



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
GRADO EN INGENIERÍA EN TECNOLOGÍAS DE LA TELECOMUNICACIÓN

DESPLIEGUE DE UN SISTEMA DE CADENA DE SUMINISTRO SOBRE UNA INFRAESTRUCTURA HYPERLEDGER

Autor: Esther Vázquez Rodríguez

Director: David Contreras Bárcena

Madrid

Junio 2019

Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título

**IMPLEMENTACIÓN DE UN SISTEMA DE CADENA DE SUMINISTRO SOBRE
UNA INFRAESTRUCTURA HYPERLEDGER**

en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el

curso académico 2018/19 es de mi autoría, original e inédito y

no ha sido presentado con anterioridad a otros efectos.

El Proyecto no es plagio de otro, ni total ni parcialmente y la información que ha sido
tomada de otros documentos está debidamente referenciada.

Fdo.: Esther Vázquez Rodríguez Fecha: 12/07/2019

Autorizada la entrega del proyecto

EL DIRECTOR DEL PROYECTO

Fdo.: David Contreras Bárcena

Fecha: 12/07/2019

Vº Bº del Coordinador de Proyectos

Fdo.: Francisco Javier Herraiz

Fecha: 12/07/2019

AUTORIZACIÓN PARA LA DIGITALIZACIÓN, DEPÓSITO Y DIVULGACIÓN EN RED DE PROYECTOS FIN DE GRADO, FIN DE MÁSTER, TESINAS O MEMORIAS DE BACHILLERATO

1º. Declaración de la autoría y acreditación de la misma.

El autor D. ESTHER VÁZQUEZ RODRÍGUEZ

DECLARA ser el titular de los derechos de propiedad intelectual de la obra: DESPLIEGUE DE UN SISTEMA DE CADENA DE SUMINISTRO SOBRE UNA INFRAESTRUCTURA HYPERLEDGER ésta es una obra original, y que ostenta la condición de autor en el sentido que otorga la Ley de Propiedad Intelectual.

2º. Objeto y fines de la cesión.

Con el fin de dar la máxima difusión a la obra citada a través del Repositorio institucional de la Universidad, el autor **CEDE** a la Universidad Pontificia Comillas, de forma gratuita y no exclusiva, por el máximo plazo legal y con ámbito universal, los derechos de digitalización, de archivo, de reproducción, de distribución y de comunicación pública, incluido el derecho de puesta a disposición electrónica, tal y como se describen en la Ley de Propiedad Intelectual. El derecho de transformación se cede a los únicos efectos de lo dispuesto en la letra a) del apartado siguiente.

3º. Condiciones de la cesión y acceso

Sin perjuicio de la titularidad de la obra, que sigue correspondiendo a su autor, la cesión de derechos contemplada en esta licencia habilita para:

- a) Transformarla con el fin de adaptarla a cualquier tecnología que permita incorporarla a internet y hacerla accesible; incorporar metadatos para realizar el registro de la obra e incorporar “marcas de agua” o cualquier otro sistema de seguridad o de protección.
- b) Reproducirla en un soporte digital para su incorporación a una base de datos electrónica, incluyendo el derecho de reproducir y almacenar la obra en servidores, a los efectos de garantizar su seguridad, conservación y preservar el formato.
- c) Comunicarla, por defecto, a través de un archivo institucional abierto, accesible de modo libre y gratuito a través de internet.
- d) Cualquier otra forma de acceso (restringido, embargado, cerrado) deberá solicitarse expresamente y obedecer a causas justificadas.
- e) Asignar por defecto a estos trabajos una licencia Creative Commons.
- f) Asignar por defecto a estos trabajos un HANDLE (URL *persistente*).

4º. Derechos del autor.

El autor, en tanto que titular de una obra tiene derecho a:

- a) Que la Universidad identifique claramente su nombre como autor de la misma
- b) Comunicar y dar publicidad a la obra en la versión que ceda y en otras posteriores a través de cualquier medio.
- c) Solicitar la retirada de la obra del repositorio por causa justificada.
- d) Recibir notificación fehaciente de cualquier reclamación que puedan formular terceras personas en relación con la obra y, en particular, de reclamaciones relativas a los derechos de propiedad intelectual sobre ella.

5º. Deberes del autor.

- El autor se compromete a:

- a) Garantizar que el compromiso que adquiere mediante el presente escrito no infringe ningún derecho de terceros, ya sean de propiedad industrial, intelectual o cualquier otro.
- b) Garantizar que el contenido de las obras no atenta contra los derechos al honor, a la intimidad y a la imagen de terceros.
- c) Asumir toda reclamación o responsabilidad, incluyendo las indemnizaciones por daños, que pudieran ejercitarse contra la Universidad por terceros que vieran infringidos sus derechos e intereses a causa de la cesión.
- d) Asumir la responsabilidad en el caso de que las instituciones fueran condenadas por infracción de derechos derivada de las obras objeto de la cesión.

6º. Fines y funcionamiento del Repositorio Institucional.

La obra se pondrá a disposición de los usuarios para que hagan de ella un uso justo y respetuoso con los derechos del autor, según lo permitido por la legislación aplicable, y con fines de estudio, investigación, o cualquier otro fin lícito. Con dicha finalidad, la Universidad asume los siguientes deberes y se reserva las siguientes facultades:

- La Universidad informará a los usuarios del archivo sobre los usos permitidos, y no garantiza ni asume responsabilidad alguna por otras formas en que los usuarios hagan un uso posterior de las obras no conforme con la legislación vigente. El uso posterior, más allá de la copia privada, requerirá que se cite la fuente y se reconozca la autoría, que no se obtenga beneficio comercial, y que no se realicen obras derivadas.
- La Universidad no revisará el contenido de las obras, que en todo caso permanecerá bajo la responsabilidad exclusiva del autor y no estará obligada a ejercitar acciones legales en nombre del autor en el supuesto de infracciones a derechos de propiedad intelectual derivados del depósito y archivo de las obras. El autor renuncia a cualquier reclamación frente a la Universidad por las formas no ajustadas a la legislación vigente en que los usuarios hagan uso de las obras.
- La Universidad adoptará las medidas necesarias para la preservación de la obra en un futuro.
- La Universidad se reserva la facultad de retirar la obra, previa notificación al autor, en supuestos suficientemente justificados, o en caso de reclamaciones de terceros.

Madrid, a 12 de Julio de 2019

ACEPTA

Fdo ESTHER VÁZQUEZ RODRÍGUEZ

Motivos para solicitar el acceso restringido, cerrado o embargado del trabajo en el Repositorio Institucional:



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
GRADO EN INGENIERÍA EN TECNOLOGÍAS DE LA TELECOMUNICACIÓN

DESPLIEGUE DE UN SISTEMA DE CADENA DE SUMINISTRO SOBRE UNA INFRAESTRUCTURA HYPERLEDGER

Autor: Esther Vázquez Rodríguez

Director: David Contreras Bárcena

Madrid

Junio 2019

Agradecimientos

A mis padres y a mi hermano, por apoyarme incondicionalmente, por enseñarme el valor del trabajo y del esfuerzo, y por ser mi fuente de inspiración en la vida.

A mi familia, por confiar siempre en mí.

A mis amigos de verdad, por ser auténticos y fieles.

A David, por proponerme este proyecto, confiar en mí y por guiarme hasta el final.

DESPLIEGUE DE UN SISTEMA DE CADENA DE SUMINISTRO SOBRE UNA INFRAESTRUCTURA HYPERLEDGER

Autor: Esther Vázquez Rodríguez.

Director: David Contreras Bárcena.

Entidad Colaboradora: ICAI – Universidad Pontificia Comillas.

RESUMEN DEL PROYECTO

Como consecuencia del actual “estado del arte” de la Tecnología y su impacto directo en los modelos y formas de hacer negocio, la Transformación Digital se ha convertido en una necesidad imperiosa para la gran mayoría de las empresas. Existe un gran consenso sobre que la adecuada introducción, aplicación y uso de las hoy denominadas “Tecnologías Exponenciales” generará bienestar y progreso, y permitirá a las empresas ser más eficientes, productivas y competitivas, ofreciendo soluciones y productos de mayor valor y calidad, además de, en determinados casos, generar disrupción en los actuales modelos de negocio. Aprovechando la disponibilidad, auge y evolución de nuevas tecnologías como el *Internet of Things* (IoT) y Blockchain, se propone un proyecto que, apoyándose y haciendo uso de ambas tecnologías, implementa una solución de cadena de suministro sobre una red real Hyperledger.

Palabras clave: Blockchain, IoT, Hyperledger, Chaincode

1. Introducción

Las tecnologías evolucionan en progresión no lineal y actúan como catalizadoras y dinamizadoras de la globalización del mundo en todas sus vertientes; económica, social, cultural, empresarial, etc., fenómeno éste que lleva aparejado el incremento exponencial de vínculos y “conexiones” entre todos los intervinientes en la cadena de valor. Actualmente la vida se desarrolla en un mundo “conectado”, bien sea entre propios seres humanos, entre éstos y máquinas y dispositivos artificiales o entre ambos.

En el núcleo de esas conexiones e interacciones se encuentra el denominado *Internet of Things* (IoT), actuando como elemento habilitador que permite que todos estos dispositivos integrados con Internet se comuniquen entre sí, y así, con una explotación adecuada de todo ese entramado capturar el valor latente de esa “inteligencia distribuida”.

Un elemento crítico y prioritario en esta interconexión de millones de dispositivos (en crecimiento exponencial) es el poder garantizar la autenticidad e integridad de la información distribuida. Es precisamente aquí donde Blockchain adquiere realmente un protagonismo especial y diferencial.

Blockchain o cadena de bloques permite eliminar a los intermediarios, agilizar los tiempos de proceso, y gestionar la información compartiendo un registro descentralizado, distribuido, seguro e inmutable.

Aunque ambas tecnologías, IoT y Blockchain, en su conjunto siguen en fase de experimentación y desarrollo, este proyecto pretende desarrollar un caso de uso real y factible que permita sacar partido de ambas y ganar una ventaja competitiva para la empresa.

2. Definición del Proyecto

El presente proyecto se aborda en tres fases. La primera fase se basa en el estudio en profundidad de las tecnologías IoT y Blockchain, analizando las soluciones existentes en el mercado y estableciendo diferencias entre las mismas y el caso de uso planteado.

Tras explorar los distintos tipos de Blockchain existentes en el mercado, en la segunda fase se realiza el desarrollo del Back-End sobre Hyperledger.

Por último, en la tercera fase, se lanza el proyecto a una red Hyperledger sobre una interfaz que permite ver la interacción de los diferentes contratos y poner en funcionamiento un caso de uso de cadena de suministro con sensores IoT.

3. Descripción del modelo/sistema/herramienta

El caso de uso implementado en este proyecto consiste en una aplicación basada en el concepto de cadena de suministro en el que se visualizan los diferentes pasos que se efectúan en la adquisición de un producto por un cliente, desde su compra hasta su entrega final, de tal manera que todo el proceso y sus diferentes interacciones quedan registradas en un libro contable inmutable y criptográficamente seguro.

El sistema representa un modelo que trabaja a nivel de aplicación de red Blockchain, mediante la red Hyperledger, permitiendo mantener en todo momento el registro del producto de manera criptográficamente segura e inmutable. La red de Hyperledger interacciona con un API REST que a su vez permite lanzar el interfaz de usuario con Ionic gracias a NodeJs.

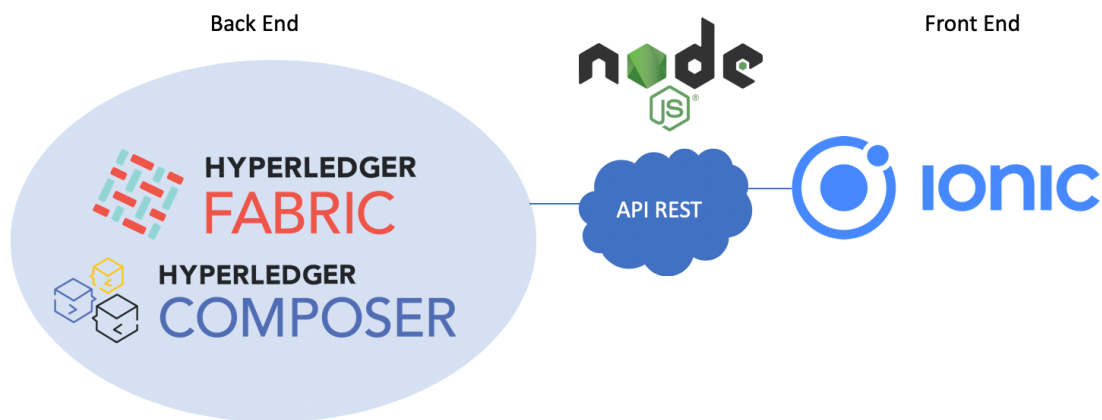


Figura 1: Esquema de las tecnologías usadas

4. Resultados

Se puede concluir que los resultados obtenidos tras el desarrollo final del proyecto han cumplido con los objetivos propuestos inicialmente. A través de la aplicación, denominada Exchange, el comprador es capaz de visualizar los productos, ver sus pedidos y ver el estado del pedido (Figura 2). Por otro lado, el proveedor tiene la opción de confirmar el pedido realizado por el comprador (Figura 3) siendo el sensor IoT, el que registra cuando el pedido se encuentra en tránsito o entregado, así como información sobre cuál es su temperatura actual y sus coordenadas de GPS (Figura 4).

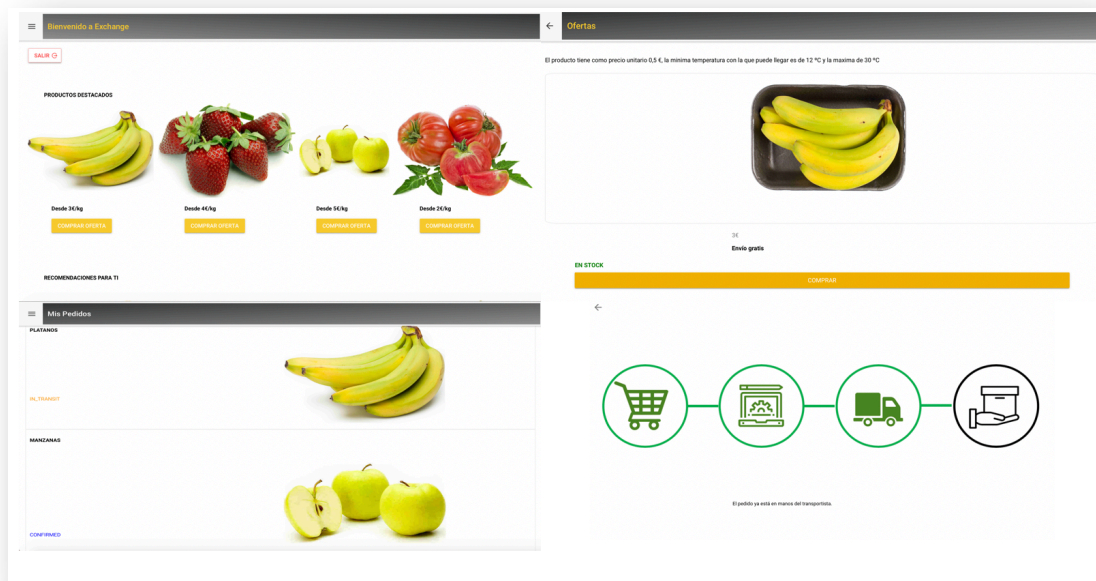


Figura 2: Vista del Comprador

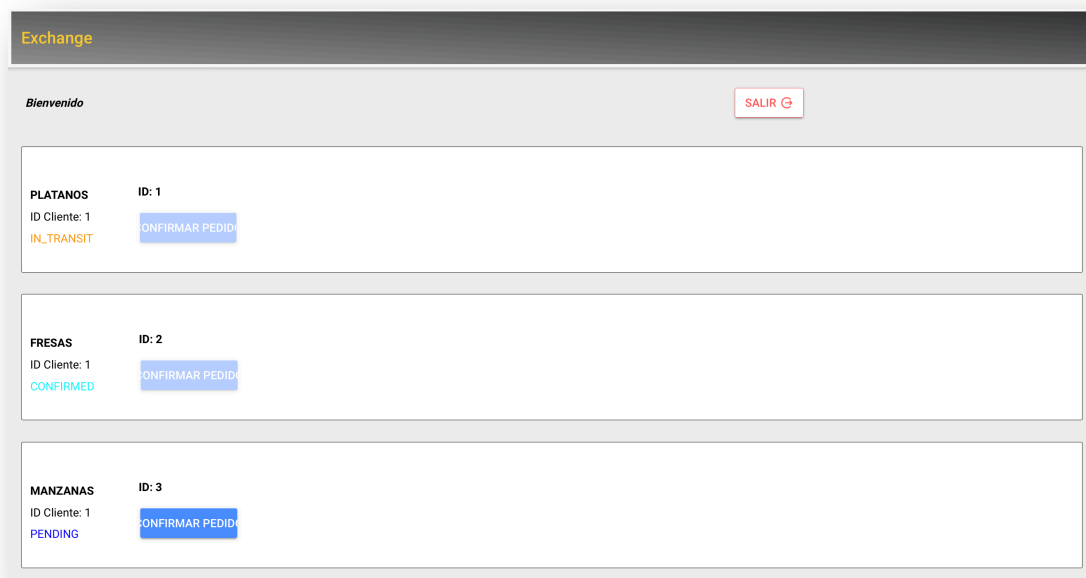


Figura 3: Vista del Vendedor

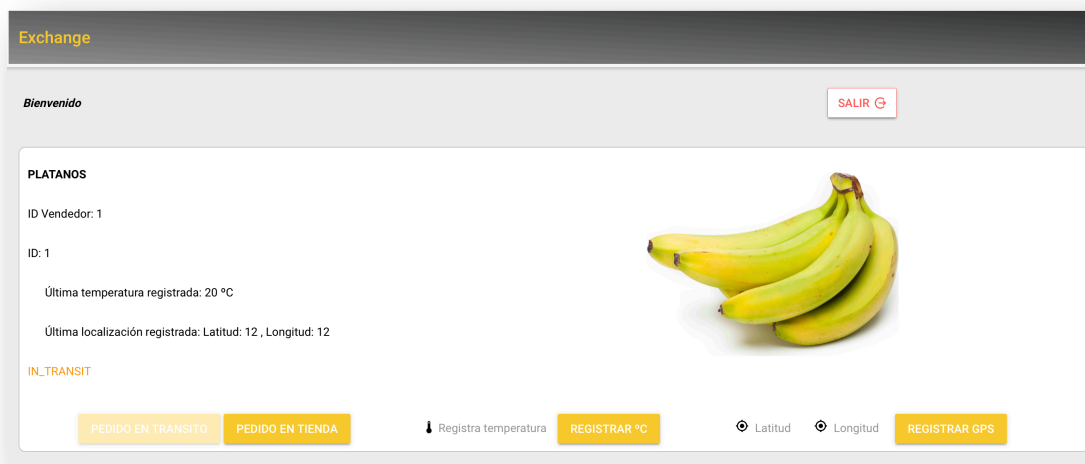


Figura 4: Vista del Sensor IoT

5. Conclusiones

Es indudable que Blockchain ha seguido un ritmo de evolución e interés creciente en todos los ámbitos; empresarial, gubernamental, académico, social, etc. Adicionalmente, cabe destacar su capacidad y potencial inherente para producir disrupción en muchos de los actuales modelos de negocio en todos los sectores de actividad: banca, seguros, energía, utilities, agricultura, etc.

En el presente proyecto se ha profundizado en el estudio, conocimiento y uso de esta tecnología, con el objetivo básico de proporcionar una solución aplicable a un problema de negocio. Se ha obtenido una visión integral y profunda de la tecnología Blockchain y, más en concreto, de la solución Hyperlegder, desde lo más básico de ésta hasta su aplicabilidad a un caso concreto de negocio. Se ha desplegado, además, el *Front End*, el cual, a pesar de lo inicialmente previsto, ha resultado especialmente laborioso y complejo, dados los diferentes problemas de madurez en la configuración y versionado de esta solución.

La tecnología Blockchain sin duda añade eficiencia y productividad a muchos problemas de negocio, simplificando su sistema de contabilización y dotando de inmediatez al propio problema de negocio, al no existir intermediarios.

Sin embargo, y de cara al futuro, esta tecnología debe madurar, requiriendo de una actualización constante para estar al día de las novedades y mejoras que vayan apareciendo.

DEPLOYMENT OF A SUPPLY CHAIN SYSTEM ON A HYPERLEDGER INFRASTRUCTURE

Author: Esther Vázquez Rodríguez.

Supervisor: David Contreras Bárcena.

Collaborating Entity: ICAI – Universidad Pontificia Comillas.

ABSTRACT

As a result of the current technology "state of the art" and its direct impact on models and ways of doing business, digital transformation has become an imperative for the vast majority of companies. In particular, the introduction, application and use of the so-called "Exponential Technologies" will generate welfare and progress, as companies will be more efficient, productive and competitive, offering solutions and products of greater value and quality, in addition to generating disruption in many of the current business models.

Taking advantage of the availability, boom and evolution of new technologies such as IoT and Blockchain, we have proposed a project that, supporting and making use of both technologies, we implement an IoT solution over a real Hyperledger network.

Keywords: Blockchain, IoT, Hyperledger, Chaincode

1. Introduction

The technologies evolve in a non-linear progression and act as catalysts and energizers of the globalization of the world in all its aspects, economic, social, cultural, business, etc., a phenomenon that entails the exponential increase of links and "connections" between all the elements intervening in the value chain. Nowadays, we live in a "connected" world, either between human beings themselves, between them and artificial machines and devices or between both of them.

At the core of these connections and interactions is the so-called Internet of Things (IoT), acting as an enabling element that allows all these devices integrated with the Internet to communicate with each other, so as to properly exploit this whole network to capture the latent value of that "distributed intelligence".

A critical and priority element in this interconnection of millions of devices (in exponential growth) is the power to guarantee the authenticity and integrity of the distributed information. It is precisely here where Blockchain really takes center stage.

Blockchain or chain of blocks proposes to eliminate intermediaries, streamline process times and manage information by sharing a decentralized, distributed, secure and immutable registry.

Although both technologies, IoT and Blockchain, are still in the experimental and development phase, this project aims to develop a real and feasible use case that allows the company to take advantage and gain a competitive advantage.

2. Project definition

This project is addressed in three phases. The first phase is based on the in-depth study of the IoT and Blockchain technologies, analyzing the existing solutions in the market and establishing differences between them and the case of proposed use.

After exploring the different Blockchain types existing in the market, the second phase begins with the development of the Back-End over Hyperledger.

Finally, the project is launched to a Hyperledger network on an interface that allows to see the interaction of contracts and put into operation a case of use of supply chain with IoT sensors.

3. System description

The use case implemented in this project consists of an application based on the supply chain concept in which the different steps that are carried out in the acquisition of a product by a customer, from its purchase to its final delivery, are viewed. so that the whole process and its different interactions are recorded in an unchangeable and cryptographically secure accounting book.

The system is a model that works at the Blockchain network application level, through the Hyperledger network, which allows the product registration to be maintained in a cryptographically secure and immutable manner at all times. The Hyperledger network keeps in contact with a REST API which in turn allows launching the user interface with Ionic thanks to NodeJs.

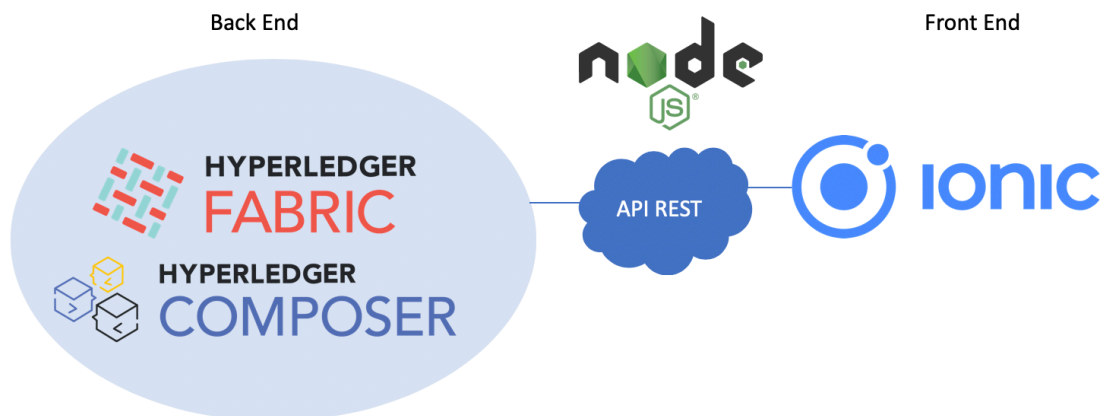


Figure 5: Technologies summary

4. Results

The results that have been obtained after the final development of the project have met the objectives proposed at the beginning. Through the application, called Exchange, the buyer is able to view the products, see their orders, see the status of the order (Figure 6). On the other hand, the supplier has the option to confirm the order made by the buyer (Figure 7) and the IoT sensor, which registers when the order is in transit, delivered, what is its current temperature and its GPS coordinates (Figure 8).

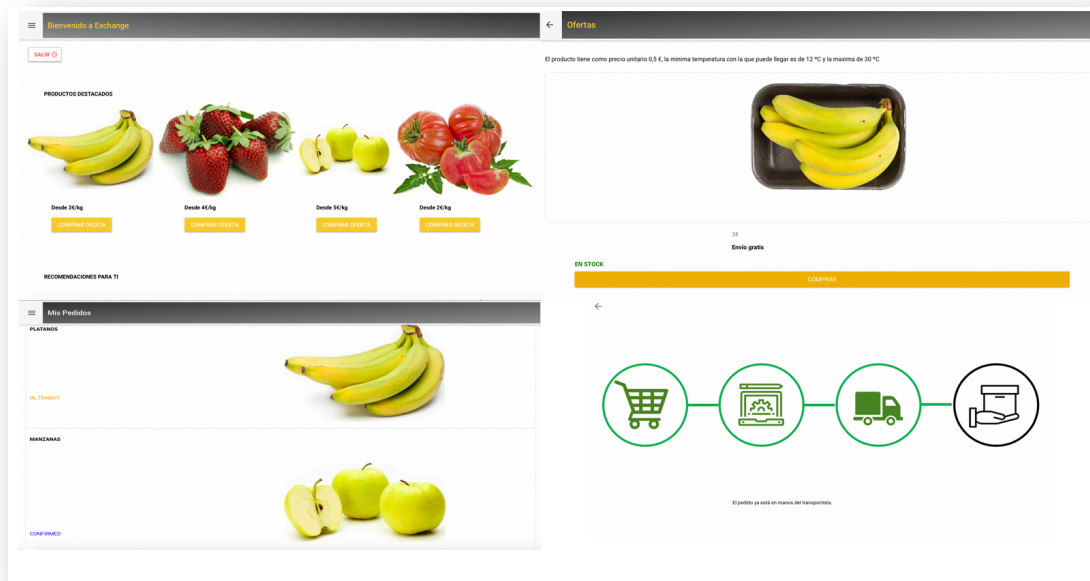


Figure 6: Buyer's view

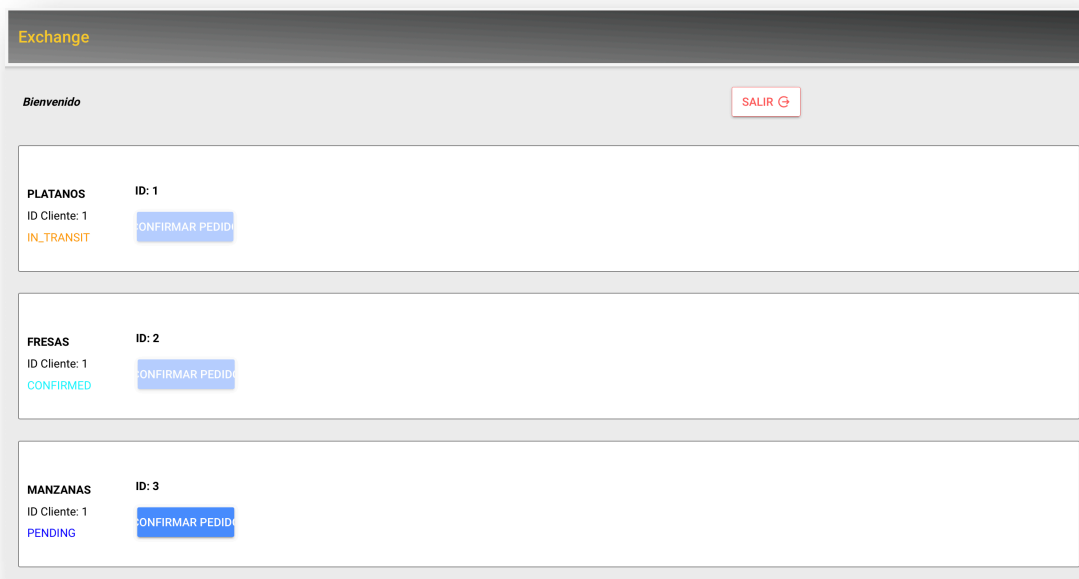


Figure 7: Vendor's view

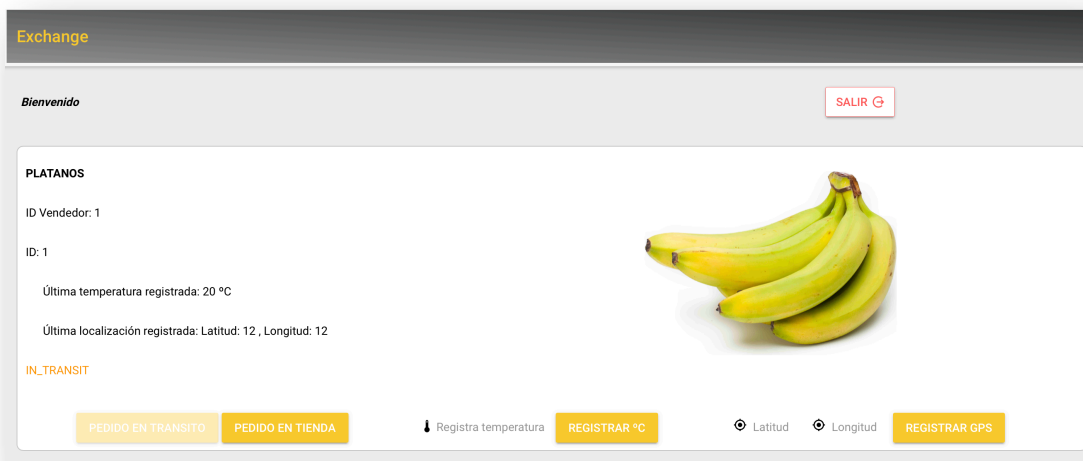


Figure 8: Sensor's view

5. Conclusions

Blockchain has followed a pace of evolution and growing interest, coming from its capacity and inherent potential to produce disruption in many of the current business models in all sectors of activity: banking, insurance, energy, utilities, agriculture, etc.

In the present project the study, knowledge and use of this technology has been deepened, with the basic objective of providing a solution applicable to a business problem. We have obtained an integral and deep vision of the Blockchain technology and, more specifically, of the Hyperledger solution, from the most basic of this to its full applicability. In addition, the Front End has been deployed, which, despite what was initially planned, has been particularly laborious and complex, given the different maturity problems in the configuration and versioning of this solution.

The Blockchain technology undoubtedly adds efficiency and productivity to many business problems, simplifying its accounting system and providing immediacy to the business problem itself, as there are no intermediaries.

However, and for the future, this technology must mature, and requires a constant update to keep abreast of the news and improvements that are appearing.

Índice de la memoria

Capítulo 1. Introducción	6
Capítulo 2. Descripción de las Tecnologías.....	8
2.1 Blockchain.....	8
2.1.1 Definición	8
2.1.2 Funcionamiento	9
2.1.3 Algoritmos De Consenso	10
2.1.4 Tipos De Blockchain.....	13
2.1.5 Bitcoin. La Primera Blockchain	14
2.2 Hyperledger Vs Ethereum	14
2.2.1 Ethereum.....	14
2.2.2 Hyperledger	15
2.2.3 Diferencias.....	15
2.3 Hyperledger Project.....	17
2.3.1 Arquitectura.....	17
2.3.2 Frameworks	18
2.3.3 Herramientas	27
2.4 Otras tecnologías usadas.....	30
2.4.1 Docker	30
2.4.2 Angular App.....	31
2.4.3 GIT.....	31
2.4.4 NodeJS.....	32
2.4.5 Ionic	32
Capítulo 3. Estado de la Cuestión	33
3.1 Internet of Things (IoT).....	33
3.1.1 Retrospectiva	33
3.1.2 Desafíos Iot.....	34
3.1.3 Redes De IoT Descentralizadas.....	36

ÍNDICE DE LA MEMORIA

3.1.4 Casos De Uso De IoT	37
3.2 Blockchain	38
3.2.1 Retrospectiva	38
3.2.2 Casos De Uso De Blockchain.....	40
3.3 Blockchain e IoT	41
3.3.1 Casos de uso de IoT con Blockchain.....	41
Capítulo 4. Definición del Trabajo	43
4.1 Justificación.....	43
4.2 Elección de la tecnología Hyperledger.....	43
4.3 Objetivos.....	44
4.4 Metodología.....	44
4.5 Planificación y Estimación Económica	47
Capítulo 5. Modelo Desarrollado.....	48
5.1 Análisis	48
5.1.1 Diagrama de Clases	49
5.1.2 Diagrama de Casos de uso.....	54
5.2 Diseño.....	56
5.3 Implementación	57
5.3.1 Desarrollo Back End	57
5.3.2 Desarrollo Front End	60
Capítulo 6. Análisis de Resultados.....	61
6.1 Hyperledger Composer.....	61
6.2 API REST	65
6.3 Front End	67
Capítulo 7. Conclusiones y Trabajos Futuros.....	74
Capítulo 8. Bibliografía.....	76

Índice de figuras

Figura 1: Esquema de las tecnologías usadas	11
Figura 2: Vista del Comprador	12
Figura 3: Vista del Vendedor.....	12
Figura 4: Vista del Sensor IoT.....	13
Figure 5: Technologies summary	15
Figure 6: Buyer's view.....	16
Figure 7: Vendor's view.....	17
Figure 8: Sensor's view.....	17
Figura 9: Comparativa de consenso entre redes públicas y privadas	13
Figura 10: Arquitectura de Hyperledger Project	18
Figura 11: Componentes de una red Hyperledger Fabric.....	20
Figura 12: Diagrama de red Hyperledger	21
Figura 13: Flujo de una solicitud de alto nivel en Hyperledger Fabric	22
Figura 14: Flujo de una solicitud de alto nivel en Hyperledger Fabric (Paso 1).....	22
Figura 15: Flujo de una solicitud de alto nivel en Hyperledger Fabric (Paso 2).....	23
Figura 16: Flujo de una solicitud de alto nivel en Hyperledger Fabric (Paso 3).....	24
Figura 17: Flujo de una solicitud de alto nivel en Hyperledger Fabric (Paso 4).....	24
Figura 18: Flujo de una solicitud de alto nivel en Hyperledger Fabric (Paso 5).....	25
Figura 19: Esquema de los archivos de programación del Back End	29
Figura 20: Evolución del IoT.....	35
Figura 21: Esquema de uso de Trello	46
Figura 22: Diagrama de Gantt del proyecto	46
Figura 23: Diagrama de clases.....	49
Figura 24: Transacción de Registrase y de Iniciar Sesión.....	51
Figura 25: Transacción compra de producto	51
Figura 26: Transacción de confirmación de envío	52
Figura 27: Transacciones de Registro de pedido en Tránsito o Entregado	52
Figura 28: Transacciones de Lectura de Temperatura y de GPS	53

INTRODUCCIÓN

Figura 29: Diagrama de caso de uso del Comprador.....	54
Figura 30: Diagrama de caso de uso del Proveedor	55
Figura 31: Diagrama de caso de uso del Sensor IoT	55
Figura 32: Esquema de las tecnologías usadas en este proyecto	56
Figura 33: Proceso de creación de la aplicación.....	57
Figura 34: Hyperledger Composer Playground.....	61
Figura 35: Hyperledger Composer Playground (Define)	62
Figura 36: Hyperledger Composer Playground (Test)	63
Figura 37: Hyperledger Composer Playground (Submit Transaction).....	64
Figura 38: Hyperledger Composer Playground (Ledger).....	64
Figura 39: Hyperledger Composer Playground (Ledger).....	65
Figura 40: API REST Exchange.....	66
Figura 41: API REST Exchange.....	66
Figura 42: API REST Exchange.....	67
Figura 43: Página principal: Login.....	68
Figura 44: Página de Registro	68
Figura 45: Página vista por el Comprador.....	69
Figura 46: Página vista por el Comprador.....	69
Figura 47: Página vista por el Comprador.....	70
Figura 48: Página vista por el Comprador.....	70
Figura 49: Página vista por el Comprador.....	71
Figura 50: Página vista por el Comprador.....	71
Figura 51: Página vista por el Vendedor	72
Figura 52: Página vista por el Sensor	72
Figura 53: Transacción de Login.....	73
Figura 54: Transacción de Lectura de Temperatura	73
Figura 55: Transacción Compra de Producto	73
Figura 56: Transacción de lectura de GPS	73

Índice de tablas

Tabla 1: Tipos de Blockchain.....	13
Tabla 2: Resumen de la comparativa (menor a mayor puntuación, 1-5).....	16
Tabla 3: Estimación de costes	47

Capítulo 1. INTRODUCCIÓN

En la sociedad actual es indudable que la tecnología juega un papel muy importante en la vida de las personas. Si bien es cierto que existen ciertas voces que alertan sobre las amenazas y potenciales efectos negativos que la tecnología puede tener sobre la humanidad, existe un consenso general y mayoritario de que es (y ha sido) fuente esencial para nuestro bienestar y progreso. La historia de la humanidad revela que la innovación basada en la introducción, aplicación y uso de la ciencia, nuevas técnicas y tecnologías genera progreso y bienestar. Incluso, hasta ahora, también en términos de empleabilidad hay que decir que, aunque los periodos transitorios en determinadas ocasiones hayan sido duros, una vez transcurridos éstos, los nuevos empleos que se han generado han compensado, y a menudo con creces, los que han sido destruidos. Además de cambiar muchas de las pautas, hábitos y conductas en la vida. Las nuevas tecnologías digitales tienen además el potencial de revolucionar la industria tradicional, al ayudar a las empresas a trabajar de manera más ágil, precisa, eficiente y sostenible. El término Industria 4.0 o Cuarta Revolución Industrial permite agilizar los procesos y lograr una producción de mayor calidad, optimizando costes, dotando de la adecuada sensorización y sistemas de información soporte.

Un concepto ligado a la Industria 4.0 es el *Internet of Things (IoT)*, que combina el mundo físico con el virtual, creando un entorno potencialmente inteligente. Tal y como lo recoge la Comisión Europea, “Internet of Things (IoT) representa el siguiente paso hacia la digitalización de nuestra sociedad y economía, donde los objetos y las personas están interconectados a través de redes de comunicación e informan sobre su estado y/o el entorno que les rodea” [1]. IoT se considera como la propia evolución o transformación del Internet que todos conocemos. Continuamente se están estudiando patrones y conductas de comportamiento que permitan dotar a los objetos de inteligencia como consecuencia de la interconexión entre sí.

La realidad es que hoy en día el auge de dispositivos conectados es notorio y palpable en múltiples sectores de actividad como, por ejemplo, el industrial, el doméstico y las ciudades inteligentes. Ejemplos de IoT hay muchos [2], abarcando desde la hostelería, comercios de

INTRODUCCIÓN

cara al público, flotas de vehículos para logística, agricultura y ganadería, uso doméstico (domótica), jardinería, salud, gestión de tráfico, etcétera.

Sin embargo, el mundo empresarial se enfrenta al gran desafío de aprovechar el potencial del Internet de las cosas (Internet of Things, IoT) para reducir costos, generar nuevos modelos y maneras de hacer negocios y ofrecer más y mejores productos y servicios al mercado, todo ello de manera segura. Los dispositivos envían información recopilada tan sensible como los datos personales, obligando en este caso, a garantizar la privacidad e integridad de la información como factores clave a proteger. Es aquí donde Blockchain adquiere un protagonismo esencial y singular.

Blockchain, en español cadena de bloques, fue empleado por primera vez en el año 2008 por Satoshi Nakamoto acuñando la primera moneda electrónica. Es una tecnología que da solución a los problemas de trazabilidad, eliminando el factor del error humano y agilizando los tiempos de proceso. Además, elimina las cadenas de intermediación, descentralizando la gestión de los intercambios y sus operaciones asociadas. Se podría describir sintéticamente con cuatro características básicas: descentralizado y distribuido dado que no hay un gobierno central que lo controla; seguro pues usa un sistema de claves criptográficas; e inmutable porque la cadena de bloques actúa como un libro de contabilidad, donde el historial de transacciones no puede ser modificado.

Actualmente existen diferentes tipos de Blockchain, privados, públicos, permissionados, no permissionados... Por tanto, para hacer un buen uso de esta tecnología, es muy importante entender los conceptos técnicos de Blockchain para comprender las consecuencias de las diversas arquitecturas existentes respecto a la seguridad, rendimiento y privacidad que proporcionan.

El presente trabajo representa una oportunidad para estudiar, desarrollar e implementar una solución basada en IoT y Blockchain, habilitando una red de dispositivos de bajo coste escalable y conservando la integridad de los datos que transporta y almacena.

Capítulo 2. DESCRIPCIÓN DE LAS TECNOLOGÍAS

Este apartado describe las diferentes tecnologías, protocolos y herramientas específicas que se han usado en este proyecto, con el objetivo básico de facilitar su lectura y comprensión.

2.1 *BLOCKCHAIN*

2.1.1 DEFINICIÓN

De manera simple se puede definir Blockchain como un libro de contabilidad descentralizado que permite a los participantes realizar y confirmar transacciones y conciliar las obligaciones subyacentes de forma segura, rápida y rentable, gracias a una infraestructura de base de datos transaccional común y compartida.

Blockchain es una base de datos que se halla distribuida entre diferentes participantes, protegida criptográficamente y organizada en bloques de transacciones relacionados entre sí algorítmicamente. Se trata de un sistema que permite que partes que no confían plenamente unas en otras (incluso que no se conozcan) puedan mantener un consenso sobre la existencia, el estado y la evolución de una serie de factores compartidos y así transaccionar con confianza.

A continuación, se muestran cuatro de los atributos que mejor definen a Blockchain:

Inmutable: Las transacciones, los contratos y los datos que son ejecutados y grabados en Blockchain no se pueden eliminar.

Descentralizado: No es necesario tener un gobierno central que dé fe sobre la veracidad de una transacción válida, es decir, un gobierno central que controle toda transacción. No requiere ni necesita intermediarios.

Consensuado: Casi todos los participantes en la red pueden manejar la misma información actualizada, de tal manera que, si uno o varios fallan, la comprobación y actualización posibilita reparar o corregir los errores gracias a que existen diferentes métodos de consenso entre ellos.

Seguro criptográficamente: La información se comparte de manera segura gracias a avanzados métodos criptográficos de seguridad.

2.1.2 FUNCIONAMIENTO

Blockchain es una tecnología que permite la transferencia e interacción de datos digitales de una manera segura. Se podría hacer una analogía con un libro contable, donde se listan todas las entradas y salidas de dinero, en este caso, todas las transacciones. Las transacciones son las interacciones que se registran dentro de la cadena de bloques y cada paquete de transacciones se conoce como bloque, por lo que cada uno de los bloques de la cadena contiene información codificada de cada transacción en la red. Es importante entender que la transferencia no requiere de un intermediario que identifique y certifique la información, sino que está distribuida.

Los participantes en la red de Blockchain se denominan nodos, y serán estos nodos los que tengan que certificar que los datos que se comparten en la red son verdaderos. Ahora bien, ¿cómo lo hacen?

Los bloques de Blockchain están entrelazados o encadenados por códigos o apuntadores denominados *hash*¹. Cada bloque de la cadena porta el paquete de transacciones y dos códigos, uno que indica el código de bloque que lo precede y otro para el que le sigue.

Anteriormente se hacía mención expresa sobre métodos de consenso. Para comprender bien su significado hay que explicar el concepto de minado o *mining*, que es realizado por los nodos. Los nodos que minan compiten para resolver una serie de problemas matemáticos que protegen los contenidos del bloque. Esta comprobación consiste en que cuando hay dos bloques que apuntan al mismo bloque previo, gana el primero en ser descryptado por la mayoría de los nodos. [3] Este proceso requiere un alto coste computacional y de electricidad, por lo que los ganadores de la competición son recompensados por su servicio. Una vez realizado el proceso de minado, se notifica a los otros miembros que verifican y añaden este nuevo bloque a sus copias de la cadena. Esto se hace gracias al anteriormente citado *hashing*.

Una función hash es un procedimiento criptográfico que emplea un algoritmo específico para transformar una información determinada, en una secuencia alfanumérica única y de longitud fija. [4] De esta manera, la misma información siempre producirá el mismo hash.

¹ Algoritmo matemático que transforma cualquier bloque arbitrario de datos en una nueva serie de caracteres con una longitud fija. Independientemente de la longitud de los datos de entrada, el valor hash de salida tendrá siempre la misma longitud.

DESCRIPCIÓN DE LAS TECNOLOGÍAS

La información que no se verifica podría ser hasheada con la misma función y comparada con la original, validando así la información compartida. Una vez la mayoría de los mineros validan el nuevo bloque, entonces este nuevo bloque lo añaden a sus copias de cadena de forma permanente e inmutable.

Sin embargo, anteriormente se mencionaba que existen diversos tipos de Blockchain, en algunos casos, como es el de Hyperledger, el concepto de minado no existe. Más adelante se explican, con cierto grado de detalle, las principales diferencias que presenta Hyperledger con respecto al resto de Blockchains.

2.1.3 ALGORITMOS DE CONSENSO

En Blockchain, al no existir un gobierno central que regule y apruebe las transacciones entre los distintos participantes, resulta necesario introducir una solución alternativa al problema de la confianza y seguridad transaccional. En este caso, son los propios nodos participantes en la red los que se encargan de consensuar estas transacciones mediante diferentes métodos de consenso. Se aplican algoritmos de consenso (difíciles de engañar y manipular, y en consecuencia que permitan mantener la integridad y confidencialidad de los datos. [5]). Estos algoritmos habilitan acuerdos sobre cuáles son los bloques de transacciones que representan el histórico según una serie de reglas predeterminadas. Ejemplos de estos algoritmos son:

Proof-of-work (PoW)

Este es el algoritmo más usado hasta el momento. Consiste en eliminar los ataques a redes informáticas a través de la realización de una prueba difícil (un problema matemático, por ejemplo) antes de permitir la realización de otra acción. Se requiere de equipos avanzados y un elevado consumo de electricidad para resolverlos. El sistema, de esta manera, lo que hace es protegerse contra los nodos que tratan de burlar a la red, obligando a que éstos tengan que resolver una cantidad de problemas de extrema complejidad, redundando en que la posibilidad de que efectivamente lo hagan se vuelva insignificante.

Adicionalmente, este método, como consecuencia de requerir una cantidad ingente de recursos, en términos de capacidad de cómputo y electricidad, resulta perjudicial para el medio ambiente.

Proof of stake (PoS)

DESCRIPCIÓN DE LAS TECNOLOGÍAS

Proof of Stake es un algoritmo que nace como alternativa a los problemas descritos en PoW y los altos costes asociados a los equipos de minado. Aquí, los participantes sólo pueden contribuir a la computación en la proporción en la cual poseen recursos en el sistema. Resulta más seguro y además es más sustentable y mejor para el medio ambiente, pues los cálculos son más fáciles de computar y por tanto requieren de menos capacidad de cómputo y gastos asociados a ésta. Además, permite que cualquiera participe, no solamente aquellos que puedan adquirir equipos especializados con un costo elevado.

Sin embargo, también presenta una serie de problemas. Potencia el acaparamiento, pues si por ejemplo se aplica al caso de criptomoneda, cuántas más monedas tiene un usuario, más recompensa recibe. Esto fomentaría que el número de monedas circulante descendiera y por tanto se aumentara su precio de manera artificial.

Delegated proof of stake (DPoS)

DPoS es un algoritmo similar al PoS en el sentido de participación. Pero DPoS presenta un novedoso sistema de votación democrática, donde los nodos de cada bloque son elegidos. Sigue siendo descentralizado, pero presenta el problema de que cuando el conjunto de nodos es muy pequeño se puede volver centralizado, aunque así más rápido y eficiente en el consenso de las transacciones.

Proof of importance (PoI)

PoI asigna un porcentaje, una puntuación a cada participante basado en su participación en el sistema, en vez de sólo en sus recursos (por ejemplo, cantidad de criptomonedas). Esto, en el caso de su aplicación a la criptomoneda, ayuda a prevenir que los participantes del sistema se dediquen a acumular monedas, y por tanto el que más monedas mueva, más puntos recibiría.

Leased proof-of-stake (LPoS)

LPoS es otra de las variantes que presenta el algoritmo PoS. Fue desarrollada por la plataforma WAVES, e intenta resolver el problema de concentración de riqueza que afecta a PoS. LPoS para ello genera una serie de incentivos para que aquellos con pocas monedas las alquilen a los nodos, de tal manera que éstos adquieren mayor peso y aumenten sus probabilidades de agregar el siguiente bloque en la cadena.

Proof-of-elapsed-time (PoET)

DESCRIPCIÓN DE LAS TECNOLOGÍAS

Este algoritmo de consenso es principalmente utilizado en Hyperledger Sawtooth, fue originalmente desarrollado por Intel. Es aplicable tanto a Blockchains públicas como privadas o permissionadas. Permite que los nodos de una Blockchain dotada de permisos alcancen un consenso, incluso cuando las partes no se conocen entre sí. Es similar a PoW, pero dado que utiliza computación confiable para definir los tiempos de espera de manera aleatoria para la construcción de bloques, presenta un mejor consumo de energía.

Sin embargo, requiere de una confianza en los chips de Intel, no está alineado directamente con la característica principal de descentralización de Blockchain.

Practical byzantine fault tolerance (PBFT)

Este algoritmo de consenso es el más popular en Blockchains privadas y dotadas de permisos, como Hyperledger Fabric.

En pocas palabras, el problema de los generales bizantinos fue concebido en 1982 como un dilema lógico que ilustra cómo un grupo de generales bizantinos puede tener problemas de comunicación cuando intentan ponerse de acuerdo sobre su próximo movimiento. El dilema supone que cada general tiene su propio ejército y que cada grupo está situado en diferentes lugares alrededor de la ciudad que intentan atacar. Los generales deben estar de acuerdo en atacar o retirarse. No importa si atacan o se retiran, siempre que todos los generales alcancen el consenso, es decir, acuerden una decisión común para ejecutarla en coordinación. Por lo tanto, se pueden considerar los siguientes requisitos [6]: Cada general tiene que decidir: atacar o retirarse (sí o no); Después de que se toma la decisión, no se puede cambiar; Todos los generales deben acordar la misma decisión y ejecutarla de manera sincronizada. PoW generalmente provee una solución tolerante a fallas bizantinas, pero dado su alto consumo de energía resulta difícil de implementar.

Sin embargo, en redes permissionadas como Hyperledger Fabric, una decisión consensuada es determinada por todos los participantes en la red. Por defecto, hay un problema de escalabilidad porque este algoritmo está principalmente diseñado de tal manera que, si hay un elevado número de nodos, podría dejar de funcionar. A pesar de esto y gracias a como está diseñada la arquitectura de Hyperledger Fabric (se verá más adelante), esto no ocurre en todos los nodos. Los beneficios de evitar usar PoW pueden ser superiores que los riesgos

DESCRIPCIÓN DE LAS TECNOLOGÍAS

asociados. Sin embargo, sólo funciona en Blockchains con permisos pues requiere que los agentes no sean anónimos.

En la siguiente figura se muestra una comparativa entre los consensos de redes privadas y el consenso PoF usado por Bitcoin (redes públicas) [7].

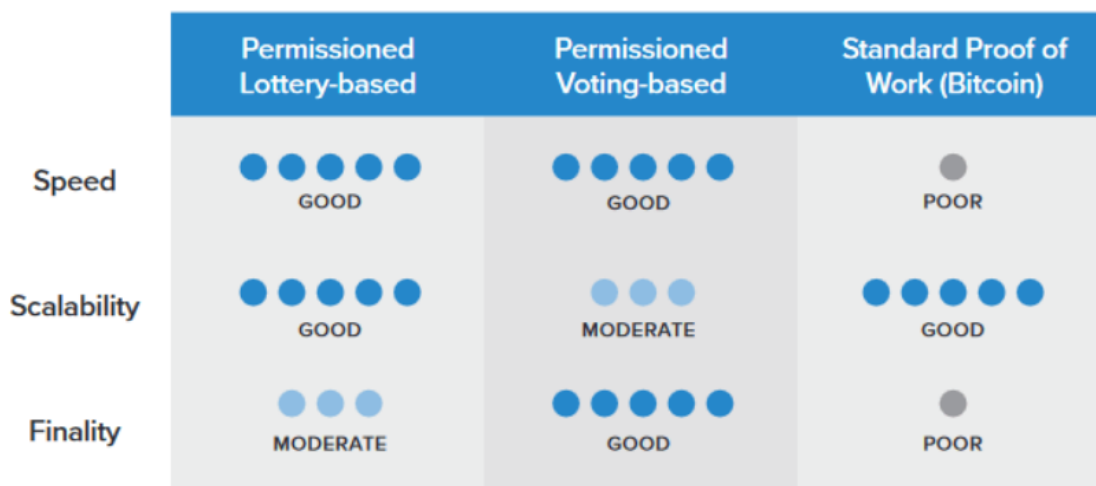


Figura 9: Comparativa de consenso entre redes públicas y privadas

2.1.4 TIPOS DE BLOCKCHAIN

Lo habitual es pensar que existe únicamente un tipo de Blockchain. Pero en realidad, existen diferentes tipos de Blockchain que se explican de manera muy sencilla en la siguiente tabla:

	Público	Privado
Permisado	Cualquier persona que cumpla con ciertos criterios predefinidos puede descargar el protocolo y validar transacciones.	Se requiere que el validador tenga una licencia o sea parte de un grupo limitado. Bueno para las empresas, ya que puede controlar el acceso y el contenido, y la información no tiene que estar disponible públicamente.
No permisado	Protocolo de código abierto. Cualquier persona puede acceder y validar transacciones fácilmente, es bueno para los individuos pues no necesitan intermediarios.	A menudo, la gestión de base de datos o la auditoría son internas a una sola empresa, por lo que en muchos casos la legibilidad pública puede no ser necesaria en absoluto.

Tabla 1: Tipos de Blockchain

Público: Cualquier persona puede descargarse el código y administrar un nodo público en su máquina local, validando transacciones en la red y participando en el proceso de consenso,

DESCRIPCIÓN DE LAS TECNOLOGÍAS

lo que hace que cualquiera pueda participar en el proceso de qué bloques se introducen en la cadena. Además, cualquier persona puede realizar transacciones y acceder y ver esas transacciones anónimas.

Privado: En este tipo de Blockchain los permisos están restringidos a una organización.

Permisado: Sólo nodos aprobados pueden unirse/modificar. Tienen la ventaja de que son más rápidas y tienen mayor escalabilidad y privacidad de las transacciones.

No permisado: Cualquier persona se puede unir.

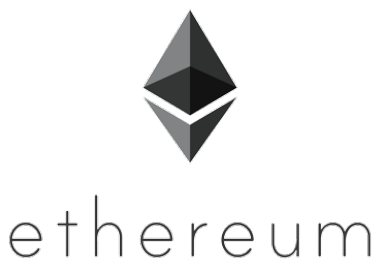
2.1.5 BITCOIN. LA PRIMERA BLOCKCHAIN

Bitcoin fue la primera aplicación Blockchain, creada por una persona o, tal vez, grupo bajo el nombre de Satoshi Nakamoto en 2008. Esta primera aplicación fue diseñada como un sistema de pago descentralizado, seguro y rápido. Está basado en un sistema “Peer To Peer”, o de usuario a usuario, que ha permitido romper con los problemas de pago al no necesitar un gobierno central que lo valide. Posibilita el cambio de criptomonedas (o bitcoins) que es accesible por todo el mundo.

La plataforma utiliza el algoritmo criptográfico Sha256 y el lenguaje de programación es C++.

2.2 *HYPERLEDGER VS ETHEREUM*

2.2.1 ETHEREUM



Ethereum es una Blockchain o Tecnología de Contabilidad Distribuida (DLT) con un lenguaje de programación Turing integrado *opensource*, que permite a cualquiera escribir *smartcontracts* y aplicaciones descentralizadas. Es un tipo de Blockchain público y no permisado que está probado y adquiriendo usuarios diariamente.

DESCRIPCIÓN DE LAS TECNOLOGÍAS

Vitalik Buterin fue quien desarrolló Ethereum como una extensión del concepto original de Blockchain [8]. Improvisó los protocolos de Bitcoin para admitir las aplicaciones más allá de la emisión de divisas, proporcionando características avanzadas como límites de retiro, contratos financieros, mercado de juegos del azar, etc. Actualmente, unos consorcios de importantes instituciones financieras se han unido para construir una plataforma Ethereum para la industria bancaria.

Cualquier persona en todo el mundo puede conectarse con la cadena de bloques Ethereum y mantener el estado actual de la red. Por tanto, Ethereum también se conoce como la “World Computer” (computadora del mundo).

2.2.2 HYPERLEDGER



Hyperledger es una plataforma también de código abierto para Blockchain, que surge en diciembre de 2015 bajo la dirección de varias organizaciones, aunque actualmente forma parte de la Fundación Linux. Basándonos en los distintos tipos de Blockchain existentes en el mercado, Hyperledger es privada y puede dotar a los participantes de permisos.

Dado que este proyecto se ha llevado a cabo con Hyperledger, su funcionamiento se explica más profundamente a continuación.

2.2.3 DIFERENCIAS

Propósito: La distinción fundamental entre Hyperledger y Ethereum es la intención para la que están diseñados. Por un lado, Ethereum ejecuta los contratos inteligentes para aplicaciones que se consideran a priori de consumo masivo y descentralizadas. Mientras que Hyperledger aprovecha la tecnología Blockchain para los negocios, ya que está diseñado para admitir implementaciones que brindan altos grados de confidencialidad, resistencia y escalabilidad. Tiene una arquitectura modular y proporciona mucha flexibilidad.

Confidencialidad: Imagínese, por ejemplo, que una persona dirige una pastelería industrial en Blockchain. A través de Hyperledger, es posible que esta persona venda los pasteles a sus

DESCRIPCIÓN DE LAS TECNOLOGÍAS

clientes a un precio con un descuento aplicado al tiempo que mantiene el acuerdo confidencial de todos. Sin embargo, con Ethereum esto no sería posible ya que Ethereum es completamente transparente y toda transacción es visible por todos en la red. Por tanto, Hyperledger permite transacciones confidenciales, lo que da cierta flexibilidad y seguridad. La siguiente tabla muestra una comparativa entre ambas tecnologías en términos de madurez, comodidad de desarrollo, permissionado, tiempo de confirmación de transacciones y privacidad entre nodos.

Solución	Madurez	Comodidad desarrollo	Permissionado	Tiempo de confirmación de transacciones	Privacidad entre nodos
Ethereum	4	3	No	~ 12 segundos	No
Hyperledger	1	2	Si	Instantáneo	Si

Tabla 2: Resumen de la comparativa (menor a mayor puntuación, 1-5)

En redes públicas como Ethereum, cualquiera puede unirse a la red, incluso participantes malintencionados. Para resolver este problema, las redes públicas ofrecen incentivos económicos por actuar en favor de la red (mining) a través de tokens o criptomonedas. Sin embargo, en el caso de Blockchain empresariales, o permissionadas, esta forma de incentivos económicos puede no tener sentido. Esto es porque las redes privadas como Hyperledger reducen el riesgo permitiendo la unión a la red sólo a participantes que son conocidos y previamente aprobados por el resto de los participantes. De tal manera que, si alguno de los participantes descubre que uno de ellos actúa malintencionadamente, siempre pueden denegarle el acceso por medio de un consenso entre los participantes.

Las redes permissionadas están pensadas para redes Blockchain con un reducido número de participantes conocidos, por lo que es posible el uso de algoritmos de consenso mucho más eficientes y rápidos que los de las redes públicas. En el caso de Bitcoin o Ethereum, cuentan con aproximadamente 10.000 y 20.000 nodos respectivamente, y permiten transaccionar entre participantes que son completamente desconocidos y en entornos sin ningún tipo de confianza, característica muy necesaria y potente para muchos otros casos de uso.

Hyperledger, por el contrario, implementa la privacidad de transacciones a través de canales, que son subredes de participantes dentro de la red.

DESCRIPCIÓN DE LAS TECNOLOGÍAS

En definitiva, Hyperledger provee características como registro único, inmutabilidad y robustez a la vez que ofrece las características necesarias de escalabilidad y privacidad en un entorno empresarial.

2.3 *HYPERLEDGER PROJECT*



HYPERLEDGER PROJECT

2.3.1 ARQUITECTURA

El proyecto Hyperledger utiliza la estrategia de Umbrella, que fomenta la reutilización de bloques de construcción comunes dentro de una arquitectura modular. Esto permite una rápida innovación de la tecnología de contabilidad distribuida (DLT), módulos funcionales comunes, y las interfaces entre ellos. Los beneficios de este enfoque modular incluyen la extensibilidad y la flexibilidad, de modo que cualquier componente se puede modificar de forma independiente sin afectar al resto del sistema. En este diseño modular extensible se puede encontrar: Infraestructura, Framework ² y Herramientas. En el siguiente esquema se muestra una descripción a alto nivel [9]:

² Un framework, entorno de trabajo o marco de trabajo es un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular que sirve como referencia, para enfrentar y resolver nuevos problemas de índole similar.

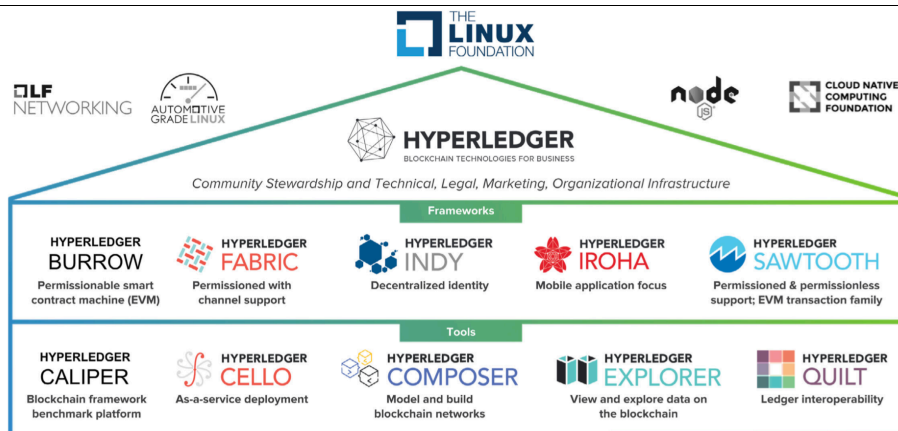


Figura 10: Arquitectura de Hyperledger Project

Por un lado, las herramientas son los proyectos transversales que tratan de apoyar los frameworks implementados. La herramienta que se utiliza en este proyecto es Composer, que se explica a continuación. Por otro lado, estarían los diferentes frameworks que provienen de desarrollos de software que poco a poco se han ido incorporando sobre el paraguas del proyecto Hyperledger. En este proyecto se implementa Fabric, explicado en cierto detalle a continuación.

2.3.2 FRAMEWORKS

2.3.2.1 HYPERLEDGER FABRIC



Hyperledger Fabric es una implementación de framework Blockchain, y uno de los proyectos de Hyperledger hospedados por Linux Foundation.

El libro de contabilidad, los contratos inteligentes, y el consenso entre los miembros se mantienen utilizando los protocolos de Fabric. Está diseñado para casos de uso comercial donde el Blockchain es operado por un conjunto de participantes conocidos e identificados. Esta capacidad se conoce como Blockchain permissionada, explicada anteriormente. Un Blockchain con permisos proporciona una manera de asegurar las interacciones entre un grupo de entidades que se conocen entre sí, tienen intereses comerciales comunes y desean

DESCRIPCIÓN DE LAS TECNOLOGÍAS

gestionar una red descentralizada (en lugar de cambiar la gestión de sus libros de contabilidad a un único interviniente).

Hyperledger Fabric ofrece altos niveles de rendimiento, protección y privacidad en las transacciones. Estas redes constan de los siguientes componentes básicos [17]:

Activos (*assets*): cualquier cosa que tenga valor, estado y propiedad.

Libro contable compartido: donde se registran el estado y la propiedad del activo. Los libros de contabilidad son replicados en un canal (o varios) y almacenados en nodos del par (*peers*).

Contrato inteligente: Los contratos inteligentes de Hyperledger Fabric se implementan en Chaincode. Chaincode es un fragmento de código que está escrito en uno de los idiomas compatibles, como Go o Java. Se instala y crea una instancia a través de un SDK o CLI en una red de nodos pares Hyperledger Fabric, lo que permite la interacción con el libro mayor compartido de esa red. Cuando una aplicación necesita interactuar con el libro mayor, invoca estos contratos mediante el envío de transacciones a la red Fabric. Esta es la razón por la que Chaincode interactúa predominantemente solo con el componente de la base de datos del libro mayor y no el registro histórico de transacciones.

Nodos del par: Los *peers* o pares son los puntos finales de las transacciones para las organizaciones y conforman gran parte de la estructura física de la red. Son mantenidos por miembros (organizaciones) cuyas identidades son conocidas por la red Blockchain. Los pares pueden mantener múltiples libros de contabilidad (tienen uno para cada canal del que son miembros) y respaldan las transacciones. Se pueden crear, iniciar, reconfigurar e incluso borrar. Exponen un conjunto de APIs que permiten a los administradores y aplicaciones interactuar con los servicios que proporcionan.

Canal: Un canal contiene un subconjunto de miembros de la red que desean comunicarse y realizar transacciones en privado. Los *ledgers* son específicos del canal (es decir, cada canal tiene un *ledger* separado). Solamente los pares en un canal pueden ver los activos y las transacciones de su libro mayor. Como resultado, Los canales garantizan la privacidad de los participantes dentro de la red.

Servicio de pedidos: El Servicio de pedidos, generalmente compuesto por varios ordenadores, proporciona el consenso y la solicitud de la transacción. Lo hace agrupando transacciones en bloques, que se añaden posteriormente a la Blockchain.

DESCRIPCIÓN DE LAS TECNOLOGÍAS

Organizaciones: a partir de los pares se construyen diferentes organizaciones que forman parte de la red y que contribuyen con los mismos.

Autoridad de certificación (CA): La autoridad de certificación (CA) identifica todas las entidades en la red: los pares, la ordenación servicio, y los participantes que envían transacciones y acceden a la *ledger*. Estas identidades se proporcionan y protegen mediante el uso de una infraestructura de clave pública (PKI). Los pares utilizan la CA para firmar de forma criptográfica las transacciones y los contratos, mientras que los participantes la usan para demostrar que tienen derecho a acceder a la red.

SDK de Hyperledger Fabric Client: Los SDK de Hyperledger Fabric Client permiten la interacción entre su aplicación cliente y la red de Blockchain. Con soporte para múltiples idiomas, el SDK contiene la API que permite que una aplicación se conecte y acceda a los contratos inteligentes y al libro de contabilidad para el canal del par está activado.

En la siguiente imagen se muestran todos los componentes de una red Hyperledger Fabric.

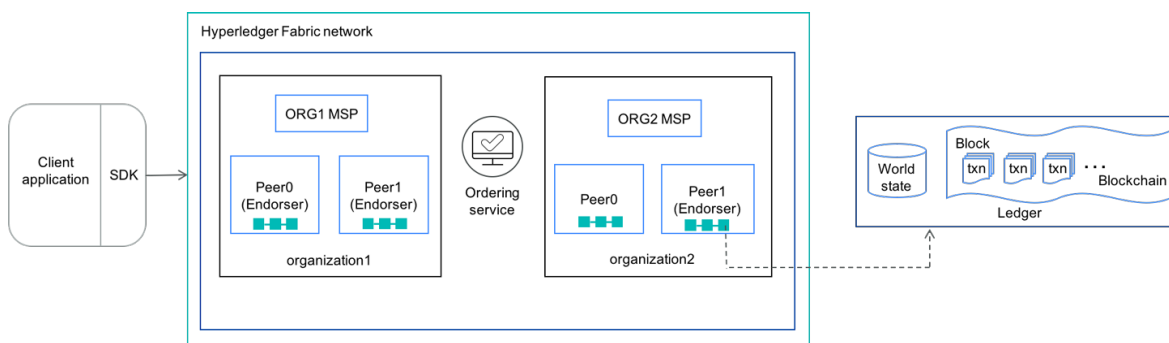


Figura 11: Componentes de una red Hyperledger Fabric

Desde un punto de vista técnico, cada participante dentro de la red de Blockchain tiene un sitio de red desplegado en su organización (o en la infraestructura de la nube), pero no tiene acceso a los sitios de red implementados en otras compañías.

DESCRIPCIÓN DE LAS TECNOLOGÍAS

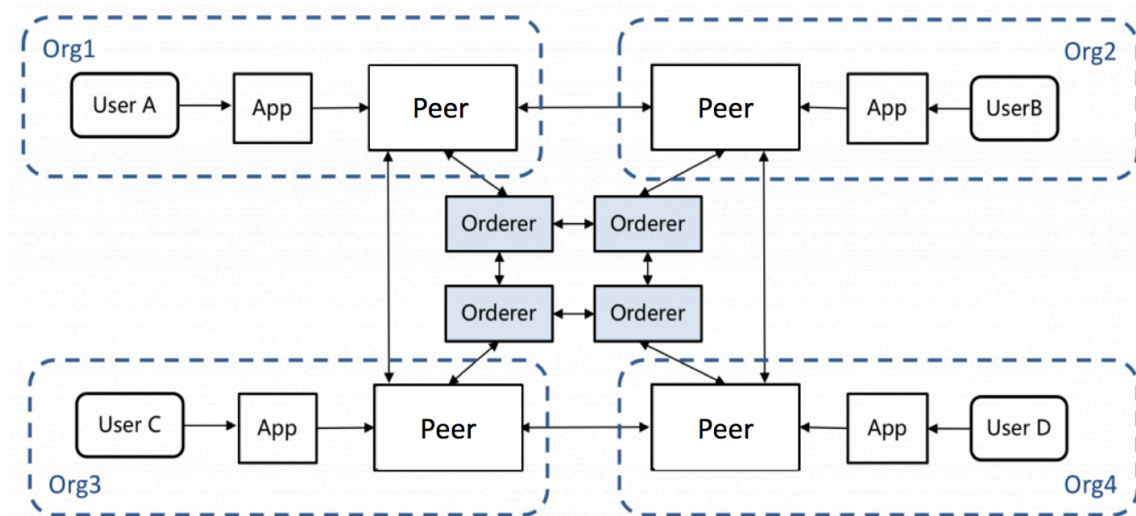


Figura 12: Diagrama de red Hyperledger

Como se puede ver en el diagrama, cada participante tiene:

- Su nodo (*peer*), que garantiza el correcto funcionamiento de la red de bloques y forma parte de Hyperledger Fabric
- Una aplicación que normalmente consiste en al menos un servidor web y una interfaz de usuario
- Servicio de pedidos, que proporciona interacción entre nodos

A continuación, se muestra el flujo de una solicitud de alto nivel de una transacción en la red de Hyperledger Fabric [28].

Para comprender el flujo, hay que asumir que un canal está configurado y funcionando, y que el usuario de la aplicación se ha registrado e inscrito en la Autoridad de certificación (CA) de la organización y ha recibido el material criptográfico necesario, que se utiliza para autenticarse en la red.

En primer lugar, el chaincode se instala en los pares y se crea una instancia en el canal. Éste contiene una lógica que define un conjunto de instrucciones de transacción. Por otro lado, tanto peer A como peer B deben respaldar cualquier transacción.

DESCRIPCIÓN DE LAS TECNOLOGÍAS

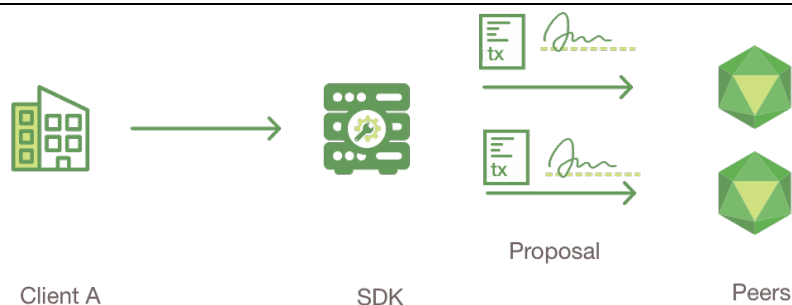


Figura 13: Flujo de una solicitud de alto nivel en Hyperledger Fabric

1. El cliente A envía una solicitud para realizar una transacción. Esta solicitud se dirige a peer A y peer B (pues ambos han de aprobar la transacción), que son respectivamente representantes del Cliente A y del Cliente B. A continuación, se construye la propuesta de transacción: Una aplicación aprovecha un SDK compatible (Node, Java, Python) y utiliza una de las API disponibles para generar la propuesta de transacción. La propuesta es una solicitud para invocar una función de chaincode con ciertos parámetros de entrada, con la intención de leer y / o actualizar el libro mayor. El SDK sirve como un complemento para empaquetar la propuesta de transacción en el formato correctamente diseñado y toma las credenciales criptográficas del usuario para producir una firma única para esta propuesta de transacción.

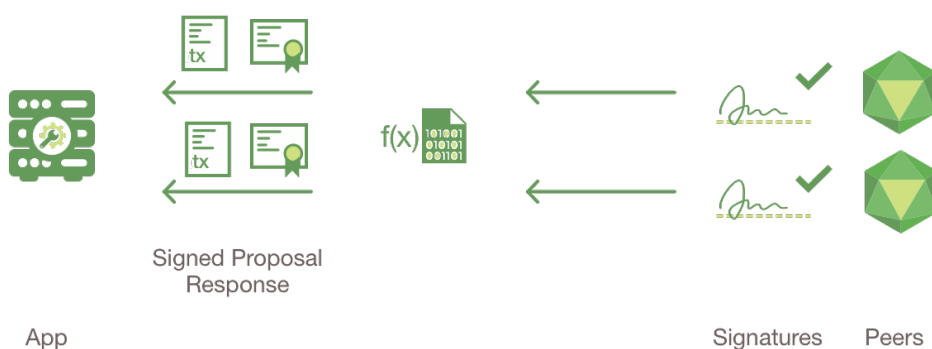


Figura 14: Flujo de una solicitud de alto nivel en Hyperledger Fabric (Paso 1)

2. Los homólogos que respaldan verifican (1) que la propuesta de transacción está bien formada, (2) no se ha enviado ya en el pasado (protección contra ataques de repetición), (3) la firma es válida (usando el MSP) y (4) que el remitente (el Cliente A, en el ejemplo) está debidamente autorizado para realizar la operación propuesta en ese canal. Los

DESCRIPCIÓN DE LAS TECNOLOGÍAS

homólogos que respaldan toman las entradas de la propuesta de transacción como argumentos de la función del chaincode invocado. Luego, el chaincode se ejecuta en la base de datos del estado actual para generar resultados de transacción que incluyen un valor de respuesta, un conjunto de lectura y un conjunto de escritura (es decir, pares clave / valor que representan un activo para crear o actualizar). No se realizan actualizaciones en el libro mayor en este momento. El conjunto de estos valores, junto con la firma del homólogo que avala, se devuelve como una "respuesta de propuesta" al SDK, que analiza la carga útil para que la aplicación consuma.



Figura 15: Flujo de una solicitud de alto nivel en Hyperledger Fabric (Paso 2)

3. La aplicación verifica las firmas de los homólogos que respaldan y compara las respuestas de la propuesta para determinar si las respuestas de la propuesta son las mismas. Si solo se consulta el código de cadena en el libro mayor, la aplicación inspeccionará la respuesta de la consulta y, por lo general, no enviará la transacción al servicio de pedidos. Si la aplicación del cliente tiene la intención de enviar la transacción al servicio de pedido para actualizar el libro mayor, la aplicación determina si la política de aprobación especificada se ha cumplido antes de enviar. La arquitectura es tal que incluso si una aplicación elige no inspeccionar las respuestas o envía una transacción sin aprobación, la política de aprobación todavía será aplicada por sus pares y se mantendrá en la fase de validación de confirmación.



Figura 16: Flujo de una solicitud de alto nivel en Hyperledger Fabric (Paso 3)

4. La aplicación "difunde" la propuesta de transacción y la respuesta dentro de un "mensaje de transacción" al servicio de pedido. La transacción contendrá los conjuntos de lectura / escritura, las firmas de los homólogos que respaldan y la ID del canal. El servicio de pedidos no necesita inspeccionar todo el contenido de una transacción para realizar su operación, simplemente recibe transacciones de todos los canales en la red, las ordena cronológicamente por canal y crea bloques de transacciones por canal.

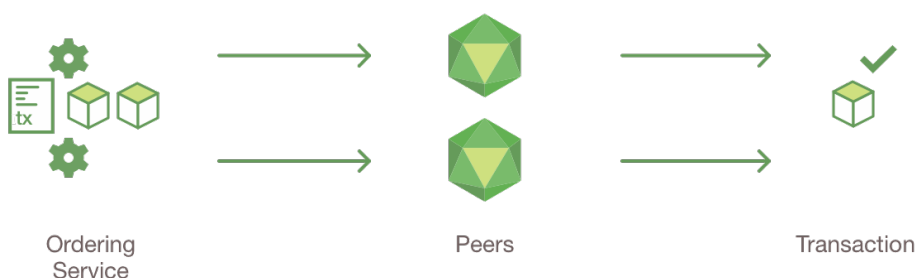


Figura 17: Flujo de una solicitud de alto nivel en Hyperledger Fabric (Paso 4)

5. Los bloques de transacciones se "entregan" a todos los pares en el canal. Las transacciones dentro del bloque se validan para garantizar que se cumpla la política de aprobación y para garantizar que no se hayan producido cambios en el estado del libro mayor para las variables del conjunto de lectura desde que la ejecución de la transacción generó el conjunto de lectura. Las transacciones en el bloque se etiquetan como válidas o no válidas.

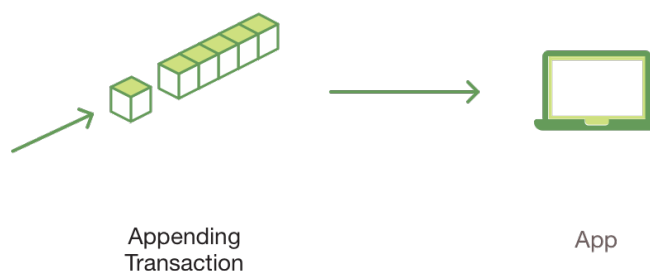


Figura 18: Flujo de una solicitud de alto nivel en Hyperledger Fabric (Paso 5)

6. Libro mayor actualizado

Cada interlocutor agrega el bloque a la cadena del canal y, para cada transacción válida, los conjuntos de escritura se confirman en la base de datos del estado actual. Se emite un evento para notificar a la aplicación cliente que la transacción (invocación) se ha agregado de manera inmutable a la cadena, así como una notificación de si la transacción fue validada o no.

2.3.2.2 HYPERLEDGER SAWTOOTH



Hyperledger Sawtooth es una plataforma que compila, e implementa libros distribuidos. Esta plataforma implementa un algoritmo de consenso orientado principalmente a grandes poblaciones que distribuyen un mínimo de recursos [10].

2.3.2.3 HYPERLEDGER IROHA



Hyperledger Iroha es un framework de negocios que tiene un diseño simple, moderno y orientado a C++ y aplicaciones móviles. Está basado en Hyperledger Fabric e implementa,

DESCRIPCIÓN DE LAS TECNOLOGÍAS

además, un novedoso algoritmo de consenso tolerante a fallas bizantinas denominado YAC (*Yet Another Consensus*) de alto rendimiento, escalabilidad y permite realizar transacciones con baja latencia.

Permite a los usuarios crear, transferir activos, etc. anulando la necesidad de redactar *smartcontracts* difíciles de probar y complicados, y permitiendo además a los desarrolladores completar tareas simples de manera rápida y con bajo riesgo [11].

2.3.2.4 HYPERLEDGER BURROW



Hyperledger Burrow fue en un principio contribuido por Monax y copatrocinado por Intel. Proporciona un cliente modular de Blockchain con un intérprete de *smartcontracts* autorizado parcialmente y desarrollado según la especificación de la Máquina Virtual Ethereum (EVM) [12].

2.3.2.5 HYPERLEDGER INDY



Indy ofrece una serie de herramientas, bibliotecas y componentes que son reutilizables para la implementación de identidades digitales. Se encuentra en plena etapa de desarrollo, por lo que aún están en espera ciertos detalles acerca de su funcionamiento y desempeño [13].

2.3.3 HERRAMIENTAS

2.3.3.1 HYPERLEDGER QUILT



Esta herramienta dispone de interoperabilidad entre los distintos sistemas de Blockchain mediante ILP. ILP es un protocolo de pagos que está diseñado para la transferencia de valores a través de bases de datos. Se encuentra aún en etapa de desarrollo [14].

2.3.3.2 HYPERLEDGER EXPLORER



Esta herramienta permite ver, ejecutar e implementar bloques y datos asociados sobre una información almacenadas en red o en el libro mayor [15].

2.3.3.3 HYPERLEDGER CELLO



Esta herramienta lleva a cabo el modelo “As-A-Service” y “On-demand” para reducir los esfuerzos requeridos para la creación y administración de la cadena de bloques de manera automática [16].

2.3.3.4 HYPERLEDGER COMPOSER



Esta herramienta es la utilizada en este proyecto por lo que se va a explicar con mayor profundidad unos puntos más adelante.

Composer es un conjunto de herramientas y un marco de desarrollo de código abierto que acelera el tiempo que conlleva programar una aplicación Blockchain. En lugar de desarrollar contratos inteligentes desde cero, Composer proporciona una capa de conveniencia y abstracciones de nivel empresarial para implementar *smartcontrats* sobre Fabric. Además, facilita la conexión a la red de negocio desde una aplicación web o móvil [17].

Composer ayuda a definir la red de negocio utilizando un lenguaje de modelado personalizado y un pequeño conjunto de primitivas, y luego desplegarla a Hyperledger Fabric. La red de negocios está formada por participantes, activos y transacciones. También se incluyen listas de control de acceso que restringen el acceso a transacciones o datos.

Están basadas en el lenguaje de programación JavaScript y pueden crear rápidamente chaincode y aplicaciones Blockchain. En vez de escribir un chaincode en Golang, lo que permite Hyperledger es generar un archivo de arquitectura de red empresarial (.bna).

1. Arquitectura de red empresarial
2. Hyperledger Composer Playground
3. Soporte para REST API

Composer requiere de una serie de archivos; un archivo JavaScript (.js), un archivo de control de acceso (.acl) y un archivo de consultas (.qry).

DESCRIPCIÓN DE LAS TECNOLOGÍAS

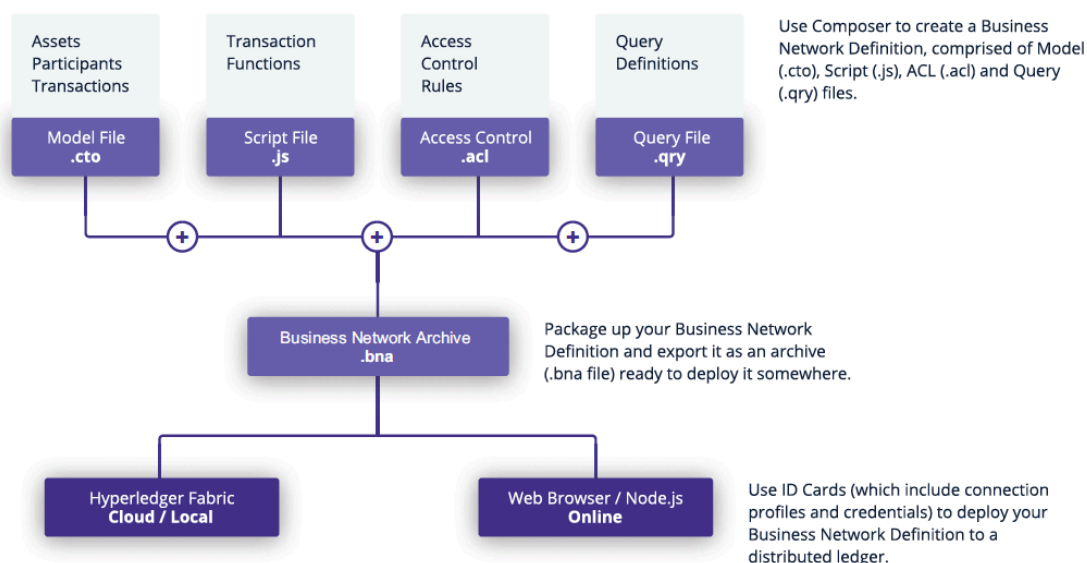


Figura 19: Esquema de los archivos de programación del Back End

Archivos de modelo de red (.cto): en estos archivos se van a encontrar lo que llamamos el modelo de nuestro *Back End*. En él se declararán los campos de un participante, de un activo, las transacciones que se van a llevar a cabo y los eventos que se pueden generar. Se desarrolla en el lenguaje propio diseñado por IBM.

Archivos de javascript (.js): estos ficheros llevan a cabo toda la lógica de programación de las transacciones, donde se relacionan a los participantes y los activos que intervienen en ellas y los cambios o acciones que han de realizarse.

Archivos con extensión (.acl): la gran particularidad de Hyperledger es precisamente el hecho de ser una red permissionada. Este archivo contiene las reglas de control de acceso que definen los derechos de los diferentes participantes en la red.

Archivos de consultas (.qry): este archivo define las consultas que se pueden hacer o ejecutar en la red empresarial.

Como se ha mencionado antes, con Hyperledger Composer se puede crear una definición de red empresarial empaquetando estos archivos y exportándolos como un único archivo (.bna). Business Network Archive (.bna) se puede implementar en una red existente Hyperledger Fabric y contiene una serie de funciones ejecutables que se podría considerar como un contacto inteligente escrito en JavaScript.

2.3.3.5 Hyperledger Composer Playground

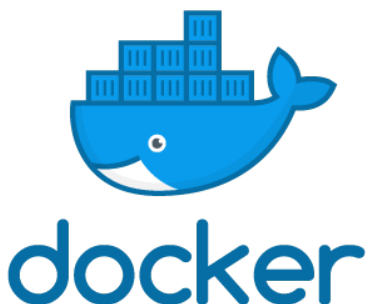
Hyperledger Composer también dispone de una interfaz de usuario web que se denomina Playground. Esta interfaz permite crear modelos y probar la red empresarial. Utiliza almacenamiento local del navegador, simulando el estado de la red de Blockchain. Es con esta interfaz con la que se ha ido testeando y compilando el proyecto.

2.3.3.6 Soporte para API REST

Además del interfaz gráfico con el que puedes crear modelos y probar tu red, Hyperledger Composer dispone también de un conector *loopback* para generar REST APIs basadas en la definición empresarial. Estas REST APIs pueden ser consumidas fácilmente por las aplicaciones cliente y se pueden integrar con aplicaciones que no son de Blockchain.

2.4 OTRAS TECNOLOGÍAS USADAS

2.4.1 DOCKER



Docker es un proyecto *opensource* que permite que una aplicación y sus dependencias se empaqueten en un contenedor. La virtualización que está basada en contenedores aísla las aplicaciones entre sí en un sistema operativo que es compartido. Puede ejecutarse desde cualquier ordenador que tenga Docker instalado. Con Docker se elimina el problema de incompatibilidad de sistemas operativos o versiones. [18]

2.4.2 ANGULAR APP



Angular App, comúnmente llamado Angular.js es un *framework* de JavaScript *opensource* que es mantenido por Google y que se utiliza para crear y mantener aplicaciones web en una sola página. El objetivo de esta plataforma es aumentar las aplicaciones basadas en navegador con MVC (Modelo Vista Controlador) intentando que las pruebas y el desarrollo de estas sea más sencillo. Se programa mediante TypeScript, JavaScript, HTML, CSS y además puede combinar con el entorno en tiempo de ejecución de Node.js, Express.js y la base de datos MongoDB [19].

2.4.3 GIT



Git es un software para el control de versiones. Almacena y modela la información como un conjunto de instantáneas de un mini sistema de archivos. Cada vez que se realiza un cambio o se guarda el estado de tu proyecto en Git, éste hace una foto del aspecto de tus archivos en el momento y guarda una referencia instantánea. Además, Git posee un repositorio de proyectos y código conocido como GitHub donde se puede acceder fácilmente para encontrar código ya desarrollado, subir código y lanzar preguntas [20].

2.4.4 NODEJS



NodeJS es un entorno de ejecución para JavaScript que está orientado a eventos asíncronos, lo que le hace más ligero y eficiente. Está diseñado para generar aplicaciones web de forma altamente optimizada. Uno de sus puntos fuertes es su Node Package Manager o npm, que es una especie de gestor que da acceso a un conjunto de librerías extenso de código abierto [21].

2.4.5 IONIC



Ionic es una herramienta *opensource* para el desarrollo de aplicaciones híbridas basadas en HTML, CSS, JavaScript y TypeScript. Está optimizado con Angular. Se trata de una herramienta híbrida pues combina aspectos de aplicaciones nativas y aplicaciones web [22].

Capítulo 3. ESTADO DE LA CUESTIÓN

En el presente capítulo se realiza un somero repaso tanto sobre el estado en el que actualmente se encuentran el Internet of Things (IoT) y Blockchain como a los principales desafíos a los que se enfrentarán en un futuro ambas tecnologías.

3.1 *INTERNET OF THINGS (IoT)*

3.1.1 RETROSPECTIVA

El Internet de las cosas, comúnmente conocido como *Internet of Things* (IoT), es una tecnología que ha acaparado la atención de la comunidad empresarial, superando a menudo las expectativas puestas en ella. Dada su omnipresencia, parece ser un concepto ya muy manido, aunque sin embargo la realidad es que es bastante reciente. Por ello, y con el objetivo básico de entender en su totalidad el origen y el potencial alcance y aplicabilidad del IoT, resulta conveniente echar un rápido vistazo a su pasado y analizar con cierta perspectiva cómo ha venido evolucionando hasta el momento actual.

En el año 1874 se realiza el primer experimento de telemetría de la historia llevado a cabo por científicos franceses. En el año 1927 Nikola Tesla conformó la base de las comunicaciones inalámbricas y de radio. Si bien hay que avanzar un poco más en la historia para identificar el origen de lo que hoy en día se conoce como Internet. En efecto, en 1969 se establece la primera conexión entre computadoras, sobre una red denominada ARPANET, entre tres universidades de California. ARPANET nace como una red descentralizada que con el tiempo fue interconectando más y más computadores entre sí. Sobre esa base han ido surgiendo nuevas redes con protocolos y mecanismos de comunicación diversos, como lo fueron Ethernet, ALOHA, etc.

Es en el año 1990 cuando se implementa con éxito la primera comunicación entre un cliente HTTP y un servidor en Internet utilizando el protocolo de transporte TCP/IP. La verdadera revolución vino de la mano de la popularización de la conectividad inalámbrica (inicialmente celular, posteriormente edge, 3G, WiFi, 4G y más recientemente el 5G). Con la creación de la primera página web, un año más tarde, la evolución de la informática y las comunicaciones

ESTADO DE LA CUESTIÓN

ha crecido exponencialmente, cambiando pautas de comportamiento social y, en determinados casos, disrupción en ciertos modelos de negocio.

El término *Internet of Things* (IoT), fue usado por primera vez de forma pública en 2009 por Kevin Ashton, en aquel entonces profesor del MIT. Ashton acuñó públicamente el término en el RFID journal: “Si tuviésemos ordenadores que fuesen capaces de saber todo lo que pudiese saberse de cualquier cosa, usando datos recolectados sin intervención humana, seríamos capaces de hacer un seguimiento detallado de todo, y así poder reducir de forma importante los costes y malos usos. Se sabría cuando las cosas necesitan ser reparadas, cambiadas o recuperadas, incluso si están frescas o pasadas de fecha. El Internet de las Cosas tiene el potencial de cambiar el mundo como ya lo hizo Internet. O incluso más.” [23] Desde su origen en la telemetría, el IoT continúa en pleno desarrollo y crecimiento exponencial, y se siguen desarrollando nuevos proyectos, redes y dispositivos.

3.1.2 DESAFÍOS IOT

IoT es una tecnología revolucionaria en el intercambio de datos y el control de dispositivos para diferentes aplicaciones. La firma EMC, empresa desarrolladora de software y sistemas para la administración y almacenamiento de datos e información, ha estimado que en 2020 habrá más de 30 billones de dispositivos electrónicos conectados a Internet. [24]

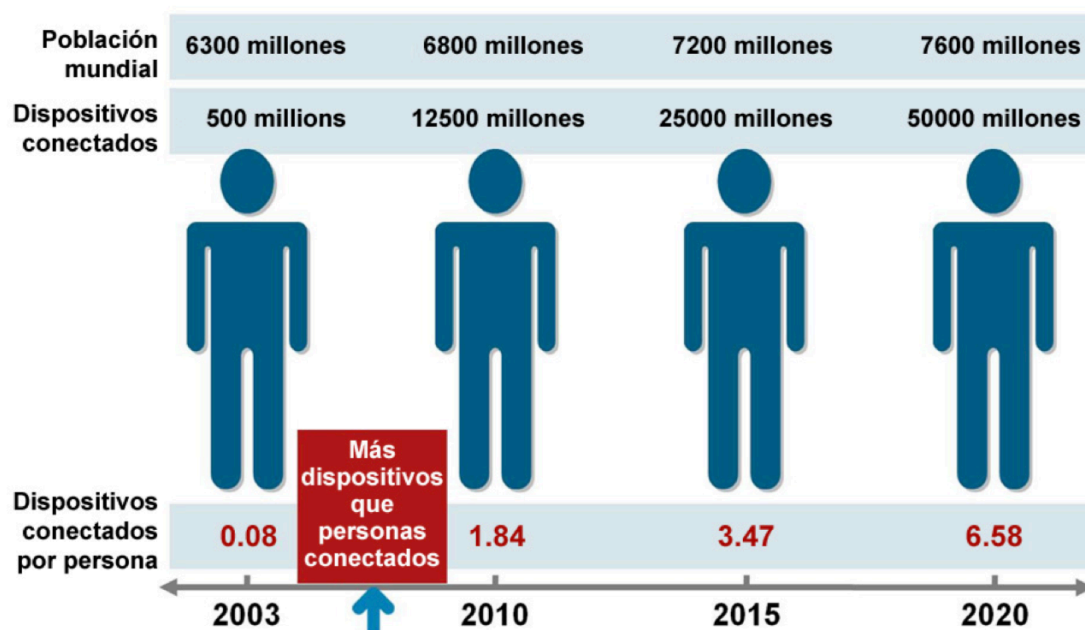


Figura 20: Evolución del IoT

Como se puede observar en la Figura 11, la evolución del número de usuarios conectados a Internet en el mundo sigue una progresión geométrica. En el año 2003 la cifra total de usuarios apenas alcanzaba los 500 millones sobre una población total de 6300 millones de personas, mientras que en el año 2020 se estima que esta cifra alcanzará un total de 7,6 millones de personas conectadas a Internet. [25]. Además, el número de dispositivos conectados a internet ha sobrepasado ampliamente al de personas y sigue creciendo de manera exponencial, otorgando a los dispositivos capacidades de interpretar, comunicar y tener su propia “voz digital” en el mundo. Esto presenta sin duda grandes desafíos, y también oportunidades, tanto a ingenieros como a investigadores:

Privacidad y seguridad

Además del claro crecimiento de dispositivos conectados a Internet, el tamaño de la comunicación y los datos que se procesan también aumenta exponencialmente. En muchos casos, los clientes y gobiernos exigen un historial de los datos que se procesan. Esto crea muchos problemas no deseados en la vida rutinaria, ya que estos datos pueden ser tan sensibles como los datos personales. En consecuencia, los problemas de privacidad y seguridad no crecen linealmente con el número de dispositivos conectados, sino que también muestran una tendencia exponencial.

Coste

El coste de mantenimiento sube, lo que en muchos casos deriva en un decrecimiento de las medidas de redundancia para intentar reducir esos incrementos de coste, lo que redunda en un notable aumento de los riesgos asociados a la pérdida de información y datos. Se está demostrando que los costos asociados a soluciones de IoT son muy altos, atendiendo al gran número de mensajes (costo de comunicaciones), los datos generados por los dispositivos (costos de almacenamiento) y a procesos analíticos (costos de los servidores). [26]

Cuello de botella

Las plataformas de IoT que existen actualmente son centralizadas y basadas en datos, lo cual inevitablemente genera grandes cuellos de botella en el escalado de las soluciones para cantidades grandes de dispositivos. En conclusión, hay un serio problema arquitectural que requiere de una solución estructural.

Falta de estándares

Por otro lado, existe una falta de estándares y uniformidad en los datos ya que no hay ninguna plataforma única para conectar los dispositivos de todos los fabricantes.

Resulta evidente canalizar estos millones de datos sensibles sobre una infraestructura que sea segura y que mantenga la autenticidad e integridad de la información distribuida, más allá de resolver los propios desafíos que tiene la tecnología. Una manera de asegurarlo es desarrollando una red IoT descentralizada con Blockchain.

3.1.3 REDES DE IOT DESCENTRALIZADAS

Las redes IoT descentralizadas, distribuidas y basadas en estándares abiertos pueden solucionar muchos de los desafíos y problemas mencionados anteriormente (seguridad, integridad, escalabilidad, costo, etc.).

Por ejemplo, los dispositivos IoT conectados se podrían comunicar directamente utilizando Blockchain, a través de unos libros contables distribuidos y descentralizados. Del mismo modo, los contratos inteligentes podrían usar estos datos para actualizarlos y validarlos y finalmente entregarlos a los participantes interesados de la red. Con todo esto lo que se conseguirá es por tanto reducir la necesidad de que personas monitoreen y realicen acciones, además de aumentar la confianza de los datos generados al estar en un entorno seguro como

ESTADO DE LA CUESTIÓN

es el de Blockchain. Blockchain implementa mecanismos específicos para el consenso que eliminan las acciones de los dispositivos comprometidos.

Proporcionar una solución a estos desafíos de escalabilidad, seguridad y descentralización es un requisito previo y necesario en cualquier implementación de red IoT basada en Blockchain.

3.1.4 CASOS DE USO DE IoT

Las aplicaciones del IoT son incontables, alcanzando su límite en la propia imaginación. Por simplicidad y manejabilidad se pueden destacar tres sectores básicos: industrial, mercado doméstico y ciudades inteligentes.

Sector industrial

Este sector cubre una amplia gama de soluciones de negocio, servicios y productos, donde el IoT se puede aplicar de numerosas y diversas maneras. Por ejemplo, en hostelería, servicios de comida rápida donde camareros son capaces de localizar la mesa o adónde va la comida; en comercios para el público, particularmente en centros comerciales, donde se puede estimar el tiempo que una persona pasa comprando o el tiempo que pasa entre llegada de los clientes... También aplicable para flotas de vehículo para logística, es decir, a modo de ejemplo conocer la localización de un paquete en tiempo real. Incluso en agricultura y ganadería facilitando a los trabajadores poder predecir y cuantificar cada cosecha antes de recogerla.

Uso doméstico

Un claro ejemplo es el analizador de piscinas inteligentes, que envía toda la información relevante para controlar la calidad del agua; productos para minimizar la interfaz con los humanos, siendo la voz la más utilizada; neveras inteligentes... También usado ampliamente en jardinería con regadíos automáticos y automatización en la plantación. Y en salud, un tema que concierne a todos, donde se pueden monitorizar pacientes y seguir sus constantes vitales a distancia.

Ciudades inteligentes

Este, sin duda, representa un entorno muy claro para la introducción de estas aplicaciones. Particularmente enfocados a la gestión de suministros (ejemplo, contadores inteligentes de

ESTADO DE LA CUESTIÓN

agua), gestión ambiental (control de calidad del aire), gestión del tráfico (envío de la posición y velocidad a través del móvil para así poder visualizar la información de las carreteras) ... A pesar de los muchos desafíos, es notorio que el IoT cuenta ya en la actualidad con infinidad de aplicaciones.

3.2 BLOCKCHAIN

3.2.1 RETROSPECTIVA

Como se ha descrito anteriormente, Blockchain consiste en una base de datos distribuida y descentralizada, que registra toda transacción digital entre los participantes y que comparte con todos los miembros de la red, la mayoría de los cuales deben verificar esa transacción mediante métodos de consenso.

La idea detrás de esta tecnología se describió en 1991, cuando Stuart Haber y W. Scott Stornetta (científicos e investigadores), introdujeron una solución para los documentos digitales con sello de tiempo para que no pudieran ser modificados o manipulados. Este sistema se implementó mediante una cadena de bloques con seguridad criptográfica para almacenar los documentos con sello de tiempo. Sin embargo, no es hasta 1992 cuando se incorporan al diseño los árboles Merkle³, lo que permitió que varios documentos se reunieran en un solo bloque. Esta tecnología caducó en 2004 al no utilizarse, y cuatro años más tarde nace la hoy conocida como red Bitcoin.

Satoshi Nakamoto (nombre de una persona o, tal vez, de un grupo bajo este seudónimo) publicó a finales de 2008 en una lista restringida de correos electrónicos un libro blanco (un artículo académico) que introdujo un nuevo sistema de efectivo electrónico descentralizado entre pares, denominado Bitcoin. En este artículo, el autor explicaba la idea de minar una cantidad limitada de la moneda a través de una red persona a persona (peer-to-peer) y que no dependiera de un gobierno central o un banco. En este documento además se explicaba que cuando la gente empezara a negociar en bitcoins, el gobierno no podría intervenir, dada la naturaleza de su tecnología.

³ Un árbol de Merkle o árbol hash es una estructura de datos en árbol, binario o no, en el que cada nodo que no es una hoja está etiquetado con el hash de la concatenación de las etiquetas o valores (para nodos hoja) de sus nodos hijo. Son una generalización de las listas hash y las cadenas hash.

ESTADO DE LA CUESTIÓN

Esta idea no era innovadora, pues desde el surgimiento de Internet y el movimiento Cyberpunk, en los años 1990, se defendía la idea del dinero “libre”. Sin embargo, fue inviable hasta que Nakamoto consiguió algo llamado doble gasto, cuando las transacciones se pueden copiar y gastar dos veces. Esto es porque Bitcoin tiene un protocolo de confianza donde cada transacción es auditada y confirmada por el próximo bloque abierto a toda la comunidad de mineros. Con el paso del tiempo, el Bitcoin dejaba de ser una red limitada de mineros para convertirse en un fenómeno global.

La reputación del Bitcoin no fue de las mejores, pues dado el anonimato de cada participante podría utilizarse para financiación del tráfico de drogas, para lavar dinero, incluso para solicitar rescates por secuestros físicos. A pesar de ello, fue exactamente el cibercrime que arrojó luz y fama a la criptomoneda. [27]

En 2014 muchas iniciativas y emprendedores vieron en el Bitcoin una oportunidad para lanzar nuevos negocios, desde casas de cambio de monedas, hasta compraventa de PCs. Pero en ese mismo año la japonesa Mt. Gox cerró las puertas tras sufrir un ataque hacker que habría robado 850 mil Bitcoins.

La red de Bitcoin se desarrolla por programadores que se dedican a evolucionarla y mejorarla. En julio de 2015, tras varios prototipos, se pone en marcha la primera versión de la plataforma Ethereum con un plan de evolución para dotarle de todas las funcionalidades previstas.

Pero la historia y evolución de Blockchain no se detienen en Ethereum y Bitcoin, sino que en los últimos años han surgido una serie de proyectos aprovechando todas las capacidades de la tecnología. Estos nuevos proyectos han tratado de abordar algunas de las deficiencias de ambas redes, además de ofrecer nuevas características que aprovechan las capacidades de Blockchain.

Algunas de estas nuevas aplicaciones incluyen NEO e IOTA, redes públicas de Blockchain, por lo que cualquier persona puede acceder a los contenidos de una red. Sin embargo, con la evolución de esta tecnología, varias empresas han comenzado a adoptar Blockchains privadas e híbridas como Hyperledger, EOS.IO, Quorum...

El futuro de esta tecnología se desconoce, pero muchos gobiernos y empresas están invirtiendo enormes cantidades de dinero, buscando estimular las innovaciones y sus aplicaciones.

3.2.2 CASOS DE USO DE BLOCKCHAIN

Blockchain ofrece una amplia gama de posibles casos de uso y aplicaciones, ya que facilita enormemente la escalabilidad y el crecimiento, permitiendo a muchos participantes disfrutar de los beneficios que aporta la cadena de bloques.

Financiamiento comercial junto con la cadena de suministro

Instituciones financieras, compañías de logística e IBM están trabajando juntos en un nuevo sistema global para la financiación comercial utilizando la tecnología Blockchain que está diseñada para rastrear bienes a través de la cadena de suministro. La resolución de pagos es un proceso engorroso y en gran parte basado en papel, que deja a las empresas esperando semanas o incluso más tiempo antes de que se les pague por sus bienes. Dado que todos los miembros de una red Blockchain tienen copia del libro mayor y los contratos inteligentes, los pagos se pueden activar automáticamente a la entrega, convirtiendo las disputas comunes en cosas del pasado.

Salud

Blockchain reduce las barreras que están involucradas en el intercambio de información entre hospitales, proveedores de servicios médicos, departamentos de salud pública y centros para el Control De Enfermedades (CDE), permitiendo a las organizaciones intercambiar datos de los pacientes de manera rápida, segura, transparente y legalmente compatible. Casos de uso de la salud a menudo convergen en los casos de uso de identidad (explicados con más detalle a continuación), lo que permite a los individuos aprobar fácilmente el intercambio de información pertinente sin compartir información no esencial o innecesaria. Por ejemplo, si se requiere de una radiografía, las partes interesadas en el resultado podrían incluir el proveedor de seguros, un médico general, el técnico de rayos X y el individuo. Sin embargo, el técnico de rayos X no necesita saber que el seguro es proporcionado por el empleador. Todo lo que el técnico debe saber es que el cliente es válido y está cubiertos por el procedimiento.

Identidad

Blockchain proporciona el cifrado de datos en el libro mayor e información detallada con controles que determinan quién puede acceder a los datos. Permite poder decidir exactamente quién puede ver cada bit de los datos que posee. Por ejemplo, muchas personas

ESTADO DE LA CUESTIÓN

llevan un elemento de identificación como el DNI, que contiene información como nombre, dirección, y otros datos. A veces, se proporciona esta identificación para demostrar la edad a alguien, y sólo necesitan saber una parte de esta información: por ejemplo, la fecha de cumpleaños. Manteniendo la identidad en una cadena de bloques se permite delegar el acceso de la información pertinente cuando sea necesario, mientras el resto se mantiene privado, reduciendo así el riesgo de robo o mal uso.

Compensación de fondos

Los sistemas de liquidación de efectivo en múltiples monedas son complejos, involucrando una cámara de compensación y muchos sistemas dispares que deben interactuar para detectar el intercambio de tarifas y liquidar transferencias de divisas. El Blockchain elimina la necesidad de cámaras de compensación.

3.3 BLOCKCHAIN E IOT

3.3.1 CASOS DE USO DE IOT CON BLOCKCHAIN

La combinación de IoT y Blockchain está creando múltiples y novedosas posibilidades para el uso de dispositivos inteligentes en la gestión de activos. Cuando estos dispositivos envían información o datos, éstos se integran en una Blockchain de los participantes que brinda estos datos de manera confidencial y en tiempo real.

Con los *smartcontracts*, se pueden especificar modos de funcionamiento que permitan monitorizar y controlar parámetros como la temperatura, la calidad, el precio, la localización, la velocidad, etc

A continuación, se explican algunos de los casos de uso de IoT implementados sobre una infraestructura Blockchain para diferentes industrias.

Cadena de suministro

La cadena de suministro presenta problemas de visibilidad, de optimización y de demanda. Blockchain puede ayudar a resolver estos problemas, ya que puede garantizar el control de acceso adecuado para los datos que los participantes comparten. Además, se podrán optimizar los datos que se envían en tiempo real de una forma más eficiente. Por ejemplo, en el seguimiento de artículos alimenticios, desde el importador hasta el transportista y el

ESTADO DE LA CUESTIÓN

envío, o la identificación de la contaminación y el desperdicio de alimentos. Estos dispositivos IoT envían flujos de datos como por ejemplo de temperatura, de ubicación, etc.

Automoción

La industria de automoción es una de las mayores beneficiadas en este sector. Por ejemplo, Toyota ha comenzado con la implementación de Blockchains para realizar el seguimiento de los millones de piezas que pasan por los distintos países, fábricas y proveedores, entre otros.

Energía y servicios públicos

Blockchain puede transformar la industria energética, construyendo, por ejemplo, redes en malla de dispositivos IoT que supervisen redes eléctricas y corrijan los errores que vayan surgiendo de manera rápida y eficiente.

Salud

El uso de Blockchain facilitará la seguridad de los datos privados de los pacientes, obtenidos de la supervisión médica. Estos datos se almacenarían de forma segura en una Blockchain permissionada, donde sólo aquellos participantes con permisos puedan recibir en tiempo real estos datos.

Automatización

Los dispositivos IoT se están utilizando sobre todo en ciudades inteligentes, o edificios inteligentes para mejorar la seguridad y monitorización de sus residentes. Pueden proteger los dispositivos y datos que recopilan de ellos en una Blockchain que automatice el proceso de pago garantizando la calidad del servicio.

Capítulo 4. DEFINICIÓN DEL TRABAJO

4.1 JUSTIFICACIÓN

Actualmente, la sociedad goza del privilegio de asistir a un período de gran innovación y transformación. Empresas y organismos de todo el mundo están dedicando grandes cantidades de recursos (personas, tiempo y dinero) en investigación y desarrollo de proyectos sobre Blockchain. Las posibilidades de esta tecnología son prácticamente infinitas y afectan a todos los ámbitos de negocio que se puedan imaginar. Prueba de ello son los gigantescos ecosistemas de consorcios y empresas concebidos para generar tanto soluciones horizontales como soluciones verticales en los diferentes sectores.

IoT, por otro lado, representa mucho más que la simple conexión de dispositivos inteligentes, dotando a éstos de una potencial inteligencia distribuida que permita aportar un valor añadido adicional en todos los niveles de la cadena de valor y suministro.

Tomando como base esta realidad, se ha considerado oportuno desarrollar el presente proyecto, donde además de realizar un riguroso estudio de ambas tecnologías, Blockchain e IoT, se desarrolla un caso de uso orientado a empresas que pretende dar solución a problemas reales existentes hoy en día en el mercado, implementando una solución de cadena de suministro sobre una infraestructura Hyperledger.

4.2 ELECCIÓN DE LA TECNOLOGÍA HYPERLEDGER

Actualmente la tecnología Blockchain cuenta con un futuro muy prometedor en el mundo empresarial dado su enorme potencial y a pesar de existir determinadas incógnitas que todavía no se han resuelto. Blockchain representa un nuevo paradigma tecnológico, haciendo posible que la información que se transmite (entre otros, datos sensibles) sea descentralizada tanto en almacenamiento como en confianza, además de proporcionar inmutabilidad y seguridad en los datos y en las propias transacciones.

Se ha decidido optar por Hyperledger, dada las ventajas que ofrece al mundo empresarial. En primer lugar, por su escalabilidad, personalización y confianza. La estructura de Hyperledger es modular, por lo que resulta totalmente escalable. Además, dado que no todos los nodos participan en las transacciones, puesto que dotan de ciertos permisos elegidos a

DEFINICIÓN DEL TRABAJO

priori, el nivel de confianza y la verificación están personalizados y sólo las personas directamente implicadas en una operación específica obtienen información.

Por otro lado, Hyperledger Fabric trabaja con SDKs y tiene una estructura modular Plug-in. Gracias a ello, las empresas disponen de un marco muy versátil y adecuado para todas las industrias, pues con ello pueden desarrollar una red acorde a sus necesidades e ideas, e incluso hacer que determinadas operaciones sean visibles para un grupo elegido de personas. En conclusión, por su modularidad, versatilidad, funcionalidad, facilidad de uso y por las múltiples posibilidades que ofrece, se ha optado en este proyecto por trabajar con esta tecnología.

4.3 OBJETIVOS

Este trabajo se estructura en torno a cuatro objetivos fundamentales:

1. Analizar y estudiar con rigor y en profundidad las tecnologías Blockchain e IoT. Conocer sus bases, funcionamiento y estado actual, para así ser capaces de inferir su potencial de aplicación y uso.
2. Desplegar una red basada en IoT sobre una infraestructura Hyperledger. Ser capaz de implementar la red con sus correspondientes participantes y *peers* y conocer su infraestructura y arquitectura de red.
3. Entender cómo funciona y qué es chaincode en Hyperledger, su programación y los requisitos necesarios para su implementación.
4. Desarrollar una demostración real del funcionamiento de ésta, implementando un caso de uso donde se aplican todos los puntos anteriores. Con ello, se finaliza por completo el estudio de las tecnologías y se recopila toda la información generada a lo largo del proyecto.

4.4 METODOLOGÍA

Con el fin de cumplir los objetivos mencionados anteriormente, se ha seguido una metodología de trabajo *agile*, de tal forma que se ha dividido el trabajo en pequeñas partes que se han ido entregando cada dos/tres semanas.

DEFINICIÓN DEL TRABAJO

Las ventajas de usar esta metodología de trabajo es que mejora la calidad, existe un mayor compromiso y ayuda a aumentar la rapidez y la productividad. Esta metodología se enfoca para grupos de trabajo grandes, sin embargo, también es válido para bordar un proyecto por un sólo individuo dado que se regulan las expectativas sobre el tiempo invertido para cumplir con los objetivos establecidos.

Para ello, se ha usado Trello, una aplicación de gestión de proyectos utilizada tanto a nivel profesional como personal. Se trata de una herramienta que permite organizar prácticamente todo, trabajos en grupo de forma colaborativa mediante los denominados tableros virtuales compuestos de listas de tareas en forma de columnas. Mejora las rutinas de trabajo de un equipo gracias a su funcionalidad: genera prioridades, tiempos, avisos, etc.

Las ventajas de usar Trello, además de las ya comentadas, son:

- La aplicación es online siendo editable y compartida en tiempo real por todos
- La aplicación es muy simple e intuitiva
- Dota de un sistema de notificaciones y función de etiquetado de colores
- El motor de búsquedas es bueno y rápido
- Se pueden publicar tanto tableros públicos como privados
- Contiene conexión segura y confidencialidad de los datos con copias de seguridad cifradas similares a las que utilizan los bancos
- Además, la versión es gratuita y muy útil.

En la Figura 12 se muestra la organización que se ha seguido, basada en el método Kanban donde se utilizan los famosos To Do, Doing y Done.

DEFINICIÓN DEL TRABAJO

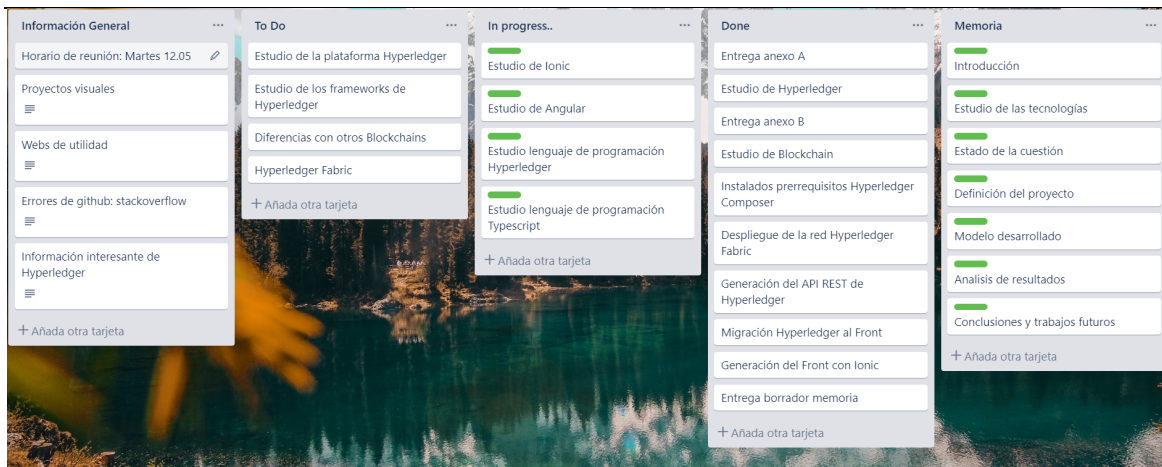


Figura 21: Esquema de uso de Trello

Además, se ha realizado un diagrama de Gantt con la planificación de las tareas de este proyecto:

Nombre	Duración	Inicio	Terminado
DOCUMENTACIÓN	213 days	13/11/18	14/6/19
Entrega Anexo A	1 day	13/11/18	13/11/18
Entrega Anexo B	1 day	18/2/19	18/2/19
Entrega borrador memoria	1 day	9/6/19	11/6/19
Entrega memoria	1 day	14/6/19	14/6/19
ESTUDIO Y ANALISIS	155 days	5/1/19	9/6/19
Estudio de Blockchain	54 days	5/1/19	28/2/19
Estudio de Hyperledger	101 days	28/2/19	9/6/19
Estudio de herramientas Hyperledger	91 days	10/3/19	9/6/19
LEVANTAR INFRAESTRUCTURA	22 days	16/5/19	7/6/19
Despliegue de la red Hyperledger	14 days	16/5/19	30/5/19
Despliegue de API REST	8 days	30/5/19	7/6/19
DESARROLLO CASO DE USO	14 days	16/5/19	30/5/19
Requisitos	14 days	16/5/19	30/5/19
Diseño	14 days	16/5/19	30/5/19
Entregas	14 days	16/5/19	30/5/19
DESPLIEGUE DE RED	8 days	30/5/19	7/6/19
Diseño	8 days	30/5/19	7/6/19
Pruebas	8 days	30/5/19	7/6/19
REVISIÓN Y VALIDACIÓN	37 days	5/1/19	7/6/19
Tests parciales	14 days	16/5/19	30/5/19
Validación final	22 days	16/5/19	7/6/19
DEFENSA DEL PROYECTO	1 day	18/6/19	16/5/19

Figura 22: Diagrama de Gantt del proyecto

4.5 PLANIFICACIÓN Y ESTIMACIÓN ECONÓMICA

Atendiendo a la planificación del apartado anterior, se estiman los siguientes costes recogidos en siguiente Tabla:

	Justificación	
Infraestructura	IBM Cloud. Linux Foundation. Plan gratuito 6 meses	0 €
Equipo	Ordenador portátil 16 GB RAM	0 €
Herramienta	Open Source	0 €
Implementación	Coste de un desarrollador 30 €/h + un consultor analista de Blockchain 30 €/ h (dos meses a media jornada)	9.600 €
Total		9.600 €

Tabla 3: Estimación de costes

Coste de infraestructura: no es necesario realizar ninguna instalación, ya que Linux Foundation ofrece a estudiantes de Universidad un plan gratuito semestral, por lo que no ha hecho falta adquirir ningún clúster ni almacenamiento extra.

Equipo: un ordenador con suficiente capacidad de procesamiento

Herramienta: no se necesita ninguna licencia, dado que es *opensource*.

Desarrollador: *full-stack* con conocimientos en Hyperledger Fabric, JavaScript, JSON, Go y capaz de maquetar aplicaciones web, y un analista consultor de Blockchain y cadena de suministro.

Capítulo 5. MODELO DESARROLLADO

En el presente capítulo se expone el caso de uso desarrollado en este proyecto. Tal y como se mencionó en anteriores capítulos, Blockchain permite muchas posibilidades de aplicación en el mundo empresarial. Una de ellas, la elegida para este proyecto, es la aplicación basada en la cadena de suministro.

Si se echa la vista atrás, anteriormente la mayoría de los productos se fabricaban y se consumían localmente, y sólo una parte se importaba de otros países. Hoy en día, sin embargo, como consecuencia de un acelerado proceso de globalización, los productos se producen y consumen en cualquier país, incrementándose enormemente la complejidad.

Este caso de uso está conformado por tres actores principales: un sensor IoT, un comprador y un proveedor. Se busca un funcionamiento integrado y ágil para ganar eficiencia y responder a las demandas del comercio. Hyperledger ofrece la posibilidad de gestionar mejor la trazabilidad del producto al agilizar y aligerar la gestión de los procesos de logística; así como automatizar la ejecución de las distintas relaciones contractuales.

Se busca obtener registros del origen de un producto, así como su recorrido hasta llegar a la tienda final, de tal manera que toda transacción se graba en un bloque inalterable que se copia en múltiples nodos. Los datos de cada uno de ellos no se pueden modificar, son inmutables, por lo que si hubiera una alteración, el sistema lo impediría por defecto.

El fin último de este caso de uso es permitir que los usuarios sepan con seguridad que el producto está llegando a sus manos y que no se ha producido ninguna alteración.

5.1 ANÁLISIS

En el presente apartado se muestra un análisis en alto nivel de todas las clases de este proyecto. Previamente se comentó que Hyperledger Fabric y Composer constan de tres archivos principales: .cto, .js y .acl.

El archivo .cto es el que define el modelo de negocio con sus participantes, activos y transacciones.

MODELO DESARROLLADO

Por un lado, están los **participantes** de la red de Hyperledger Fabric: Comprador, Vendedor y Transportista. Cada uno de ellos tiene una serie de atributos asociados necesarios para el correcto funcionamiento de las transacciones que se pueden generar.

Por otro lado, se encuentran los **activos**: Producto, Envío y Contrato. Cada uno de estos activos se encuentran asociados a uno o varios participantes. El producto es vendido por un Vendedor, el envío está relacionado con el que compra el producto que vende el Vendedor y que es entregado por el Transportista, y el Contrato es establecido entre un vendedor y un comprador, donde participa a su vez el transportista. Cada uno de estos activos a su vez tiene una serie de características o atributos que los definen.

5.1.1 DIAGRAMA DE CLASES

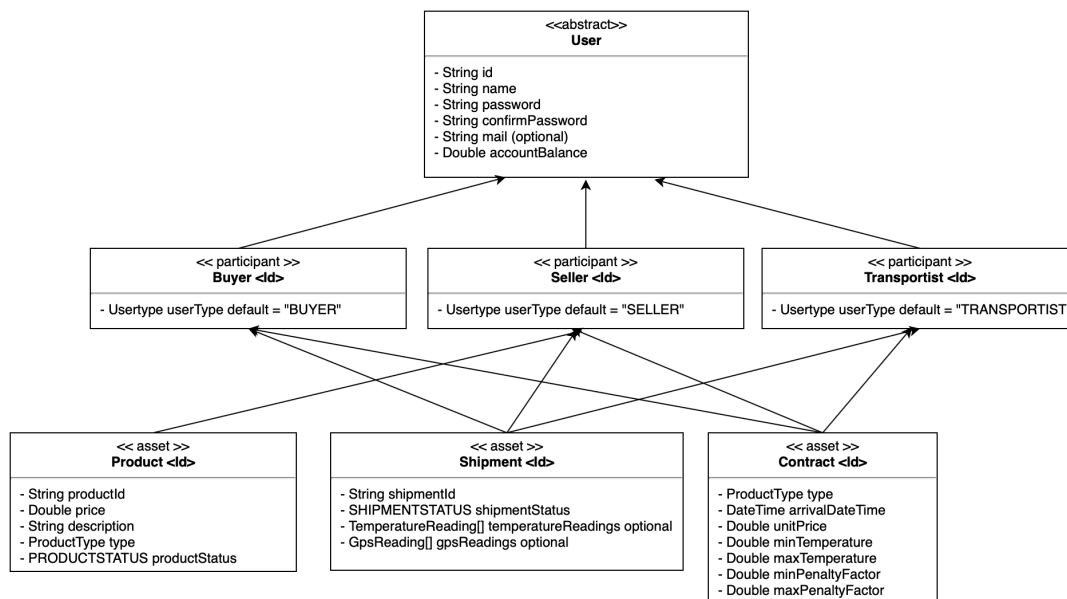


Figura 23: Diagrama de clases

Estas clases se encuentran en el archivo de modelo de red **.cto**. A continuación se muestra el código:

```

abstract participant User identified by id {
  o String id
  o String name
  o String password
  o String confirmPassword

```

```

  o String mail optional
  o Double accountBalance
}

```

```

participant Seller extends User {
  o UserType userType default = "SELLER"

```


MODELO DESARROLLADO

<pre> } participant Buyer extends User { o Ustertype userType default = "BUYER" } participant Transportist extends User { o Ustertype userType default = "TRANSPORTIST" } asset Product identified by productId { --> Seller seller o String productId o Double price o String description o ProductType type o PRODUCTSTATUS productStatus } asset Shipment identified by shipmentId { --> Seller seller --> Buyer buyer --> Transportist transportist </pre>	<pre> --> Product product --> Contract contract o String shipmentId o SHIPMENTSTATUS shipmentStatus o TemperatureReading[] temperatureReadings optional o GpsReading[] gpsReadings optional } asset Contract identified by contractId { --> Seller seller --> Buyer buyer --> Transportist transportist o String contractId o ProductType type o DateTime arrivalDateTime o Double unitPrice o Double minTemperature o Double maxTemperature o Double minPenaltyFactor o Double maxPenaltyFactor } </pre>
--	--

Por último, se va a explicar el modelo de **transacciones** y su relación con los activos y participantes. Las transacciones que se pueden llevar a cabo en el caso de uso implementado son:

- Registrarse (Figura 19)
- Iniciar sesión (Figura 19)
- Comprar producto (Figura 20)
- Confirmar pedido (Figura 21)
- Cambiar estado del pedido a en transito (Figura 22)
- Cambiar estado del pedido a en entregado (Figura 22)
- Registrar temperatura (Figura 23)
- Registrar coordenadas de GPS (Figura 23)

MODELO DESARROLLADO

La relación entre los activos y participantes se muestra en el siguiente diagrama:

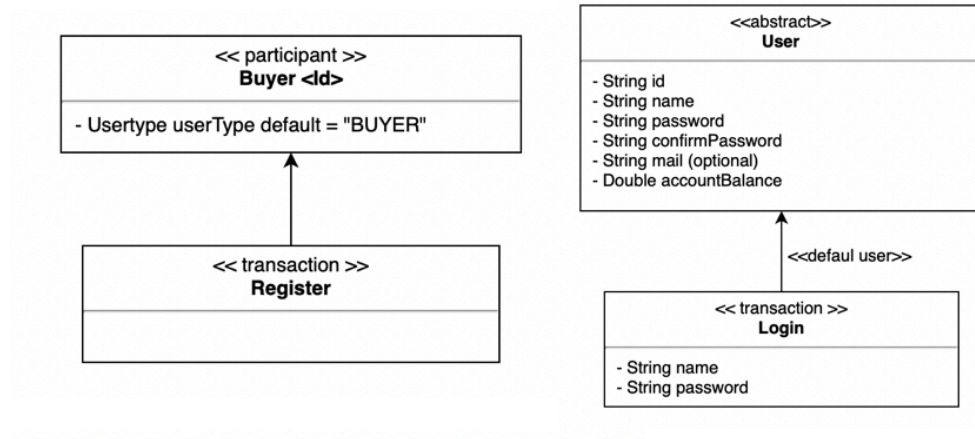


Figura 24: Transacción de Registrarse y de Iniciar Sesión

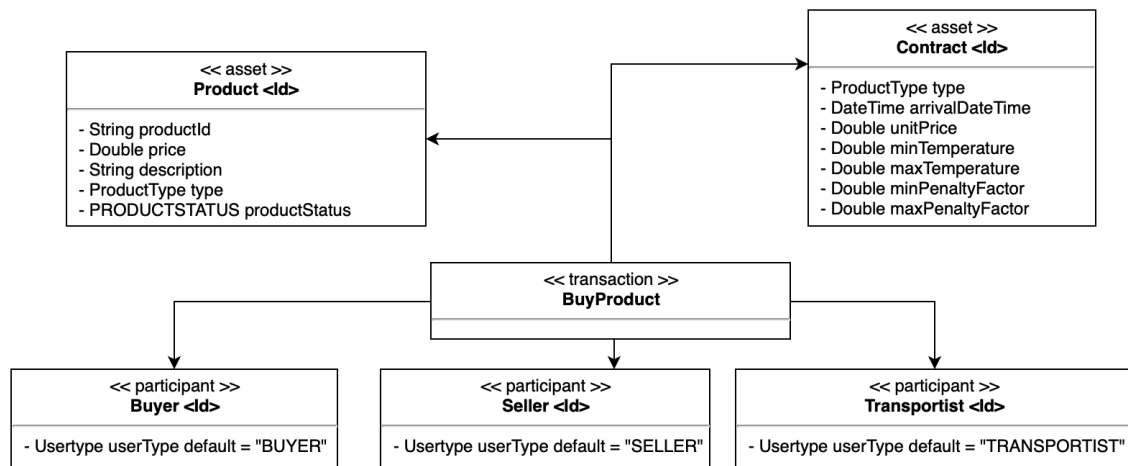


Figura 25: Transacción compra de producto

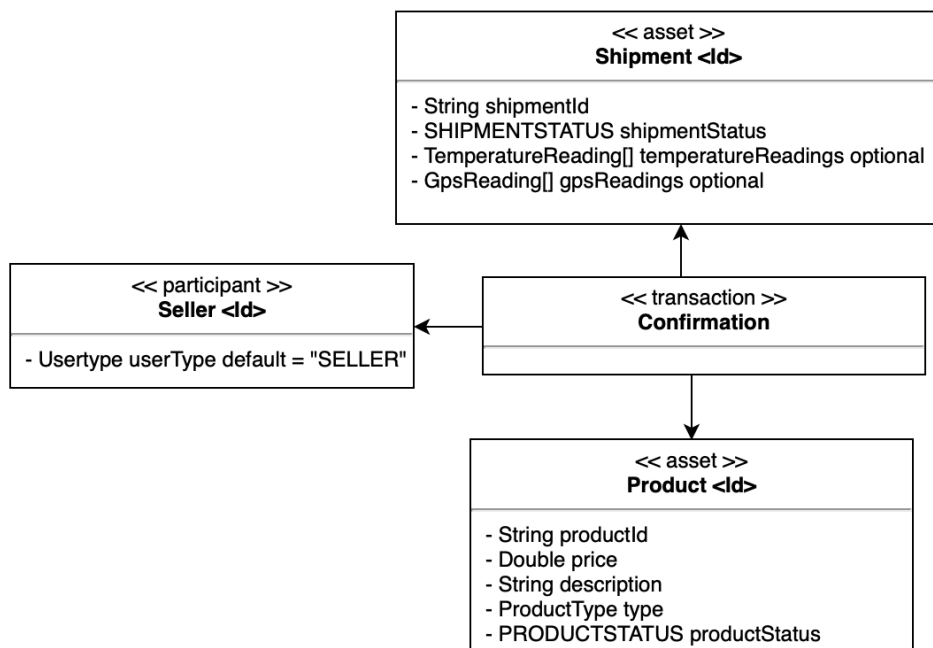


Figura 26: Transacción de confirmación de envío

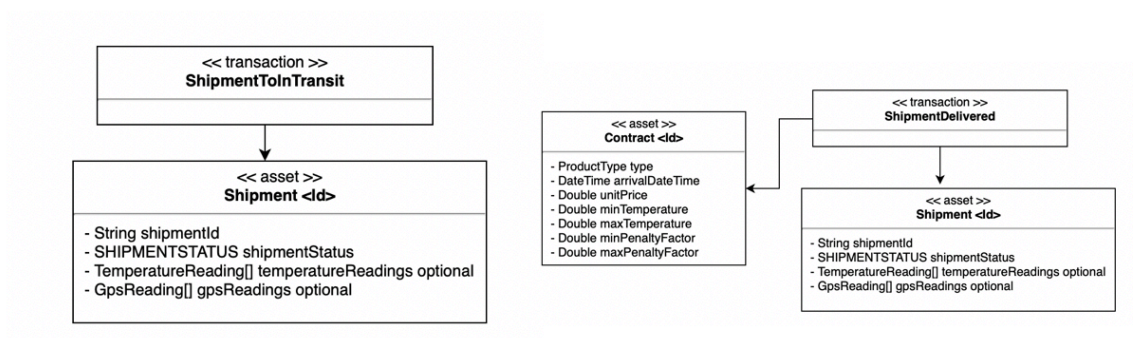


Figura 27: Transacciones de Registro de pedido en Tránsito o Entrega

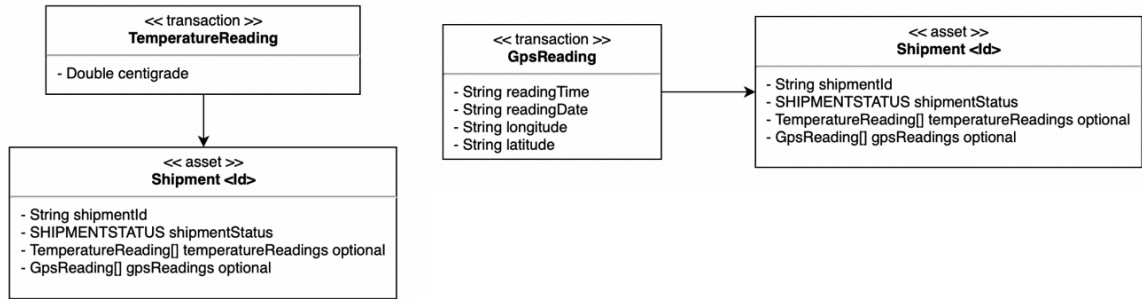


Figura 28: Transacciones de Lectura de Temperatura y de GPS

A continuación se muestra el código del modelo .cto donde se definen estas transacciones y sus relaciones.

```

transaction Register {
  o Buyer buyer
}
transaction Login {
  --> User defaultUser
  o String name
  o String password
}
transaction BuyProduct {
  --> Seller seller
  --> Buyer buyer
  --> Transportist transportist
  --> Product product
  --> Contract contract
}
transaction Confirmation {
  --> Shipment shipment
  --> Seller seller
  --> Product product
}
  
```

```

transaction ShipmentToInTransit{
  --> Shipment shipment
}
transaction ShipmentDelivered{
  --> Shipment shipment
  --> Contract contract
}
transaction TemperatureReading{
  --> Shipment shipment
  o Double centigrade
}
transaction GpsReading {
  --> Shipment shipment
  o String readingTime
  o String readingDate
  o String latitude
  o String longitude
}
  
```

5.1.2 DIAGRAMA DE CASOS DE USO

El participante tipo **Buyer** debe ser capaz de realizar las siguientes actividades:

- Iniciar sesión y registrarse
- Visualizar los productos disponibles
- Comprar productos
- Ver el estado de su pedido
- Ver el tracking del pedido

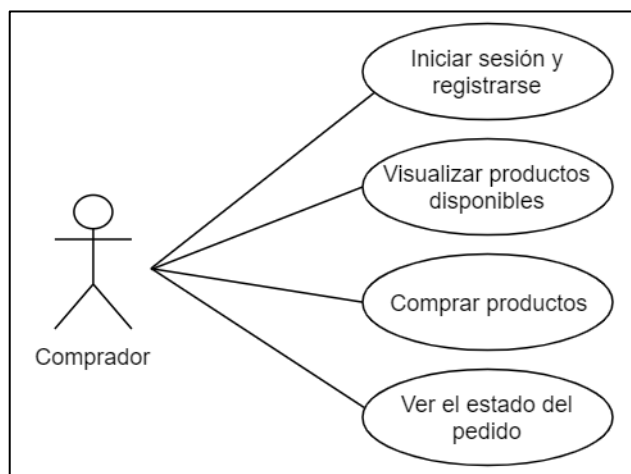


Figura 29: Diagrama de caso de uso del Comprador

El participante tipo **Vendor** debe ser capaz de realizar las siguientes actividades:

- Iniciar sesión y registrarse
- Visualizar todos los productos vendidos al comprador
- Confirmar el pedido, validándolo de este modo y dando opción al transportista de llevar a cabo el reparto
- Ver el estado del pedido

MODELO DESARROLLADO

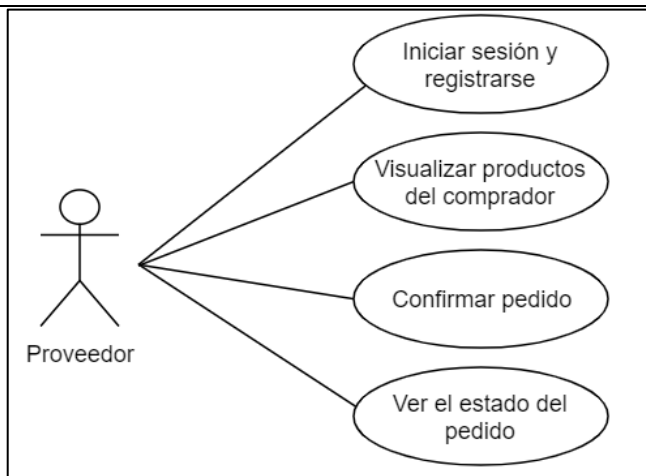


Figura 30: Diagrama de caso de uso del Proveedor

El participante tipo **Transportist** (*sensor IoT*) debe ser capaz de realizar las siguientes actividades:

- Iniciar sesión y registrarse
- Visualizar todos los productos previamente confirmados por *Vendor*
- Cambiar el estado del envío a “IN TRANSIT” o a “DELIVERED” y añadir registro de temperatura y GPS como un sensor IoT

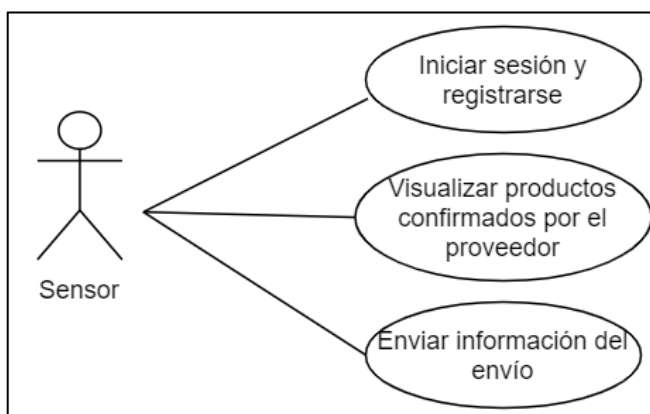


Figura 31: Diagrama de caso de uso del Sensor IoT

5.2 DISEÑO

Con el objetivo último de disponer del registro total de una cadena de suministro basada en el campo de sensores IoT e integrar esta información en una Blockchain, ha sido necesario programar por un lado el *Back End* y por otro el *Front End*.

Para el desarrollo del *Back End*, se ha usado la tecnología Hyperledger, desplegando la red de nodos, usando el consenso e implementando chaincode mediante el lenguaje de programación de Hyperledger Fabric y JavaScript. Para probar e ir testeando la red antes de lanzarla, se ha usado Hyperledger Composer. Por último, para lanzar la red, se necesita un servidor que deje corriendo el API REST que se genera y con el que interactúa el *Front End*. Para el desarrollo del *Front End*, se ha usado un *Framework* de JavaScript de código abierto, denominado Ionic, para el cual ha sido necesario el aprendizaje del lenguaje de programación TypeScript, CSS y HTML.

Para generar la aplicación web de manera optimizada se ha usado Node.js, entorno de JavaScript de código abierto del lado del servidor, basado en eventos. Ayuda a establecer la comunicación entre el *Back* y el *Front* y proporciona una forma de ejecutar el chaincode del usuario, de realizar transacciones en la red, de supervisar eventos, etc.

La siguiente imagen muestra un esquema general de las tecnologías usadas:

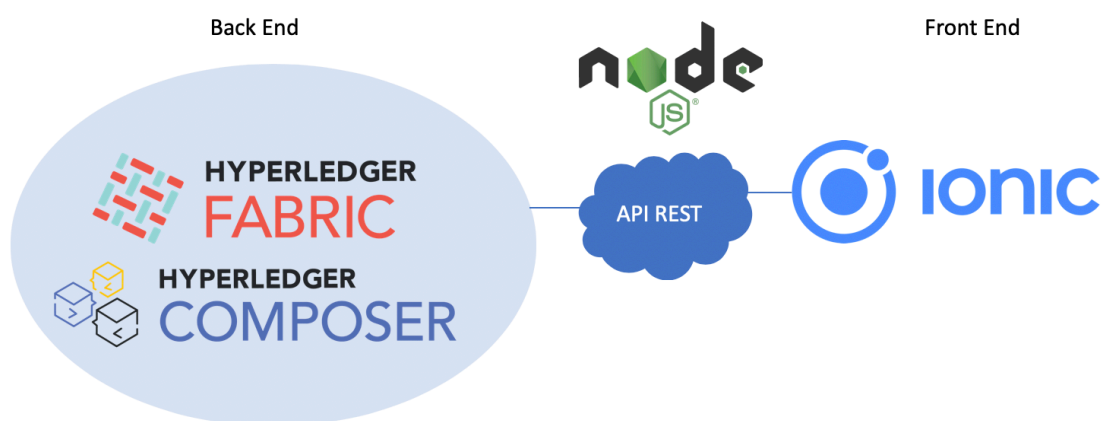


Figura 32: Esquema de las tecnologías usadas en este proyecto

5.3 IMPLEMENTACIÓN

5.3.1 DESARROLLO BACK END

En la siguiente imagen se muestra el proceso que se ha llevado a cabo para crear la aplicación Hyperledger [29]:

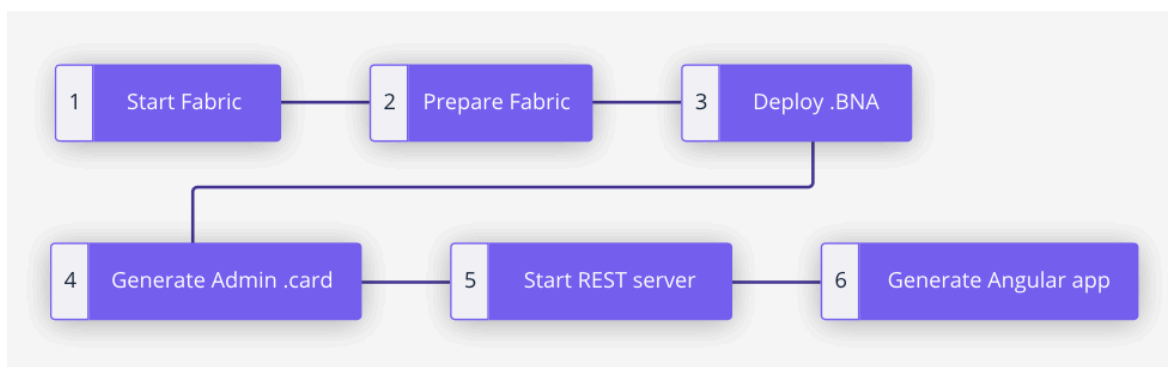


Figura 33: Proceso de creación de la aplicación

En primer lugar, es necesario tener los siguientes requisitos instalados:

- Sistemas operativos: Ubuntu / Mac OS
- Docker [30]
- Node.js [31]
- npm
- Git
- Python
- Un editor de código (VSCode es el recomendado)
- Herramientas útiles de CLI para los desarrolladores de Hyperledger Composer.
 - composer-cli que contiene todas las operaciones esenciales
 - generator-hyperledger-composer
 - composer-rest-server
 - Yeoman

Teniendo en cuenta el esquema previo e instalados todos los prerrequisitos, a continuación, se muestra paso por paso la implementación de este en el entorno local del ordenador [32].

1. Instalar Hyperledger Fabric

Antes de iniciar, es importante asegurarse de que Docker se está ejecutando.

```
mkdir ~/fabric-dev-servers && cd ~/fabric-dev-servers

curl -O https://raw.githubusercontent.com/hyperledger/composer-tools/master/packages/fabric-dev-servers/fabric-dev-servers.tar.gz

tar -xvf fabric-dev-servers.tar.gz

cd ~/fabric-dev-servers

export FABRIC_VERSION=hlfv11

./downloadFabric.sh
```

El comando `./downloadFabric.sh` extrae todas las imágenes de Docker.

2. Iniciar Hyperledger Fabric

```
cd ~/fabric-dev-servers

export FABRIC_VERSION=hlfv11

./startFabric.sh

./createPeerAdminCard.sh
```

El comando `./startFabric.sh` inicia Hyperledger Fabric en local.

El comando `./createPeerAdminCard.sh` envía una tarjeta ID para el administrador, que es PeerAdmin. Ésta, se necesita para modificar el código que se ejecuta en los *peers* de la implementación de red.

Cuando este script finaliza, Hyperledger Fabric se está ejecutando en el ordenador. Para apagarlo simplemente:

```
./stopFabric.sh
```

Una vez Docker está funcionando, y Fabric está inicializado, el siguiente paso es escribir el código de Hyperledger Fabric.

MODELO DESARROLLADO

Como se ha mencionado en apartados anteriores, Hyperledger consta de tres archivos fundamentales: **.cto**, **.js**, y **.acl**.

En el archivo **.cto** se han definido:

- Participantes: Comprador, Proveedor y Transportista. Éste último actúa como un sensor IoT, ya que simula el registro de datos que un sensor enviaría.
- Activos: Producto, Envío y Contrato.
- Transacciones: Registrarse, Login, Confirmar un producto, Envío en tránsito, Envío Entregado, Lectura de GPS y Lectura de temperatura.

Una vez desarrollado el modelo, se tiene que programar lo que debe realizar cada transacción, en los archivos JavaScript y dotar de los permisos a los participantes en el archivo **.acl**.

Tras la programación de todos los archivos, para levantar la red, hay que asegurarse de que Docker está corriendo, Fabric está instalado y la tarjeta (PeerAdmin) está creada correctamente.

En primer lugar, se genera el archivo **.bna**, que es el archivo de nuestra red, que autentica la tarjeta PeerAdmin.

```
composer archive create -t dir -n .
```

Cuando el despliegue finaliza, se emite una tarjeta ID para el administrador de la red, cuyas credenciales se almacenan en el archivo **networkadmin.card**.

El siguiente paso es instalar la red mediante el siguiente comando:

```
composer network install --archiveFile exchange@0.0.1.bna --card PeerAdmin@hlfv1
```

Este comando de instalación de red se utiliza para instalar el archivo de red empresarial en los pares de Hyperledger Fabric de la red de Blockchain a la que se está conectando. Este comando debe ejecutarse antes del comando de inicio de red del compositor.

A continuación se inicia la red mediante:

MODELO DESARROLLADO

```
composer network start --networkName exchange --networkVersion 0.0.1 --networkAdmin admin --  
networkAdminEnrollSecret adminpw --card PeerAdmin@hlfv1 --file networkadmin.card
```

La utilidad de inicio de red del compositor se utiliza para iniciar una versión específica de una red de negocios que se ha instalado previamente en una red Hyperledger Fabric v1.1. Por último es necesario importar una tarjeta que se crea automáticamente.

```
composer card import --file networkadmin.card
```

Para generar el API REST lo único que hay que hacer es asegurarse de que nuestra red está funcionando y escribir el siguiente comando en una ventana de terminal:

```
composer-rest-server -c admin@exchange -n always
```

5.3.2 DESARROLLO FRONT END

Este apartado es el que más tiempo ha requerido dada la naturaleza de la tecnología y sus distintos versionados. El desarrollo, tal y como se mencionó en capítulos anteriores, se ha desarrollado con el Framework de Ionic. Con este framework se utilizan 3 archivos: typescript (.ts), HTML y SCSS.

Para desarrollarlo, en primer lugar, es necesario instalar su generador. Una de las maneras de hacerlo, y como se ha hecho en este proyecto es la siguiente:

```
npm install -g generator-ionic
```

Una vez realizado este paso, lo único que hay que hacer es crearse una carpeta y generar la aplicación mediante el siguiente comando:

```
ionic start myapp
```

El siguiente paso requiere su tiempo, ya que se trata de programar la aplicación final. Para ir viendo cómo queda finalmente y testeando los errores hay que escribir en el terminal:

```
ionic serve --port 8989
```

Capítulo 6. ANÁLISIS DE RESULTADOS

En el presente capítulo se van a mostrar los resultados obtenidos a lo largo de este proyecto.

6.1 HYPERLEDGER COMPOSER

Tal y como se ha mencionado en anteriores apartados, Hyperledger Composer es uno de los proyectos Hyperledger alojados por The Linux Foundation, que facilita la construcción de estas aplicaciones.

Hyperledger Composer ofrece una interfaz basada en navegador, denominada Playground, ya que permite ir modelando y testeando la red que se quiere montar. Para comenzar, desde una ventana de un terminal, hay que ejecutar el siguiente comando:

```
composer-playground
```

Tras ello, se abrirá una ventana en el navegador corriendo en el puerto 8080:

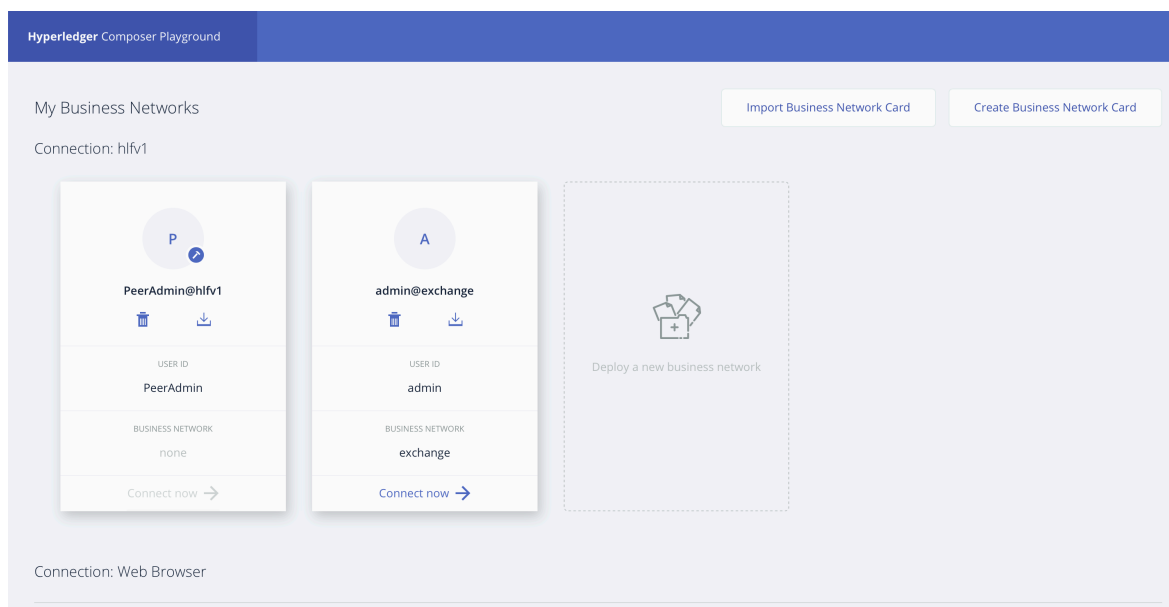


Figura 34: Hyperledger Composer Playground

Composer permite acceder a las redes que se hayan desarrollado en el servidor. En este caso, PeerAdmin es obligatorio tenerlo instalado y corriendo. Una vez instalas la red en tu

ANÁLISIS DE RESULTADOS

ordenador, aparece la tarjeta llamada admin en *wallet*. Este *wallet* puede contener múltiples redes de negocios implementadas.

Cuando se conecta a una cadena de bloques externa, las tarjetas de red representan todo lo necesario para conectarse a la red, pues incluyen detalles de conexión, material de autenticación y metadatos.

Para conectarse a la red, ver el modelo desarrollado y testear con el es necesario clicar en “Connect now”. A continuación aparece la siguiente pantalla:

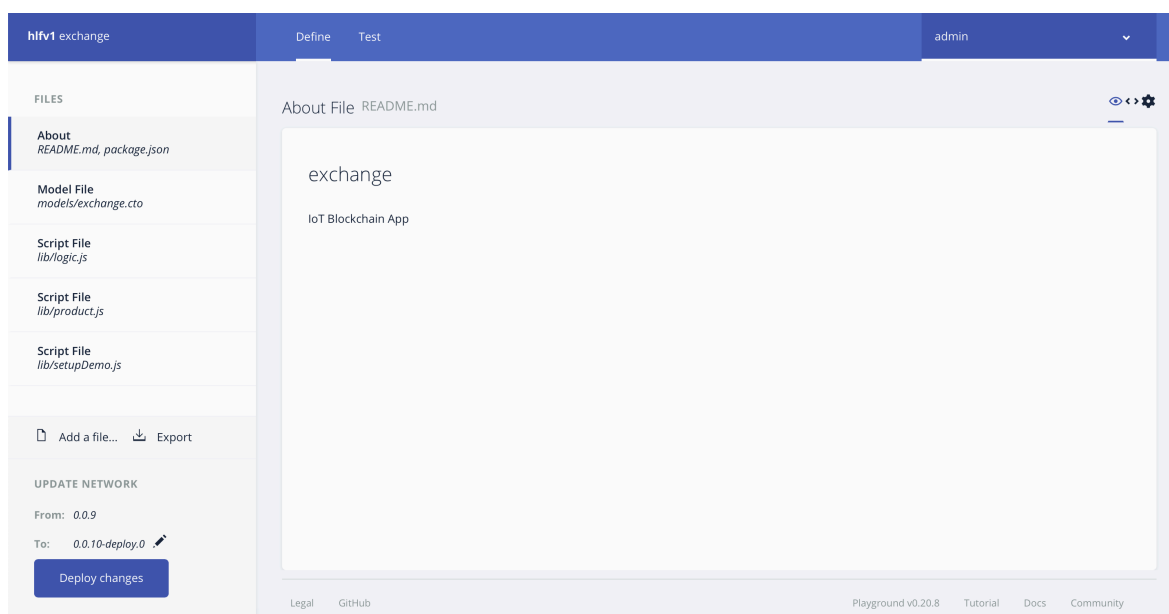


Figura 35: Hyperledger Composer Playground (Define)

En la pestaña "Define" se encuentran los archivos donde se programa la red. En el archivo `exchange.cto` se encuentra definido el modelo de la red, mientras que los archivos javascript representan toda la lógica de transacciones y contratos desplegados. Para probar y ver si el modelo funciona correctamente, simplemente hay que acceder a la pestaña "Test", donde se encontraría:

hlfv1 exchange Define Test admin

PARTICIPANTS

Buyer

Seller

Transportist

ASSETS

Contract

Product

Shipment

TRANSACTIONS

All Transactions

Submit Transaction

Participant registry for org.acme.Buyer

+ Create New Participant

ID	Data
1	<pre>{ "\$class": "org.acme.Buyer", "userType": "BUYER", "id": "1", "name": "ana", "password": "ana", "email": "ana@acme.com" }</pre> Show All
1562753453	<pre>{ "\$class": "org.acme.Buyer", "userType": "BUYER", "id": "1562753453", "name": "rai", "password": "rai", "email": "rai@acme.com" }</pre> Show All

Legal GitHub Playground v0.20.8 Tutorial Docs Community

Figura 36: Hyperledger Composer Playground (Test)

Como se puede observar en la figura, a la izquierda obtenemos los participantes de la red: *Buyer*, *Seller* y *Transportist* ; los activos: *Contract*, *Product* y *Shipment* ; y el *ledger*, donde se encuentran registradas todas las transacciones que se invoquen.

El botón “Submit Transaction” permite probar las transacciones previamente programadas. Para enviar una transacción, es necesario saber JSON, dado que es la naturaleza con la que está diseñado Hyperledger.

El *ledger* se encuentra en “All Transactions”. Como se observa en la Figura 32, ahí se quedan registradas todas las transacciones que se realizan a lo largo del proyecto, de manera inmutable y completamente segura.

ANÁLISIS DE RESULTADOS

Submit Transaction

Transaction Type: BuyProduct

JSON Data Preview

```

1 {
2   "$class": "org.acme.BuyProduct",
3   "seller": "resource:org.acme.Seller#9900",
4   "buyer": "resource:org.acme.Buyer#0089",
5   "transportist": "resource:org.acme.Transportist#8905",
6   "product": "resource:org.acme.Product#2503",
7   "contract": "resource:org.acme.Contract#1785"
8 }

```

☐ Optional Properties

Just need quick test data? [Generate Random Data](#)

Cancel Submit

Figura 37: Hyperledger Composer Playground (Submit Transaction)

hlft1 exchange	Define	Test	admin
PARTICIPANTS			
Buyer			
Seller			
Transportist			
ASSETS			
Contract			
Product			
Shipment			
TRANSACTIONS			
All Transactions			
Submit Transaction			

Date, Time	Entry Type	Participant	
2019-07-11, 18:31:54	Login	admin (NetworkAdmin)	view record
2019-07-11, 18:31:54	Login	admin (NetworkAdmin)	view record
2019-07-11, 18:31:45	UpdateAsset	admin (NetworkAdmin)	view record
2019-07-11, 18:24:19	Login	admin (NetworkAdmin)	view record
2019-07-11, 18:24:19	Login	admin (NetworkAdmin)	view record
2019-07-11, 18:23:51	Login	admin (NetworkAdmin)	view record

Legal GitHub Playground v0.20.8 Tutorial Docs Community

Figura 38: Hyperledger Composer Playground (Ledger)

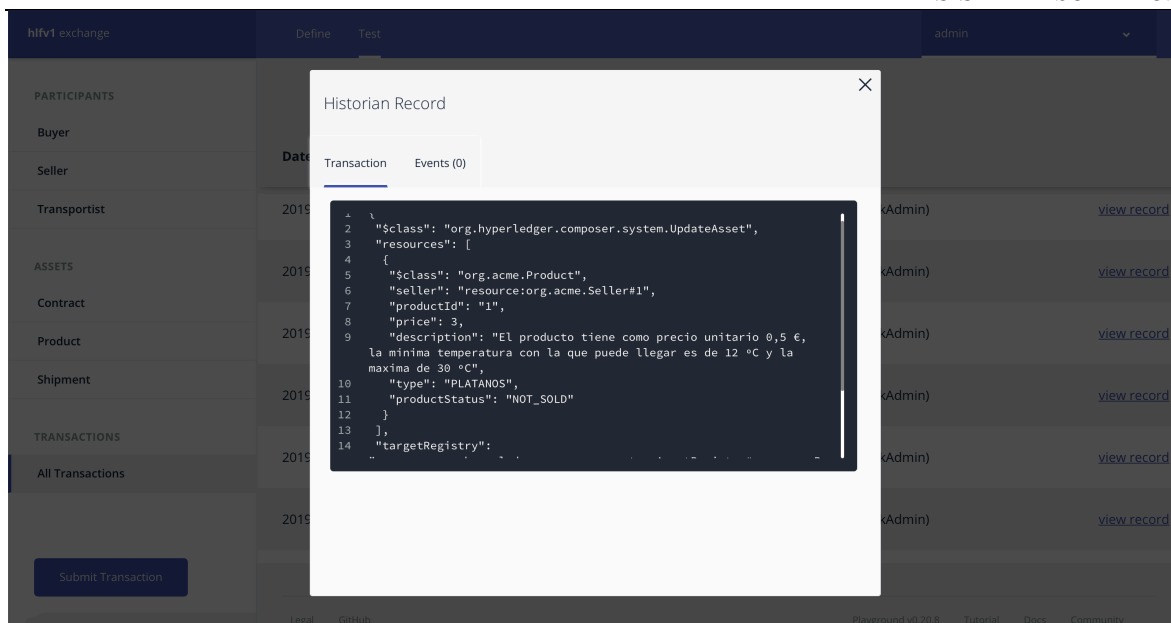


Figura 39: Hyperledger Composer Playground (Ledge)

6.2 API REST

Hyperledger Composer incluye en un proceso NodeJs independiente, exponer una red expresarial como un API REST. Para ello, es necesario abrir un terminal y escribir el siguiente comando:

```
composer-rest-server -c admin@exchange -n always
```

A continuación hay que abrir una ventana del navegador, donde se encontrará corriendo la API en el puerto 3000 (Figura 34). Esta es otra manera de ver y testear los participantes, activos y transacciones que tiene la red.

En la actualidad no existe proyecto o aplicación que no disponga de una API REST para la creación de servicios. De nuevo, para obtener los datos (get) y generarlos (post), es necesario el conocimiento de JSON. Esto facilita la conexión con el interfaz gráfico ya que sólo haría falta acceder al punto final del API REST y definir los recursos con una URI.

ANÁLISIS DE RESULTADOS

Hyperledger Composer REST server			
org_acme_Buyer : A participant named Buyer	Show/Hide	List Operations	Expand Operations
org_acme_BuyProduct : A transaction named BuyProduct	Show/Hide	List Operations	Expand Operations
org_acme_Confirmation : A transaction named Confirmation	Show/Hide	List Operations	Expand Operations
org_acme_Contract : An asset named Contract	Show/Hide	List Operations	Expand Operations
org_acme_GpsReading : A transaction named GpsReading	Show/Hide	List Operations	Expand Operations
org_acme_HumidityReading : A transaction named HumidityReading	Show/Hide	List Operations	Expand Operations
org_acme_Login : A transaction named Login	Show/Hide	List Operations	Expand Operations
org_acme_Product : An asset named Product	Show/Hide	List Operations	Expand Operations
org_acme_Register : A transaction named Register	Show/Hide	List Operations	Expand Operations
org_acme_Seller : A participant named Seller	Show/Hide	List Operations	Expand Operations
org_acme_SetupDemo : A transaction named SetupDemo	Show/Hide	List Operations	Expand Operations
org_acme_Shipment : An asset named Shipment	Show/Hide	List Operations	Expand Operations
org_acme_ShipmentDelivered : A transaction named ShipmentDelivered	Show/Hide	List Operations	Expand Operations
org_acme_ShipmentToInTransit : A transaction named ShipmentToInTransit	Show/Hide	List Operations	Expand Operations
org_acme_TemperatureReading : A transaction named TemperatureReading	Show/Hide	List Operations	Expand Operations
org_acme_Transportist : A participant named Transportist	Show/Hide	List Operations	Expand Operations
System : General business network methods	Show/Hide	List Operations	Expand Operations

Figura 40: API REST Exchange

Hyperledger Composer REST server			
org_acme_Buyer : A participant named Buyer	Show/Hide	List Operations	Expand Operations
GET /org.acme.Buyer	Find all instances of the model matched by filter from the data source.		
POST /org.acme.Buyer	Create a new instance of the model and persist it into the data source.		
GET /org.acme.Buyer/{id}	Find a model instance by ({id}) from the data source.		
HEAD /org.acme.Buyer/{id}	Check whether a model instance exists in the data source.		
PUT /org.acme.Buyer/{id}	Replace attributes for a model instance and persist it into the data source.		
DELETE /org.acme.Buyer/{id}	Delete a model instance by ({id}) from the data source.		
org_acme_BuyProduct : A transaction named BuyProduct	Show/Hide	List Operations	Expand Operations
org_acme_Confirmation : A transaction named Confirmation	Show/Hide	List Operations	Expand Operations
org_acme_Contract : An asset named Contract	Show/Hide	List Operations	Expand Operations
org_acme_GpsReading : A transaction named GpsReading	Show/Hide	List Operations	Expand Operations
org_acme_HumidityReading : A transaction named HumidityReading	Show/Hide	List Operations	Expand Operations
org_acme_Login : A transaction named Login	Show/Hide	List Operations	Expand Operations
org_acme_Product : An asset named Product	Show/Hide	List Operations	Expand Operations
org_acme_Register : A transaction named Register	Show/Hide	List Operations	Expand Operations
org_acme_Seller : A participant named Seller	Show/Hide	List Operations	Expand Operations

Figura 41: API REST Exchange



Figura 42: API REST Exchange

6.3 FRONT END

Lo primero que se ve al arrancar la aplicación es un Login, donde es necesario que uno de los participantes inicie sesión, en caso contrario, la aplicación dará error al no haber seleccionado un tipo de usuario.

En esta parte también está la opción de registrarse como comprador, rellenando los datos que son necesarios para completar el registro (Figura 38).

Si hacemos click en Iniciar Sesión, se hace una llamada al API REST de la transacción Login, y de manera análoga haciendo click en Registrarse.

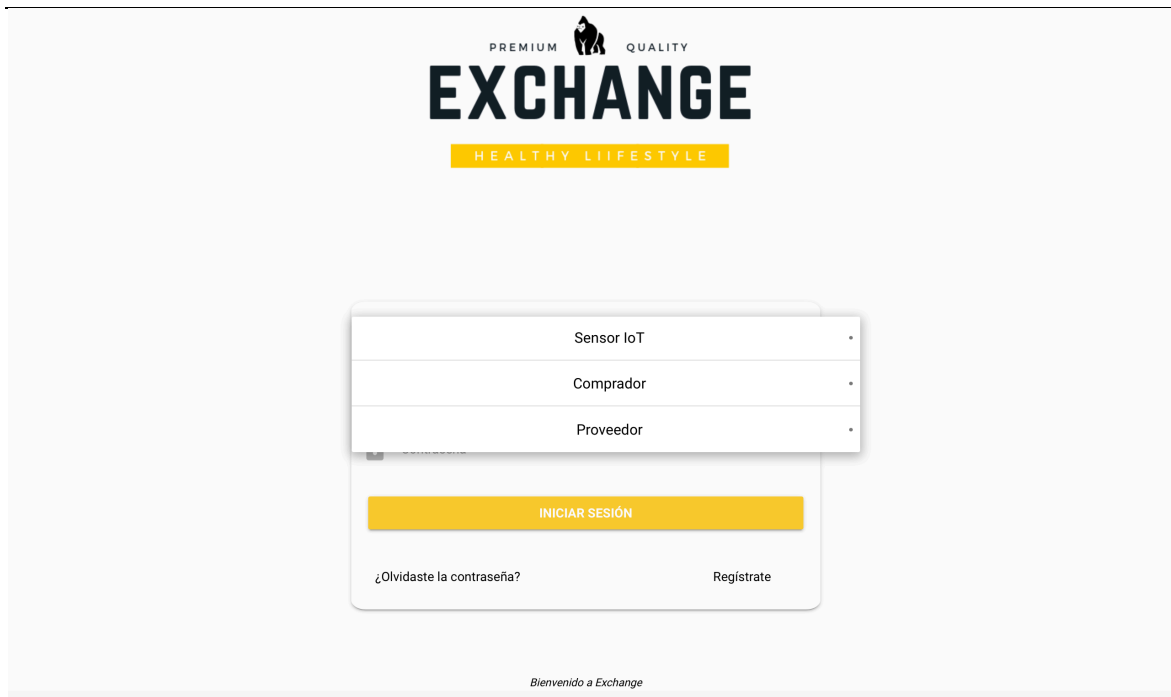


Figura 43: Página principal: Login

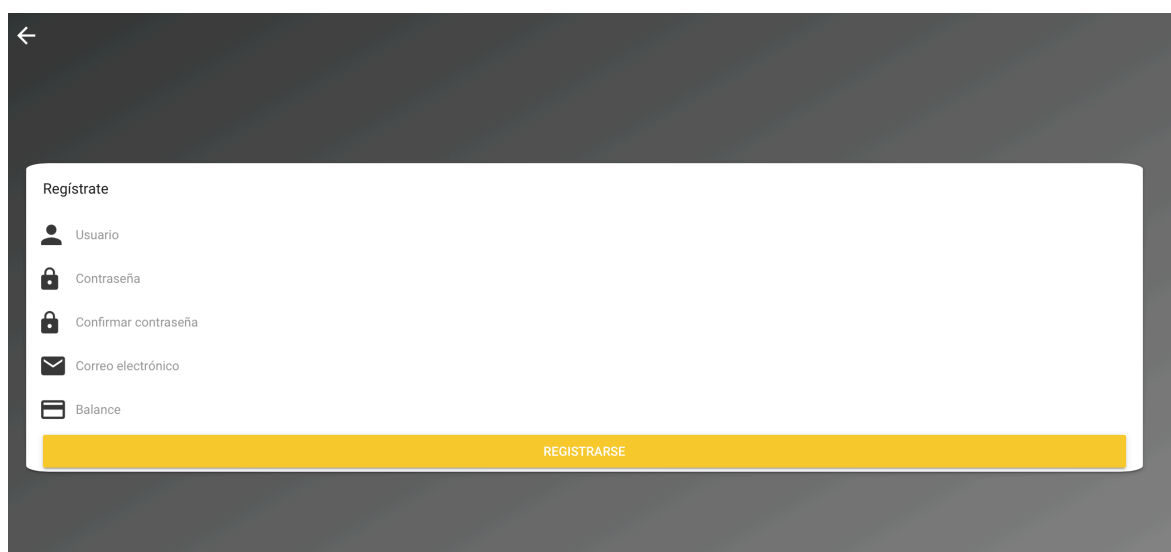


Figura 44: Página de Registro

Al iniciar sesión como Comprador, observamos la lista de productos que se pueden comprar en Exchange. Al seleccionar un producto, tienes la opción de ver sus características: precio, descripción, si se encuentra en STOCK, etc.

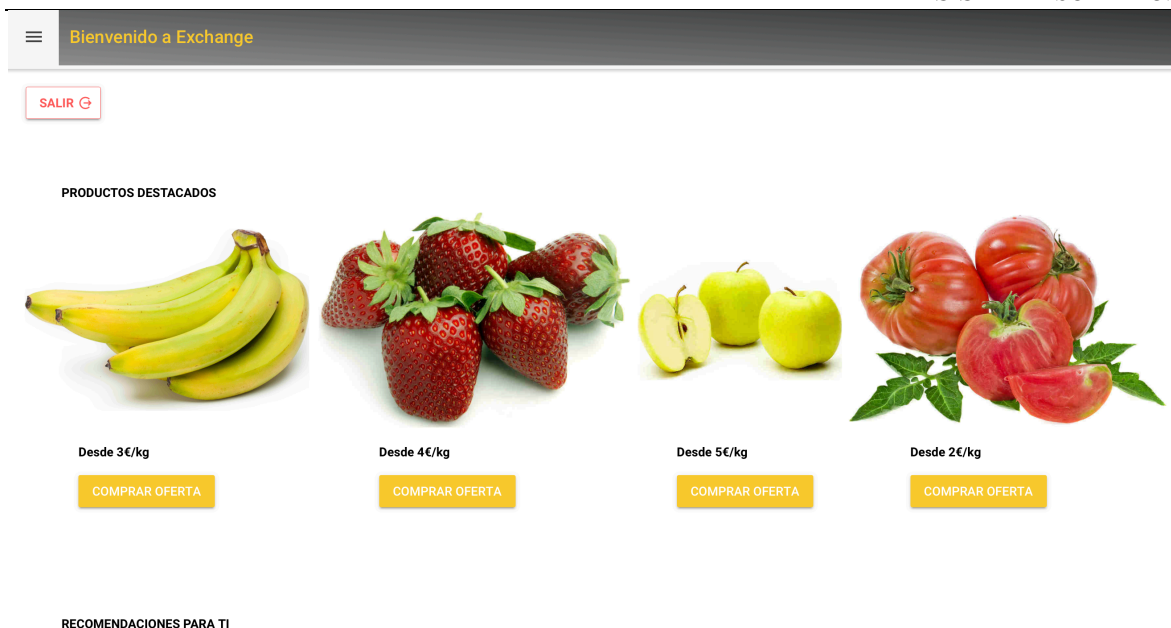


Figura 45: Página vista por el Comprador

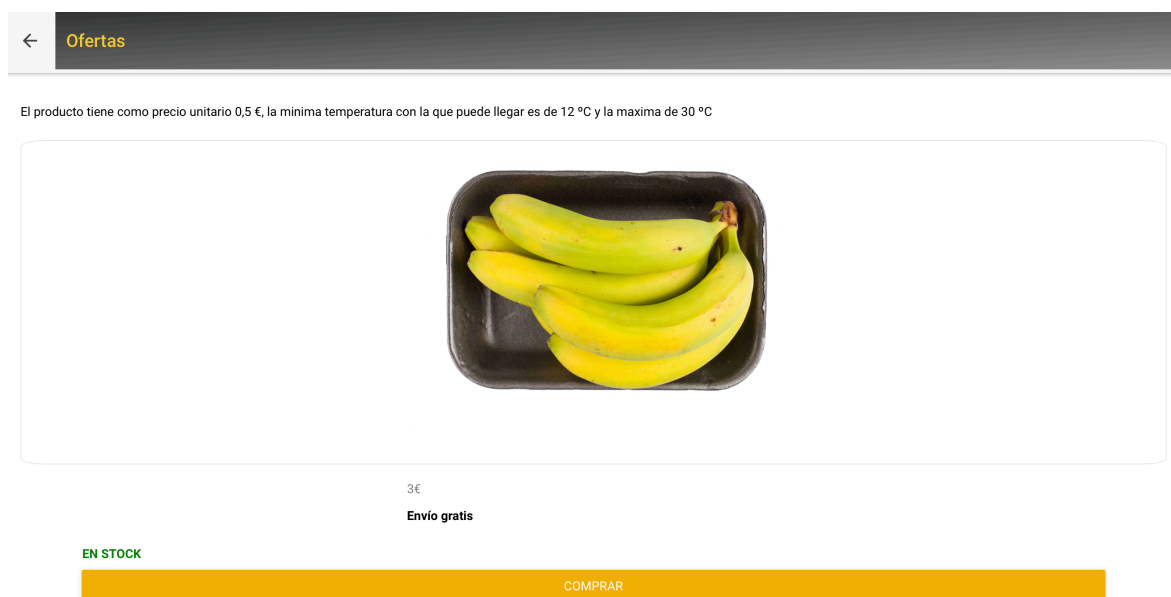


Figura 46: Página vista por el Comprador

Cuando el comprador compra un producto, éste pasa a tener el estado “PENDIENTE”, ya que está pendiente de ser confirmado por el Proveedor.

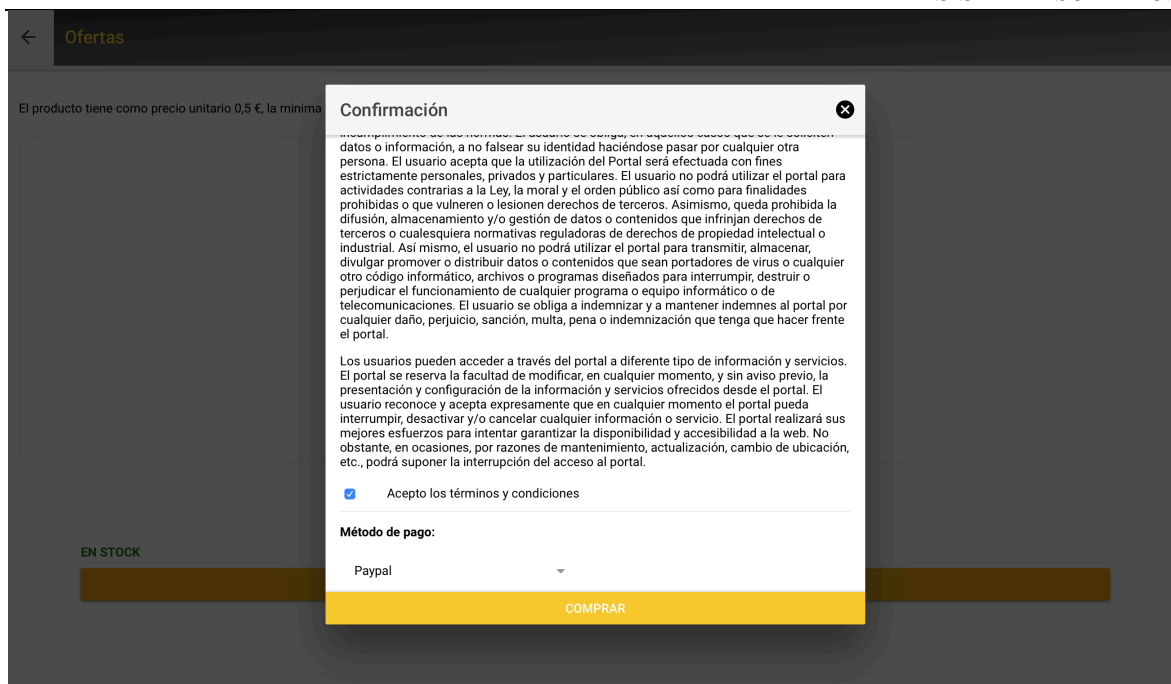


Figura 47: Página vista por el Comprador

El comprador puede acceder a sus pedidos y ver en qué estado se encuentran:

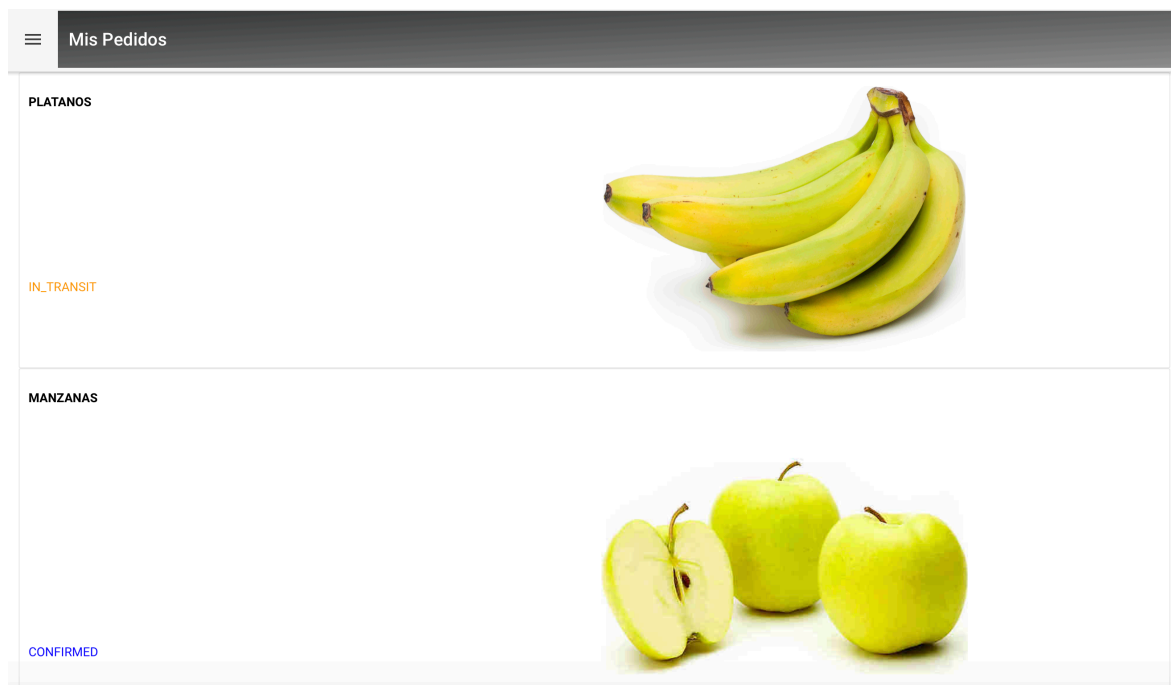


Figura 48: Página vista por el Comprador

Además, si pincha encima del pedido, puede observar el tracking:



El pedido ya está en manos del transportista.

Figura 49: Página vista por el Comprador



El pedido ha sido confirmado por el vendedor y se está preparando para ser entregado al transportista.

Figura 50: Página vista por el Comprador

Si inicia sesión el proveedor de productos, observará los productos que ha comprado el cliente, y podrá confirmar el pedido para que el transportista se ponga en marcha.

Exchange

Bienvenido SALIR G

PLATANOS ID: 1

ID Cliente: 1

IN_TRANSIT CONFIRMAR PEDIDO

FRESAS ID: 2

ID Cliente: 1

CONFIRMED CONFIRMAR PEDIDO

MANZANAS ID: 3

ID Cliente: 1

PENDING CONFIRMAR PEDIDO

Figura 51: Página vista por el Vendedor

El sensor IoT podrá registrar cuando se encuentre el pedido en tránsito, entregado, su temperatura y su posición por GPS. Todas las actualizaciones de tracking son actualizadas a su vez al Comprador y Proveedor.

Exchange

Bienvenido SALIR G

PLATANOS

ID Vendedor: 1

ID: 1

Última temperatura registrada: 20 °C

Última localización registrada: Latitud: 12 , Longitud: 12

IN_TRANSIT

PEDIDO EN TRANSITO PEDIDO EN TIENDA Registra temperatura REGISTRAR °C Latitud Longitud REGISTRAR GPS



MANZANAS

ID Vendedor: 1

ID: 2

Última temperatura registrada: 3 °C



Figura 52: Página vista por el Sensor

ANÁLISIS DE RESULTADOS

Por último, es necesario entender que la aplicación muestra la parte visual, sin adentrarse en las profundidades de Blockchain ni de programación, sin listar ningún resultado de las transacciones de Hyperledger.

Pero si accedemos al sistema de Hyperledger observaremos cómo se han ido haciendo llamadas a las transacciones y se han quedado registradas de manera inmutable y completamente segura, al quedarse registradas como *hashes*:

```
[
  {
    "$class": "org.hyperledger.composer.system.HistorianRecord",
    "transactionId": "00c1b64778f81093f2b7dc021bec49e04f2fc28bbb0fab3f0ad7625b32e40646",
    "transactionType": "org.acme.Login",
    "transactionInvoked": "resource:org.acme.Login#00c1b64778f81093f2b7dc021bec49e04f2fc28bbb0fab3f0ad7625b32e40646",
    "participantInvoking": "resource:org.hyperledger.composer.system.NetworkAdmin#admin",
    "identityUsed": "resource:org.hyperledger.composer.system.Identity#105b617f2846010d715dc95af54908b7a34fcb62e406b97c2e372eef6b2e0e0f",
    "eventsEmitted": [],
    "transactionTimestamp": "2019-07-08T12:26:54.292Z"
  },
]
```

Figura 53: Transacción de Login

```
{
  "$class": "org.hyperledger.composer.system.HistorianRecord",
  "transactionId": "03647b97738df0977adcb7f0a0b8204bc8a38fd78944d7113ee1f681dcb00f0",
  "transactionType": "org.acme.TemperatureReading",
  "transactionInvoked": "resource:org.acme.TemperatureReading#03647b97738df0977adcb7f0a0b8204bc8a38fd78944d7113ee1f681dcb00f0",
  "participantInvoking": "resource:org.hyperledger.composer.system.NetworkAdmin#admin",
  "identityUsed": "resource:org.hyperledger.composer.system.Identity#105b617f2846010d715dc95af54908b7a34fcb62e406b97c2e372eef6b2e0e0f",
}
```

Figura 54: Transacción de Lectura de Temperatura

```
{
  "$class": "org.hyperledger.composer.system.HistorianRecord",
  "transactionId": "07faee510e31b2030b89aa192d9f8d66dbcc7255319e7ddf450271f9dfce573",
  "transactionType": "org.acme.BuyProduct",
  "transactionInvoked": "resource:org.acme.BuyProduct#07faee510e31b2030b89aa192d9f8d66dbcc7255319e7ddf450271f9dfce573",
  "participantInvoking": "resource:org.hyperledger.composer.system.NetworkAdmin#admin",
  "identityUsed": "resource:org.hyperledger.composer.system.Identity#105b617f2846010d715dc95af54908b7a34fcb62e406b97c2e372eef6b2e0e0f",
  "eventsEmitted": [],
  "transactionTimestamp": "2019-07-10T10:00:59.195Z"
},
```

Figura 55: Transacción Compra de Producto

```
{
  "$class": "org.hyperledger.composer.system.HistorianRecord",
  "transactionId": "2abc6865fba974d430818dd59186f3e2f576aecc62086fee581ebf25fab5fd7",
  "transactionType": "org.acme.GpsReading",
  "transactionInvoked": "resource:org.acme.GpsReading#2abc6865fba974d430818dd59186f3e2f576aecc62086fee581ebf25fab5fd7",
  "participantInvoking": "resource:org.hyperledger.composer.system.NetworkAdmin#admin",
  "identityUsed": "resource:org.hyperledger.composer.system.Identity#105b617f2846010d715dc95af54908b7a34fcb62e406b97c2e372eef6b2e0e0f",
  "eventsEmitted": [],
  "transactionTimestamp": "2019-07-11T16:21:59.257Z"
},
```

Figura 56: Transacción de lectura de GPS

Capítulo 7. CONCLUSIONES Y TRABAJOS FUTUROS

Desde su surgimiento en el año 2008, el interés por la tecnología Blockchain ha seguido un ritmo de evolución creciente, como lo demuestra el hecho de que tanto gobiernos como empresas y, muy especialmente, el ecosistema de emprendedores de todo el mundo, han venido realizando continuos esfuerzos e inversiones (en muchos casos millonarias) para desarrollarla y adquirir una posición de privilegio en su aplicación y uso. Gran parte de este interés proviene de su capacidad y potencial inherente para producir disrupción en muchos de los actuales modelos de negocio en todos los sectores de actividad: banca, seguros, energía, utilities, agricultura, etc.

En el presente proyecto se ha profundizado en el estudio, conocimiento y uso de esta tecnología, con el objetivo básico de proporcionar una solución aplicable a un problema de negocio.

En cuanto al estudio de la tecnología, se ha comenzado con el análisis de los conceptos más generales desde su nacimiento, prestando una atención muy especial al actual estado del arte y examinando las diferentes alternativas y posibilidades existentes, para finalmente aplicarlas más específicamente en la concepción, diseño y programación de una solución Hyperledger, con la singularidad que ésta representa con respecto al resto de Blockchains existentes.

Fruto de este enfoque, se ha obtenido una visión integral y profunda de la tecnología Blockchain y, más en concreto, de la solución Hyperlegder, desde lo más básico de ésta hasta su aplicabilidad completa. Así mismo, cabe destacar que se ha conseguido desplegar la red Hyperledger Fabric, se ha generado un API REST y se ha desplegado finalmente un interfaz gráfico de usuario que consigue la interacción del usuario con la red y los contratos inteligentes.

Para el correcto funcionamiento de la red, se han ido realizando pruebas con el interfaz de usuario Hyperledger Composer, que posibilita al programador ir compilando y testeando la

CONCLUSIONES Y TRABAJOS FUTUROS

red, permitiendo la motorización de transacciones y viendo lo que sucede antes de llevarla a Fabric. El siguiente paso ha sido desplegar la red Hyperledger Fabric en local, de manera que se ha generado un API REST que ha permitido por último programar el Front End, el cual, a pesar de lo inicialmente previsto, ha resultado especialmente laborioso y complejo, dados los diferentes problemas de madurez en la configuración y versionado de esta solución.

Con el objetivo de mostrar un resultado final atractivo, amigable y sencillo de usar, se han aplicado diferentes lenguajes de programación como Hyperledger, JavaScript, HTML, CSS, Typescript, JSON, entre otros.

En el presente caso de uso, la aplicación de Blockchain en el mundo empresarial en la cadena de suministro, tal y como se propone en este proyecto, acorta exponencialmente los tiempos de proceso. La tecnología Blockchain sin duda añade eficiencia y productividad a muchos problemas de negocio, simplificando su sistema de contabilización y dotando de inmediatez al propio problema de negocio, al no existir intermediarios.

Sin embargo, y de cara al futuro, esta tecnología debe madurar, ya que existen algunas Blockchains incompatibles con la nueva ley de protección de datos (GDPR). Hay que analizar cuándo y cómo trabajar con datos personales, puesto que, al ser una tecnología inmutable, el usuario se encuentra con ciertas barreras a la hora de gestionar su información. Es por ello que esta tecnología requiere de una actualización constante para estar al día de las novedades y mejoras que van apareciendo.

Como trabajo futuro se podría añadir a este caso de uso un sensor IoT que enviase datos a la API REST en tiempo real, de manera que se consiga hacer una demostración real de una cadena de suministro. Además, como trabajo adicional, se podría levantar una red Hyperledger Fabric en un servidor que formase parte de la Universidad. Esto culminaría el desarrollo de este proyecto y el conocimiento de Hyperledger.

Capítulo 8. BIBLIOGRAFÍA

- [1] “The Internet Of Things,” *European Comission*, 2019. [Online]. Available: <https://ec.europa.eu/digital-single-market/en/internet-of-things>.
- [2] Sachchidanand Singh ; Nirmala Singh, “Internet of Things (IoT): Security challenges, business opportunities & reference architecture for E-commerce,” 2019. [Online]. Available: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7380718&isnumber=7380415>.
- [3] “Qué es el blockchain y cómo funciona,” *I’m Novation*, 2019. [Online]. Available: <https://www.imnovation-hub.com/es/transformacion-digital/que-es-blockchain-y-como-funciona-esta-tecnologia/>.
- [4] “Función hash: concepto y aplicación en Bitcoin,” 2018. [Online]. Available: <https://academy.bit2me.com/como-funciona-el-hash-en-bitcoin/>.
- [5] K. Niebuhr, “Los diferentes tipos de consenso del Blockchain,” 2017. [Online]. Available: <https://www.karlbooklover.com/consensos-del-blockchain/>.
- [6] “Tolerancia a fallas bizantinas,” 2018. [Online]. Available: <https://www.binance.vision/es/blockchain/byzantine-fault-tolerance-explained>.
- [7] “Arquitectura de Hyperledger y consenso.” [Online]. Available: <https://andresarayafalcone.cl/>.
- [8] “Vitalik Buterin.” [Online]. Available: <https://criptotario.com/quien-creo-ethereum-quien-es-vitalik-buterin>.
- [9] R. Vipin Bharathan, Mic Bowman, Sally Cole, Silas Davis, Nathan George, Gordon Graham, Lovesh Harchandani, V. Jagadeesan, Tracy Kuhrt, Stanislav Liberman, Todd Little, Dan Middleton, Hart Montgomery, Binh Nguyen, and G. W. Panicker, Mark Parzygnat, Vitor Quesma, “Hyperledger Architecture, Volume II.”
- [10] “Hyperledger Sawtooth.” [Online]. Available: <https://sawtooth.hyperledger.org/>.
- [11] “Hyperledger Iroha.” [Online]. Available: <https://iroha.readthedocs.io/en/latest/>.
- [12] “Hyperledger Burrow.” [Online]. Available: <https://www.hyperledger.org/projects/hyperledger-burrow>.
- [13] “Hyperledger Indy.” [Online]. Available: <https://hyperledger-indy.readthedocs.io/en/latest/>.
- [14] “Hyperledger Quilt.” [Online]. Available: <https://www.hyperledger.org/projects/quilt>.
- [15] “Hyperledger Explorer.” [Online]. Available: <https://www.hyperledger.org/projects/explorer>.
- [16] “Hyperledger cello.” [Online]. Available: <https://www.hyperledger.org/projects/cello>.

BIBLIOGRAFÍA

- [17] S. Maheshwari, “Aspectos básicos de blockchain: Hyperledger Fabric e Hyperledger Composer,” 2018. [Online]. Available: <https://www.ibm.com/developerworks/ssa/library/cl-blockchain-hyperledger-fabric-hyperledger-composer-compared/index.html>.
- [18] “Docker.” [Online]. Available: <https://www.toolboxtve.com/es/por-que-usar-docker-cuando-la-infraestructura-se-vuelve-codigo/>.
- [19] “Angular.” [Online]. Available: <https://victorroblesweb.es/2017/08/05/que-es-angular-y-para-que-sirve/>.
- [20] “Fundamentos de Git.” [Online]. Available: <https://git-scm.com/book/es/v1/Empezando-Fundamentos-de-Git>.
- [21] “NodeJS.” [Online]. Available: <https://www.netconsulting.es/blog/nodejs/>.
- [22] “Ionic.” [Online]. Available: <https://medium.com/biotec/qué-es-ionic-47e03c0d4b88>.
- [23] B. Cendón, “El Origen Del IoT,” 2017. [Online]. Available: <http://www.bcendon.com/el-origen-del-iot/>.
- [24] O. B. School, “En 2020, más de 30 mil millones de dispositivos estarán conectados a Internet.” [Online]. Available: <https://www.obs-edu.com/es/noticias/estudio-obs/en-2020-mas-de-30-mil-millones-de-dispositivos-estaran-conectados-internet>.
- [25] S. Galeano, “El número de usuarios de Internet en el mundo crece un 9,1% y alcanza los 4.388 millones (2019),” 2019. [Online]. Available: <https://marketing4ecommerce.net/usuarios-internet-mundo/>.
- [26] y A. M. Amitranjan Gantait, Joy Patra, “Cómo implementar blockchain para aplicaciones de IoT cognitivas,” 2018. [Online]. Available: <https://www.ibm.com/developerworks/ssa/cloud/library/cl-blockchain-for-cognitive-iot-apps-trs/index.html>.
- [27] IDGNow, “Bitcoin completa 10 años; Una breve retrospectiva de la criptomoneda más famosa del mundo,” 2017. [Online]. Available: <https://www.assespropr.org.br/es/bitcoin-completa-10-anos-um-breve-retrospecto-da-criptomoeda-mais-famosa-do-mundo/>.
- [28] Hyperledger, “Transaction Flow,” 2019. [Online]. Available: <https://hyperledger-fabric.readthedocs.io/en/release-1.4/txflow.html>.
- [29] H. Composer, “Writing Web Applications.” [Online]. Available: <https://hyperledger.github.io/composer/v0.19/applications/web>.
- [30] “Instalación de Docker y Docker Compose.” [Online]. Available: <https://www.docker.com/products>.
- [31] “Instalación de NodeJS.” [Online]. Available: <https://nodejs.org/en/download/>.

BIBLIOGRAFÍA

- [32] H. Composer, “Installing the development environment.” [Online]. Available: <https://hyperledger.github.io/composer/latest/installing/development-tools.html>.