmite por UART el mensaje al usuario para iniciar la configuración de conexión a internet. No se puede avanzar dentro de la ejecución del

programa hasta que no se haya establecido una conexión efectiva. El usuario debe tener el dispositivo conectado a una computadora u otro dispositivo que soporte la comunicación serie. La red configurada en el equipo, se guarda en memoria FLASH interna [23] para que el equipo se vuelva a conectar cuando vuelva a iniciarse; con el mismo procedimiento se puede reiniciar la configuración para cambiar la red.

Con el dispositivo conectado a internet, se realiza una petición a un servidor público NTP [25] para configurar la hora del RTC. Se inicializa la comunicación I2C de la placa de desarrollo y se establece la comunicación con el módulo sensor de presión y temperatura BMP180.

Se establece la comunicación con el servicio de interconexión con BFA, que se encuentra en la SBC, y se despliega el smart contract mediante una petición POST a un endpoint que expone la API (alojada en la SBC), dicha petición obtendrá una respuesta con la dirección del smart contract desplegado. Después se hace la solicitud de datos al módulo sensor; se leen los datos de temperatura, presión y altitud del momento. Se manipula la medición para darle formato JSON, a lo que se le suma los datos de UID (identificador único de dispositivo), hora y fecha de la medición. Una vez construido el JSON se procede a la encriptación del mismo por medio de una función hash (SHA256), en este momento se efectúa otra petición POST a otro endpoint de la API y en conjunto con la dirección del smart contract se envía el hash de la medición para efectuar finalmente la inmutabilidad del dato. Se utiliza el colector de basura [26] y se vuelve a realizar una medición.

Una vez concretado el código, se realizaron pruebas unitarias para verificar su correcto funcionamiento. Montando los componentes en una *protoboard* se ejecutó el firmware en la ESP32. El programa reconoció el sensor correctamente y realizó la medición de presión y temperatura del momento. Los datos fueron enviados al base de datos y se realizó el proceso de inmutación de datos.

Finalizadas las pruebas se procedió al ensamblando el gabinete, conectado los componentes electrónicos correspondientes en la placa de desarrollo grabada (con el firmware) y colocando la batería necesaria para que funcione de manera autónoma, se realizó el montaje completo y verificación de funcionamiento del prototipo.

7 PRUEBA PILOTO

Basado en la motivación por la cual se desarrolla el proyecto se plantea la situación hipotética de un ambiente estéril y controlado, que deba mantener sus condiciones de presión y temperatura constante. Igualmente, importante, se debe tener registro de las condiciones del ambiente en un periodo determinado, para garantizar que no hubiera adulteraciones; esta situación se puede encontrar en la correcta conservación de medicamentos. Los medicamentos deben ser conservador en un ambiente refrigerado, entre 2ºC y 8ºC [27] en condiciones normales de presión (1013 milibar/760 mm Hg).

7.1 Experimento

Se instaló el dispositivo, ensamblado y con batería suficiente, dentro de una conservadora descartable con gel refrigerante suficiente para bajar la temperatura del recinto a 2ºC. Se mantuvo la conservadora cerrada durante una hora y media, para simular la situación de un medicamento que se encuentra transportado desde una cámara refrigerada a un hospital, dentro de un recipiente controlado. El dispositivo realizó las mediciones de la temperatura de la conservadora y las envío al servicio de integración donde toda la información quedó registrada en una base de datos, al

mismo tiempo que fue inmutada en la BFA. En la figura 3 vemos la evolución de la temperatura registrada.

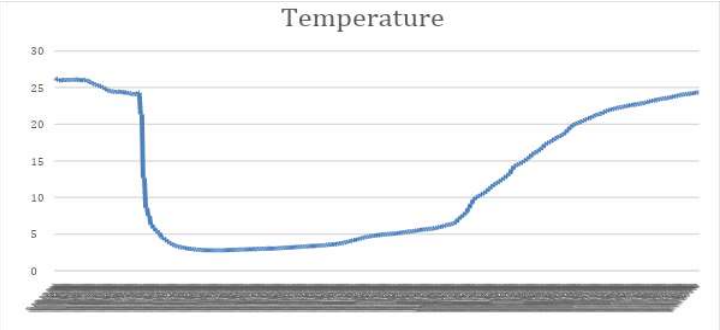

Figura 3: curva de temperatura

7.2 Evidencia

Todos los datos quedaron registrados en la BFA. Como evidencia tomamos como muestra una de las mediciones realizadas:

```
{'lat': '-34.603580', 'date': 1583342918, 'temperature': 6.007388, 'altitude': 12.80066,
'pressure': 101162.8, 'device-id': b'3c71bffe4c54', 'lon': '-58.381615'}
```

Ejecutamos un script desarrollado en python para encriptar la medición.

```
import hashlib
import binascii

medicion = "{'lat': '-34.603580', 'date': 1583342918, 'temperature': 6.007388, 'altitude':
12.80066, 'pressure': 101162.8, 'device-id': b'3c71bffe4c54', 'lon': '-58.381615'}"

def _hash(rawObject):
  objectHash = hashlib.sha256(str(rawObject).encode('utf-8'))
  return binascii.hexlify(objectHash.digest()).decode('utf-8')

hash = _hash(medicion)
print('0x'+_hash(hash))
```

El resultado de la ejecución sería el siguiente:

```
0x1cb80a8c1522757d988e5cb19c852ad86343ab335b2cd68288eeccb4ddaef1d7
```

Ingresamos en la BFA y buscar el hash de la transacción. En la figura 4 se podrá ver el registro con la inmutabilidad de la medición seleccionada, efectuadas en el experimento.

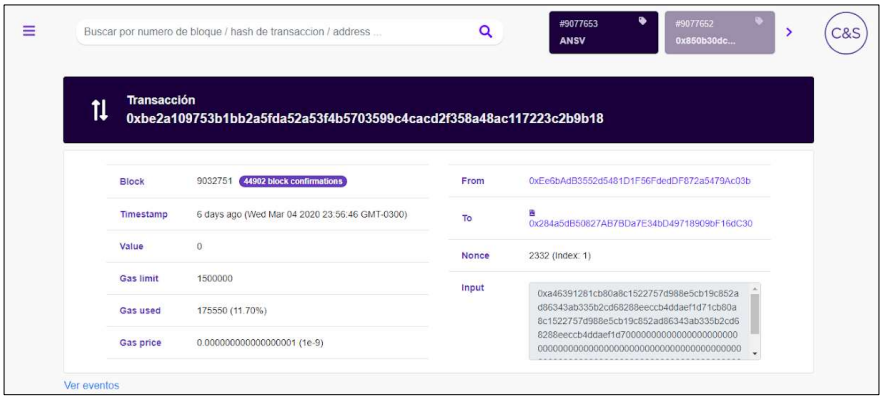

Figura 4: curva de temperatura

Dentro del mismo registro, en la sección de **Ver eventos**, estará el hash inmutado de la medición, que sirve como registro para determinar que no fue modificado. Vemos en la figura 5 que el hash del registro es igual a obtenido en la ejecución del script

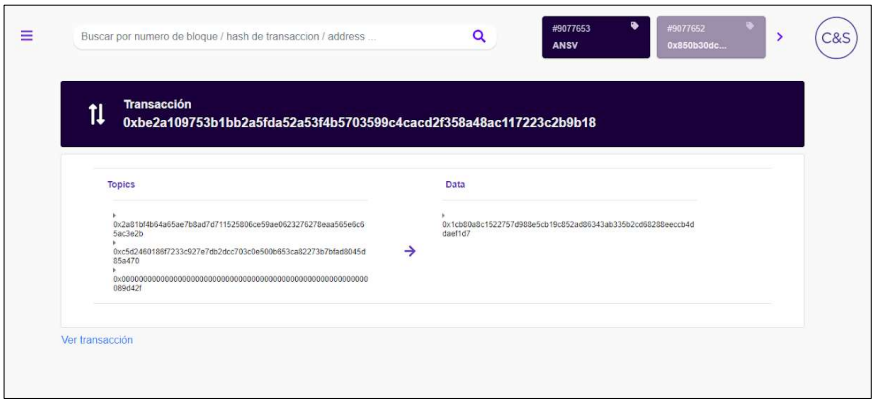

Figura 5: curva de temperatura

El enlace de acceso al registro es el siguiente:

https://bfaexplorer.com.ar/bfa/tx/0xbe2a109753b1bb2a5fda52a53f4b5703599c4cacd2f358a48ac117223c2b9b18

8 CONCLUSIÓN

El sistema completo, con todos sus componentes, podría ser instalado en cualquier ambiente donde sea útil (hospitales, bodegas, humidores, etc.) y combinarlo con otros sistemas para tener un control integral y confiable. Los datos, al quedar registrados en una blockchain poseen la confianza suficiente para garantizar que no fueron modificados por ningún intermediario ajeno al proyecto. No obstante, la naturaleza inherente de los sensores provoca lecturas erróneas que

podrían ser producto de las propias fallas del circuito, ruido eléctrico, envejecimiento del sistema, problemas de conexión [28], hacen necesario agregar más elementos de chequeo dentro del flujo de funcionamiento. La propia necesidad de calibración que tienen los sensores, requiere que se contemple en la misma blockchain un registro donde quede asentado la última calibración y mantenimiento del dispositivo, para sumar a la cadena de confianza el detalle del estado del equipo. Sumado a estas cuestiones técnicas, la necesidad de conectividad obliga al dispositivo a tener acceso a internet para poder realizar el registro constante y en tiempo real.

9 TRABAJO FUTURO

En el área de Investigación, Desarrollo e Innovación se está trabajando con la evidencia recolectada, como cimientos para desarrollar un dispositivo más robusto, dedicado, para realizar mediciones en ambientes menos predecibles. Integrando todos los elementos en un misma PCB, quitando el ruido eléctrico natural de las conexiones cableadas. Así, se puede tener más control para calibrar los sensores y mejorar el firmware. Agregar un módulo para conexión internet remoto, con GSM/GPRS y GPS, para brindarle al equipo conectividad autónoma (sin depender de WIFI) para poder registrar la actividad en tiempo real desde el sitio donde se encuentre.

Para agregar robustez al sistema se incorporaría una capa de seguridad más compleja para contemplar la autenticación de los dispositivos. Cumpliendo normas de seguridad informática.

Si se consigue lograr un código más complementario que realice todas las acciones de procesamiento y comunicación con la blockchain, se puede lograr independizarse de un servicio externo, creando una descentralización mayor.

Como otras sugerencias, se ampliará la cantidad y variedad de sensores. De esta forma, ampliaría el campo de implementaciones del sistema. Se sumaría un sistema de alertas parametrizables que permita tomar acciones tempranas sobre las mediciones realizadas, con suficiente tiempo para reducir riesgos y daños, productos de malos funcionamientos en las instalaciones donde se encuentre el sistema. Se contempla incorporar servicios de análisis de datos provistos por el propio sistema.

REFERENCIAS

1. *Areny, Ramón Pallás. Sensores y Acondicioadores de Señal 4a. Marcombo, 2005. e*
2. *Kalpakjian, Serope, and Steven R. Schmid. Manufactura, ingeniería y tecnología. Pearson educación, 2002.*
3. *Andreas M Antonopoulus "Mastering Mastering Bitcoin: Unlocking Digital Cryptocurrencies". 2nd Edition. Editorial O'Reilly - Chapter 9: The Blockchain. [Online]. Disponible: https://github.com/bitcoinbook/bitcoinbook/blob/develop/ch09.asciidoc AND Bitcoin Whitepaper - Satoshi Nakamoto Avaiable: https://bitcoin.org/bitcoin.pdf (visitado 01-01-2020)*
4. *Nathan Reiff. Blockchain Explained - What is Blockchain? https://www.investopedia.com/terms/b/blockchain.asp (visitado 01-01-2020)*
5. *SOLER, Víctor Gisbert; MOLINA, Ana Isabel Pérez. Blockchain vs ISO 9001: 2015. 3C Tecnologia, 2019, vol. 8, no 2, p. 36. Disponible: http://ojs.3ciencias.com/index.php/3c-tecnologia/article/view/823*
6. *Garrell, Antoni, and Llorenç Guilera. La industria 4.0 en la sociedad digital. Marge books, 2019.*

7. *Tapscott, Don, and Alex Tapscott. "La revolución blockchain." Descubre cómo esta nueva tecnología transformará la economía global. ediciones deusco. séptima edición. recuperado en webdelprofesor. ula. ve/economia/oscared/materias/E_E_Mundial/Economia_ Internacional_Krugman_Obstfeld. pdf (2017).*
8. *Raspberry Pi Foundation. Putting the power of digital making into the hands of people all over the world. Disponible: https://static.raspberrypi.org/files/about/RaspberryPiFoundationStrategy2016-18.pdf (visitado 03-01-2020)*
9. *Presentacion bfa – Blockchaing federal Argentina. Disponible: https://gitlab.bfa.ar/blockchain/docs/wikis/uploads/2f1ea2f0b7a40121dd273d30ef7090ee/Brief_BFA.pdf (visitado 08-01-2020)*
10. *¿Qué es el open source? – RedHat. Disponible: https://www.redhat.com/es/topics/open-source/what-is-open-source (visitado el 05-02-2020)*
11. *ESP32, A Different IoT Power and Performance. Disponible: https://www.espressif.com/en/products/hardware/esp32/overview (visitado 17-01-2020)*
12. *EasyEDA gives makers PCB layout in a browser – www.seeedstudio.com (blog). Disponible: http://www.seeedstudio.com/blog/2014/06/25/easyeda-gives-makers-pcb-layout-in-a-browser/ (visitado 08-01-2020)*
13. *Al Williams. A tale of two browser PCB tools - Hackaday . Disponible: https://hackaday.com/2015/08/21/a-tale-of-two-browser-pcb-tools/ (visitado 08-01-2020)*
14. *DOIT Esp32 DevKit v1. Pin Mapping. Disponible: https://docs.zerynth.com/latest/official/board.zerynth.doit_esp32/docs/index.html*
15. *BMP180. Digital pressure sensor. [Datasheet]. Disponible: https://cdn-shop.adafruit.com/datasheets/BST-BMP180-DS000-09.pdf*
16. *FTDI. [Datasheet]. Disponible: https://www.ftdichip.com/Support/Documents/DataSheets/ICs/DS_FT232R.pdf*
17. *Katie Macdonald. How To Get Started In 3D Printing. Disponible: https://www.popularmechanics.com/technology/gadgets/a19698/get-started-3d-printing/(visitado 17-01-2020)*
18. *Micropython. Disponible: https://micropython.org/ (visitado 17-01-2020)*
19. *Nick Szabo. Formalizing and Securing Relationships on Public Networks. Disponible: https://journals.uic.edu/ojs/index.php/fm/article/view/548/469 (visitado 31-01-2020)*
20. *Aceca de Node.js. Disponible: https://nodejs.org/es/about/ (visitado 03-01-2020)*
21. *Ethereum Whitepaper. Disponible: https://github.com/ethereum/wiki/wiki/white-paper (visitado 01-01-2020)*
22. *[13] Solidity – Ethereum Foundation. Disponible: https://solidity.readthedocs.io/en/latest/# (visitado 31-01-2020)*
23. *DOIT Esp32 DevKit v1. Flash Layout. Disponible: https://docs.zerynth.com/latest/official/board.zerynth.doit_esp32/docs/index.html*
24. *[20] Quick reference for the ESP8266. 4 - Network basic. Disponible: https://docs.micropython.org/en/latest/esp8266/tutorial/network_basics.html*
25. *NTP Pool Project. Disponible: https://www.ntppool.org/es/*
26. *Python standard libraries and micro-libraries - control the garbage collector. Disponible: https://docs.micropython.org/en/latest/library/gc.html*
27. *Best-Practice Guide Pharmaceutical-Chain Temperature Control and Recording. ABB. Disponible: https://library.e.abb.com/public/b0c7522c15decafcc125766d0055530f/Best%20Use%20Guide.pdf*
28. *Juan Calderon, Yamilet Sanchez Montero. Medicion e Instrumentacion industrial. 5ta revisión (2004).*