



Universidad
Zaragoza

Trabajo Fin de Grado

Sistema de control de trazabilidad de la cadena de suministro basado en Blockchain para evitar el fraude.

Blockchain-based supply chain traceability control system to reduce fraud.

Autor

Alejandro Omist Casado

Director

Marcos Rubio Redondo

Ponente

Francisco Javier Zarazaga Soria

Escuela de Ingeniería y Arquitectura

2020/2021

Sistema de control de trazabilidad de la cadena de suministro basado en Blockchain para evitar el fraude

RESUMEN

Blockchain se ha convertido en una tecnología que ha despertado el interés de grandes empresas y administraciones públicas. En el proceso de tomar control de esta tecnología, el presente TFG se aborda como una fase de experimentación dentro de la empresa Inycom. Para ello, se ha buscado un escenario de aplicación en el que pueda cobrar sentido esta tecnología, y éste ha sido de la cadena de distribución de productos digitales. Concretamente en un hipotético escenario de compra-venta de películas.

Se plantea que el uso de la tecnología Blockchain permite reducir el fraude en gran medida mediante el seguimiento de la trazabilidad de un producto; de esta manera, permite registrar todos los procesos por los que ha pasado dicho producto, haciendo que todos estos sean visibles para el usuario. Por estas razones, debido a que en el proyecto se quiere hacer un control de la trazabilidad de películas, ya que es el activo digital más común en la piratería, se ha elegido esta tecnología en este caso de uso.

En este Trabajo de Fin de Grado se ha realizado una pequeña aplicación en la cual distintos usuarios pueden realizar acciones de compra-venta que incluyen seleccionar una película dentro de las que tiene un usuario para ponerla a la venta, deseleccionar una película de las ofertadas para que ya no esté disponible para la venta, y comprar una película que otro usuario esté ofreciendo. Más allá de una funcionalidad avanzada, lo que se persigue es disponer de un ejemplo de uso para la experimentación con Blockchain. En este escenario, la cadena de bloques registra todas las compraventas. Para ello, se verifica que la película a vender es del usuario que la ofrece y, en caso afirmativo, registra el cambio de dueño. La aplicación de Blockchain ha sido realizada mediante Hyperledger Fabric y se encarga de levantar la red, realizar las conexiones, desplegar el *Smart Contract*, realizar la comunicación con la aplicación desarrollada y registrar y almacenar la información.

DECLARACIÓN DE AUTORÍA

TRABAJOS DE FIN DE GRADO / FIN DE MÁSTER



DECLARACIÓN DE AUTORÍA Y ORIGINALIDAD

(Este documento debe remitirse a sacina@unizar.es dentro del plazo de depósito)

D./Dña. **ALEJANDRO OMIST CASADO**

en aplicación de lo dispuesto en el art. 14 (Derechos de autor) del Acuerdo de 11 de septiembre de 2014, del Consejo de Gobierno, por el que se aprueba el Reglamento de los TFG y TFM de la Universidad de Zaragoza,

Declaro que el presente Trabajo de Fin de Estudios de la titulación de Grado en Ingeniería Informática (Título del Trabajo)

Sistema de control de trazabilidad de la cadena de suministro basado en Blockchain para evitar el fraude.

es de mi autoría y es original, no habiéndose utilizado fuente sin ser citada debidamente.

Zaragoza, 25/11/2021



Fdo: **ALEJANDRO OMIST CASADO**

ÍNDICE

Resumen	1
Declaración de autoría	2
Índice.....	3
Agradecimientos	5
1 Introducción.....	6
1.1 Contexto del Trabajo	6
1.2 Contexto Tecnológico	6
1.3 Motivación y problema que se aborda	7
1.4 Alcance, objetivos y limitaciones.....	8
1.5 Herramientas de trabajo	9
1.6 Esquema general de la memoria del proyecto	10
2 Trabajo desarrollado.....	11
2.1 Requisitos del sistema	11
2.2 Arquitectura software del sistema	12
2.3 Diseño del sistema.....	16
2.3.1 Diseño del almacenamiento.....	16
2.3.2 Diseño de la navegación.....	17
2.4 Despliegue del sistema	18
2.5 Dimensión del trabajo realizado.....	21
2.6 Problemas encontrados.....	22
3 Lecciones aprendidas y conclusiones.....	23
3.1 Conocimientos adquiridos	23
3.2 Ideas Futuras	23
3.3 Conclusiones.....	24
4 Bibliografía	25
5 Anexo I. ¿Qué es Blockchain?	27
6 Anexo II. PROBLEMAS DETALLADOS	29
7 Anexo III. AÑADIR PEERS Y ORGS	31
8 Anexo IV. MANUAL DE USUARIO	33



9	Anexo V. CONTROL DE LA TRAZABILIDAD	36
---	---	----

AGRADECIMIENTOS

En primer lugar, me gustaría agradecer a mis profesores su dedicación y tiempo. Gracias a ellos he conseguido tener el conocimiento necesario para poder aprender a hacer una aplicación de este tipo.

En segundo lugar, gracias al equipo de Inycom, en especial a Marcos Rubio y a Rodrigo Casamayor, por darme esta oportunidad y esta experiencia tan enriquecedora y nueva para mí, me he sentido muy apoyado y siempre han encontrado el tiempo necesario para ayudarme siempre que lo he necesitado.

Gracias también a mi ponente Javier Zarazaga por guiarme en este camino.

Y por último, gracias a mi familia y a mis amigos, que para mí son como una segunda familia, por su apoyo incondicional y su soporte en los momentos buenos y en los más difíciles.

1 INTRODUCCIÓN

1.1 Contexto del Trabajo

El proyecto ha sido realizado en la empresa y consultora tecnológica Inycom en el marco de un convenio con la Universidad de Zaragoza dentro del alcance de la Cátedra que esta compañía tiene con la misma.

El TFG ha sido realizado combinando trabajo presencial en las oficinas de la empresa y trabajo remoto en una de las líneas de experimentación tecnológica que actualmente está desarrollando la empresa, el BlockChain. Concretamente, la empresa está orientando esta experimentación hacia la reducción del fraude mediante la trazabilidad de un producto, de tal manera que un cliente o usuario tenga completa disponibilidad del seguimiento de este producto y evitar de esta manera todo tipo de infracción, defecto o engaño.

1.2 Contexto Tecnológico

La tecnología principal que ha sido utilizado en el proyecto es **BlockChain** ya que es una manera segura de almacenar información y de realizar un control de la trazabilidad de un producto, asegurando que no se va a realizar ningún fraude. BlockChain es una tecnología actualmente muy en boga que se presenta en detalle en el [Anexo I](#).

Dentro de las diferentes implementaciones de BlockChain disponibles, en este TFG se ha utilizado **Hyperledger Fabric**, que es una plataforma de tecnología de libro mayor distribuido (DLT) de código abierto y grado empresarial [3]. Se ha escogido la tecnología Hyperledger Fabric en vez de otro tipo de tecnologías tan o más populares populares como Ethereum para desarrollar este trabajo por las siguientes razones:

- Ofrece canales que permiten realizar una comunicación privada y segura entre los nodos de la red.
- Utiliza el formato Protocol Buffer (Protobuf¹) para el intercambio de información, haciendo que se pueda elevar el número de transacciones en un tiempo determinado, aumentando la eficiencia del intercambio.
- Se utilizan organizaciones que emiten sus propios certificados a cada participante o pieza de la arquitectura de su red. Es por esto que una red de Fabric puede tener uno o más MSPs

¹ https://en.wikipedia.org/wiki/Protocol_Buffers

- Los contratos inteligentes se pueden programar en diferentes lenguajes en lugar de tener que aprender arquitecturas y lenguajes personalizados.
- Se establece confianza descentralizada en una red de participantes conocidos en lugar de en una red abierta de participantes anónimos.

El Smart Contract ha sido implementado en **go**², que es un lenguaje de programación concurrente desarrollado por Google inspirado en la sintaxis de C, pero con la dinamicidad de Python y el rendimiento de C++.

La aplicación web de la aplicación ha sido realizada en **Angular**³, desarrollando código en distintos lenguajes como html, typescript, css, javascript etc. Para la comunicación entre la API y el Smart Contract se ha desarrollado una aplicación externa en **javascript** que se comunica con la interfaz mediante **node**⁴, **express**⁵ y **axios**⁶ y con el SmartContract haciendo llamadas a las funciones del contrato cuando estas son requeridas. También se ha utilizado Mongo como repositorio de distintos productos de la aplicación web. Gran parte de las fuentes de la aplicación de BlockChain, como tutoriales, ejemplos, y parte del código obtenido han sido obtenidas de la referencia [3]. También se han tomado como referencia diferentes páginas y repositorios como referencia para desarrollar la aplicación de Angular [13].

1.3 Motivación y problema que se aborda

BlockChain es una tecnología novedosa y puntera cada vez más presente en el día a día de la informática, por lo que aprender cómo funciona y la idea de realizar este proyecto basado en ella fue una gran motivación desde el principio. Así mismo, el hecho de realizar el trabajo con una de las empresas en tecnología más importantes de España, como es Inycom, fue una gran oportunidad para tener una experiencia nueva y acercarse al mundo laboral. Además, el hecho de usar tecnologías y propuestas metodológicas no vistas anteriormente fue un incentivo para aprender nuevas facetas de la informática y ampliar el rango de conocimientos.

BlockChain es una tecnología famosa por ser la base para la implementación de las criptomonedas. No obstante, también tiene muchos otros usos y las empresas tecnológicas más reconocidas, como Microsoft y Amazon, están centrando sus esfuerzos

² <https://es.wikipedia.org/wiki/Go>

³ [https://es.wikipedia.org/wiki/Angular_\(framework\)](https://es.wikipedia.org/wiki/Angular_(framework))

⁴ <https://es.wikipedia.org/wiki/Node.js>

⁵ https://es.wikipedia.org/wiki/NVM_Express

⁶ <https://www.axios.com/technology/>

en aplicarla en sus productos [5]. Uno de sus usos más extendidos es para dar soporte a la trazabilidad de un producto, que es el uso que se le da en Inycom, pudiendo realizar un seguimiento de éste en todo momento, aumentando así la transparencia y reduciendo la posibilidad de fraude.

En un principio, este proyecto iba a estar relacionado con Surefish, que es un proyecto presente en Inycom sobre el uso de BlockChain en la trazabilidad de pescado [6]. No obstante, el propio avance de este producto con la aparición de un potencial cliente para el mismo, hizo que la empresa lo considerara un escenario de riesgo para abordarlo con un TFG y se buscó una aplicación algo distinta. En este punto se propuso el uso de BlockChain en el ámbito de la trazabilidad de activos digitales para evitar su piratería. La piratería de medios digitales es una acción delictiva que incumple los artículos 270, 271 y 272 del Código Penal [1]. Un 29% de la población ha pirateado alguna vez música, un 12% videojuegos y un 32% películas [2]. Concretamente, se planteó su aplicación a procesos de compra-venta entre particulares de películas en formato digital. Mediante BlockChain se puede registrar lo que ha pasado con estos productos en todo momento, impidiendo de esta manera que se produzcan acciones fraudulentas y pudiendo detectarlas más fácilmente en el caso de que estas ocurran.

1.4 Alcance, objetivos y limitaciones

El objetivo de este proyecto ha sido experimentar con BlockChain en el contexto de un problema que pudiera tener continuidad para un desarrollo industrial. En este sentido, y en un análisis previo a la propuesta de TFG, se determinó la utilidad de BlockChain para reducir el fraude mediante un control de la trazabilidad de un producto y, por ello, realizar una aplicación que pueda ser utilizada de manera fácil e intuitiva donde se puedan gestionar activos digitales como películas. Con esta tecnología se dará soporte al registro de distintas acciones del proceso de compra-venta: comprarlos, activar y desactivar la posibilidad de venta.

Una limitación presente en el proyecto es que ha sido desplegado de manera local. Al tratarse de un proyecto de carácter experimental, destinado a mostrar utilidades de la tecnología en un escenario concreto, se determinó con la empresa que no resultaba necesario poner en marcha una red de nodos en múltiples máquinas. Por ello se optó por hacer funcionar varios nodos dentro del portátil de uso para desarrollo que simulan la red.

La nomenclatura empleada en Blockchain es muy específica de esta tecnología. Por ello, se ha optado en esta memoria por incluir un anexo ([Anexo I](#)) con los términos de la misma empleados a lo largo de la memoria.

En lo referente a la aplicación web desarrollada con Angular, su cometido, como se ha mencionado previamente, es dotar de un escenario de experimentación con la tecnología de cadena de bloques. Por ello, su funcionalidad se ha ajustado a lo mínimo necesario para este fin.

1.5 Herramientas de trabajo

Para el desarrollo de este proyecto se han utilizado una serie de herramientas de comunicación entre el alumno y el ponente, el director, y otros miembros de la empresa Inycom; y herramientas técnicas para el desarrollo del proyecto.

Herramientas de comunicación:

- **Google Meet** para realizar reuniones entre el alumno y el ponente. Donde se han tratado temas de los intereses del alumno para realizar el proyecto como búsqueda de una empresa para realizarlo, seguimiento del proyecto y los distintos avances que se han ido realizando.
- **Microsoft teams** para realizar la comunicación con los miembros de Inycom. Se ha utilizado para la resolución de dudas y consultas, para realizar reuniones de seguimiento y para compartir información interesante como comandos y enlaces de utilidad.
- **Gmail** para el envío de mensajes con el ponente, director etc.

Herramientas técnicas:

- **Visual Studio Code** (versión 1.59.1) como entorno para desarrollar el código del proyecto.
- **Sublime Text** (versión Build 4107) para la modificación y visualización de código.
- **Github** para realizar un control de versiones, y como copia de seguridad del proyecto.
- **Docker Desktop** (versión 3.4.0) para permitir el uso de contenedores.
- **Microsoft Word** (versión 16.50) para desarrollar la memoria.

El proyecto se ha desarrollado combinando trabajo remoto y trabajo presencial en las oficinas de Inycom. En ambos casos se ha utilizado el mismo equipo que es:

- Mac Book Pro de 13 pulgadas con chip Apple M1, de 2020 con 16 GB de RAM, y con la versión 11.1 Big Sur

En cuanto a la red se han utilizado las siguientes:

- Red utilizada cuando se ha trabajado remotamente con 153.53 MB/s de bajada y 82.03 MB/s de subida.
- Red de Inycom con 17.70 MB/s de bajada y 21.87 MB/s de subida.

1.6 Esquema general de la memoria del proyecto

La memoria del proyecto empieza con una introducción en la que se contextualiza la situación inicial del proyecto, las tecnologías que se han empleado, los problemas que se abordan, los objetivos que este tiene y las herramientas que han sido empleadas para su desarrollo. Posteriormente, se presentan los requisitos, arquitectura software y diseño del sistema, junto a la planificación del trabajo y los problemas que han sido encontrados a lo largo del este. Más adelante, se explican los conocimientos adquiridos, cuáles son las ideas futuras que se tienen para este proyecto y las conclusiones que se sacan. Finalmente, se incluyen anexos donde se muestra información de interés que no ha sido presentada anteriormente.

En el [Anexo I](#), se describen algunos de los conceptos básicos de Blockchain y de la plataforma elegida para su uso: HyperledgerFabric. En el [Anexo II](#), se explican algunos problemas, a nivel técnico, que ha habido a lo largo el proyecto y cómo se han resuelto. En el [Anexo III](#), se explica una manera para poder añadir nodos y organizaciones a la red de Blockchain, en el [Anexo IV](#) se muestra un manual donde se explican las opciones disponibles y qué hace cada una de ellas y en el [Anexo V](#) se puede ver cómo se realiza un seguimiento de la trazabilidad del producto.

2 TRABAJO DESARROLLADO

2.1 Requisitos del sistema

Se va a presentar a continuación una lista con los requisitos funcionales y no funcionales que se encuentran en la aplicación:

Requisitos funcionales de la aplicación de experimentación:

- Un usuario puede acceder al sistema con nombre de usuario y la contraseña. El proceso de registro de usuarios ha quedado fuera del alcance de la aplicación y se lleva a cabo directamente sobre la base de datos.
- Una vez el usuario ha accedido a la aplicación, se le mostrarán las películas que otros usuarios ofertan para poder comprarlas.
- Se podrá filtrar el listado de películas para facilitar la localización de las mismas por su título.
- Pulsando el botón COMPRAR de una película, el usuario pasará a ser dueño de dicha película. Esta funcionalidad necesitará de la validación de la propiedad por parte del sistema de Blockchain y del posterior registro del cambio de propiedad de la misma.
- Pulsando el botón MOSTRAR TUS PELÍCULAS, la aplicación accederá a la sección de películas propias y mostrará al usuario las películas que posee.
- Haciendo doble clic en una película de la cual el usuario es dueño, se puede activar o desactivar la posibilidad de que esa película esté a la venta. De tal manera que, si esta opción se encuentra desactivada, la película no aparecerá al resto de usuarios para que puedan comprarla.
- Si el usuario clic en el icono que se muestra como una cruz roja en la parte de arriba a la derecha de una película que sea suya, la película es eliminada y la aplicación la borrará.
- Pulsando el botón AÑADIR, la aplicación muestra la posibilidad de crear una película nueva.
- Si se rellenan los campos necesarios para añadir una película nueva y se pulsa el botón GUARDAR PELÍCULA, la aplicación añade una película nueva donde el usuario que la ha creado es su propietario.
- Al crear una película, la aplicación le asigna un identificador único y se registra en el Blockchain como de propiedad del usuario que la ha creado. Con esta funcionalidad se simula el proceso de adquisición por primera vez de una película que, en un sistema real, se debería conectar con algún sistema de venta (por ejemplo Amazon).
- Pulsando el botón MOSTRAR PELÍCULAS DISPONIBLES PARA COMPRAR, la aplicación mostrará al usuario las distintas películas que puede comprar.

Requisitos no funcionales:

- Al activar la compra de una película, se mostrará una línea verde en la parte izquierda de la película y esta desaparecerá si la opción se desactiva.
- Al pulsar el botón AÑADIR, aparecerá la opción de rellenar los distintos campos para crear la película, y el botón cambiará su color a rojo y su texto cambiará a CERRAR.
- Al pulsar el botón CERRAR, desaparecerá la opción de rellenar los distintos campos para crear la película, y el botón cambiará su color a verde y su texto cambiará a AÑADIR.
- Una película es identificada por un identificador, y también posee los siguientes campos: título, año de creación, duración, país, director, actores, puntuación, foto, propietario y un atributo que indica si la película está a la venta o no llamado *reminder*.
- Un usuario es identificado por un identificador y también posee los campos de nombre de usuario y contraseña.

2.2 Arquitectura software del sistema

Para entender la arquitectura del sistema desarrollado, primero hay que entender la arquitectura que posee BlockChain. Como se explica en el [Anexo I](#), BlockChain se compone de distintos bloques que almacenan información que están conectados entre sí. Además, como se aprecia en el ejemplo que se muestra en la Figura 1, la red está formada por organizaciones, orderers, nodos y más componentes. Todo ello es fundamental para su comunicación, tanto interna como externa.

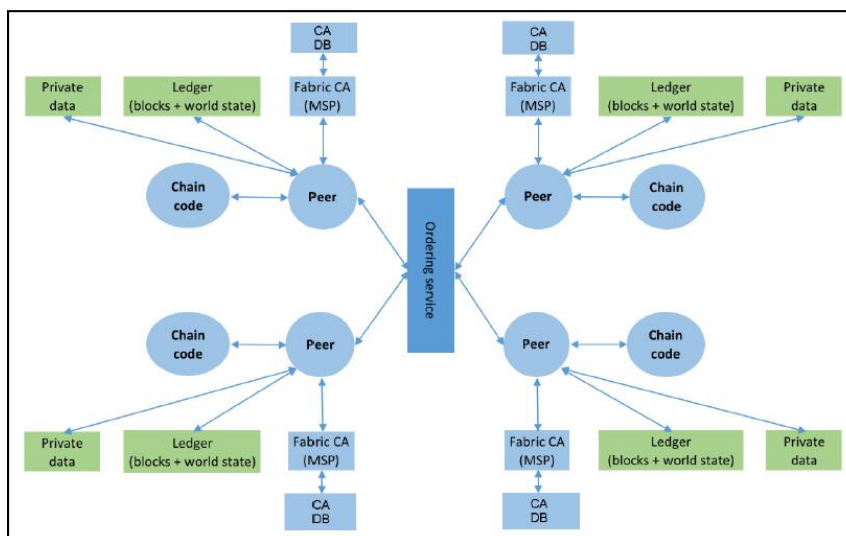


Figura 1: Estructura de orderers, peers etc. [12]

La arquitectura está integrada por la aplicación web que ha sido desarrollada con Angular, la red de Blockchain y una API de acceso a la misma que es la que se ubica entre la red y la aplicación. Esta API se encarga de transmitir la información que recibe la web a la red de Blockchain. Tal y como se ha indicado en la sección anterior, la red de Blockchain se ha desplegado con 2 nodos (peer0.org1 y peer0.org2) que hacen la función de usuarios en la red. Cada uno de ellos almacena una copia completa de la red, y para que exista una modificación de la misma debe de haber una mayoría de los nodos que acepten dicho cambio. En este caso, los 2 nodos deberían aprobar el cambio para que este ocurra, y después de eso, actualizarían la copia que poseen de la red incluyendo el cambio realizado. Junto con ellos se incluye un orderer, que se encarga de coordinar la ejecución de las transacciones, comunicándose con los diferentes nodos implicados, y un SmartContract, que contiene las funciones necesarias para poder realizar los cambios en la red. Esta arquitectura puede verse reflejada en el diagrama de componentes de la Figura 2.

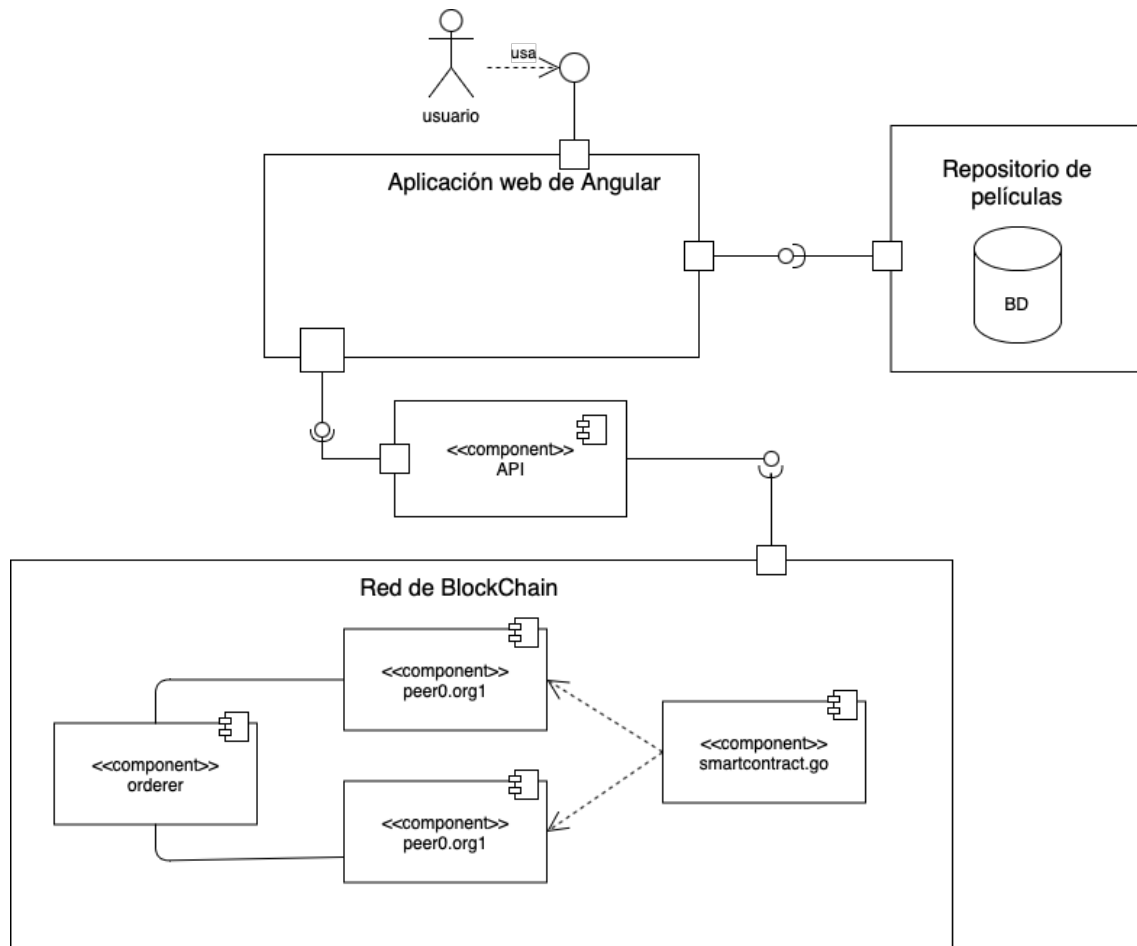


Figura 2: Diagrama de componentes

Para que pueda entenderse mejor la arquitectura del sistema, se ha realizado un ejemplo de una operación de compra de una película realizada por un usuario. (Figura 3). Cuando el usuario se encuentra en la pantalla de *login*, introduce su nombre de

usuario y su contraseña. A continuación, la aplicación realiza la conexión con el servidor donde se encuentran todas las películas almacenadas y son devueltas mediante una petición http a la aplicación web, que mostrará al usuario las películas que se encuentran disponibles para su compra. Cuando el usuario, pulsa el botón de comprar una película, la aplicación web se conecta con la API para comprobar si en la red de BlockChain se encuentra dicha película con el usuario que se indica como propietario. En el caso de que esta validación resulte exitosa, la aplicación web vuelve a hacer una petición http al servidor del repositorio con dicha actualización. El servidor vuelve a mandar las películas con la actualización a la aplicación, que, a continuación, mostrará al usuario las películas que tiene disponibles para comprar con la actualización reciente, donde la película que acaba de comprar ya no aparecerá.

Después de haberse realizado lo anterior, la aplicación se comunicará mediante otra llamada HTTP con la API que tiene conexión con la red de BlockChain, enviando la información respectiva de la transacción de compra que se ha hecho recientemente. La función de esta aplicación ahora es enviar una solicitud de aceptación a los nodos peer0.org1 y peer0.org2, que son los que se encuentran en el canal “mychannel”.

Los nodos ejecutan una copia del chaincode que ellos mismos poseen, simplemente para ver la respuesta y poder aceptar la solicitud y firmarla o no hacerlo y, cuando han obtenido esta respuesta, se la comunican a la aplicación. Si se ha llegado a un consenso, la aplicación se comunica con el orderer, que creará un bloque que ha sido validado (en caso de que se haya llegado a un consenso). Este bloque será enviado por el orderer a peer0.org1 y peer0.org2, que lo integrarán en la copia que poseen de la cadena de bloques.

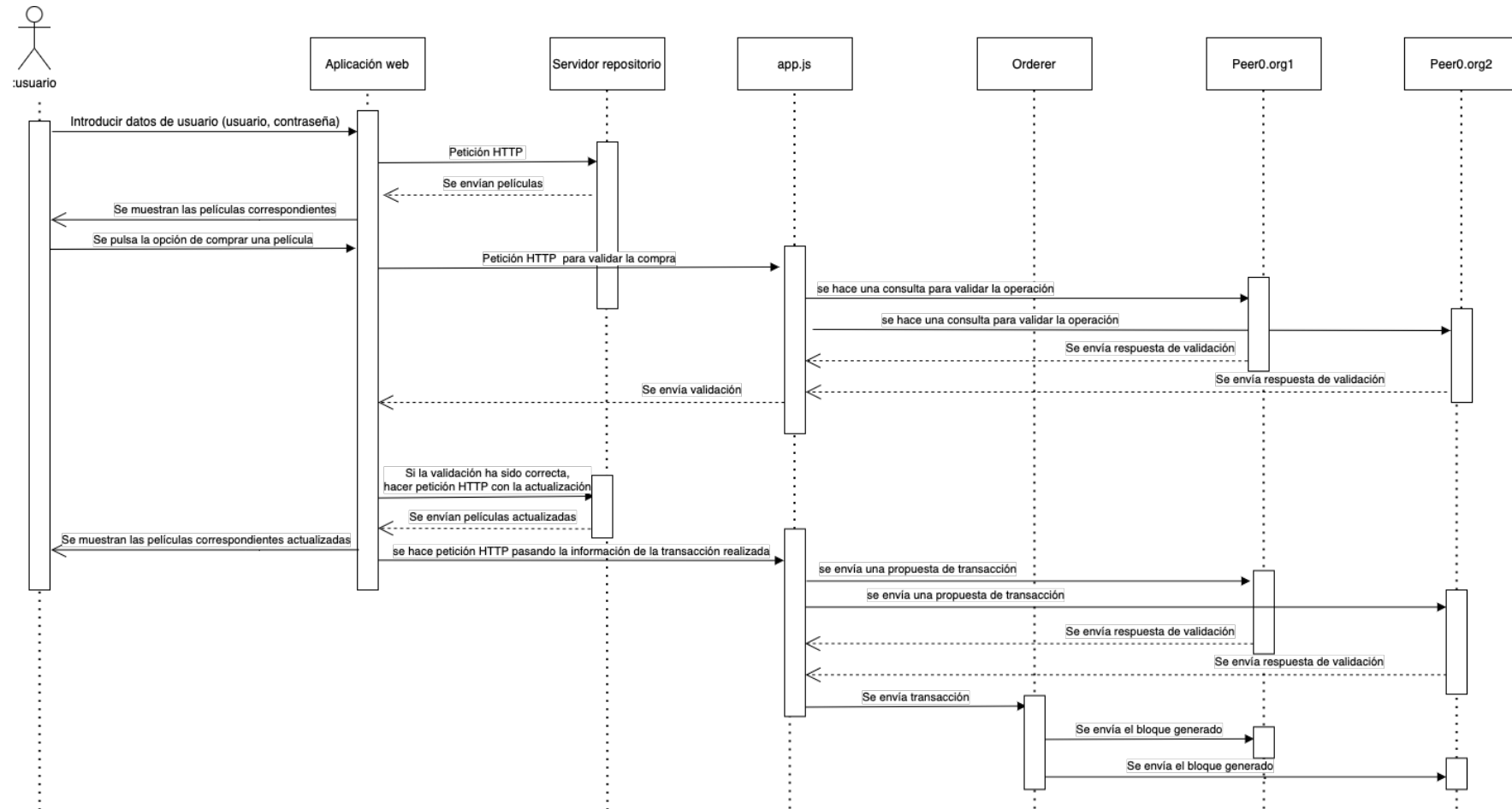


Figura 3: Diagrama de secuencias al comprar una película de manera exitosa

2.3 Diseño del sistema

2.3.1 DISEÑO DEL ALMACENAMIENTO

Como se aprecia en la Figura 4, hay 2 tipos de entidades principales: los usuarios y las películas, ambos se identifican con un identificador único. Los usuarios pueden realizar una serie de acciones con las películas que son distintas si dicho usuario es propietario de la película. Si el usuario no es dueño de la película, puede comprarla o realizar una búsqueda sobre las películas que puede comprar. En el caso de que el usuario sea dueño de una película, puede eliminar dicha película, poner en venta esa película o deshabilitar dicha opción, puede realizar una búsqueda sobre sus películas y puede añadir una película nueva en la que el propio usuario es el propietario.

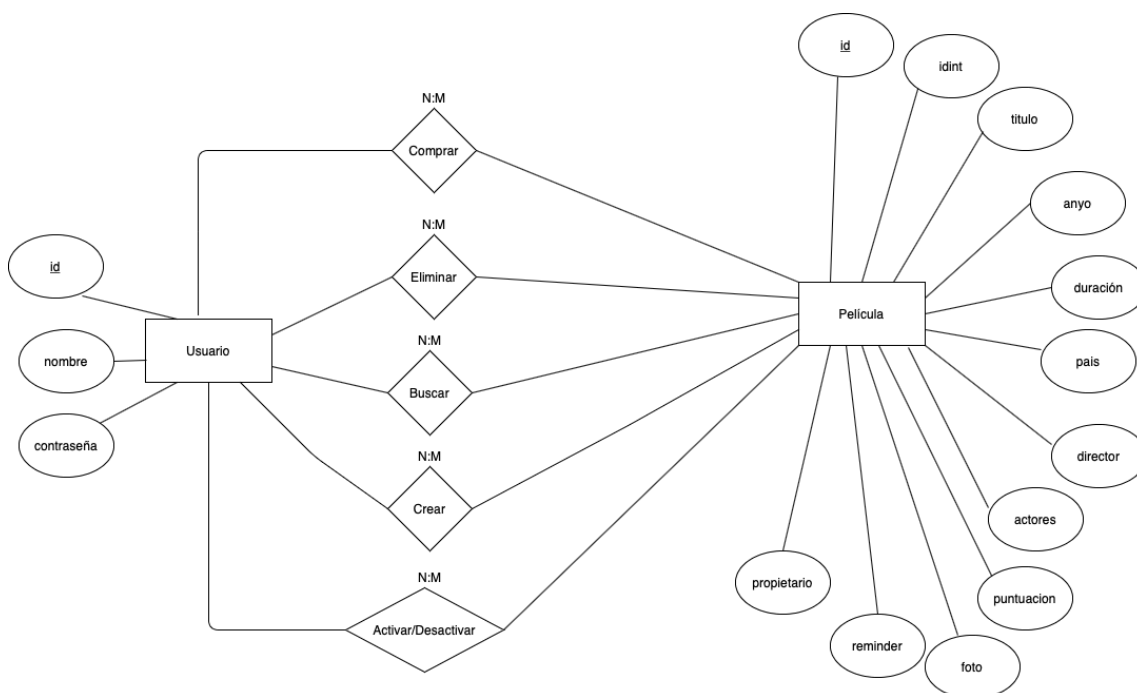


Figura 4: Diagrama de entidad-relación

Todas las acciones que ocurren en la aplicación se almacenan también en la red de Blockchain, donde además se registra cualquier cambio que está sucediendo. Los 2 nodos de la red, que son peer0.org1 y peer0.org2, mantienen una copia de la información actualizada, de tal manera que cuando ocurre algún cambio en la red, lo modifican en su copia de manera simultánea. Para poder comprender cómo funciona el almacenamiento en la red de Blockchain, se debe entender cómo funciona en Hyperledger Fabric el sistema de almacenamiento.

Existen 3 principales elementos relacionados con almacenamiento en Hyperledger Fabric: la cadena de bloques o “blockchain”, el “ledger” y el “world state” [5]. El ledger consta de 2 partes, que son el blockchain y el world state.

- El **world state** es una base de datos que contiene los valores actuales de un conjunto de estados del ledger y sirve para facilitar a un programa el acceso directo al valor actual de un estado en lugar de tener que calcularlo recorriendo todo el registro de transacciones. Puede cambiar con frecuencia, ya que los estados pueden ser creados, actualizados y eliminados. Las opciones para la base de datos del world state actualmente incluyen LevelDB y CouchDB.
- **LevelDB** es la base de datos por defecto, que es la utilizada en el proyecto, y es particularmente apropiada cuando los estados del ledger son pares clave-valor simples. Una base de datos LevelDB está ubicada junto con los 2 nodos de la red, es decir, está incrustada en el mismo proceso del sistema operativo.

La cadena de bloques o **blockchain** se compone de una serie de transacciones que registra todos los cambios que han dado lugar al estado en el que se encuentra el world state, es decir, registro todas las versiones anteriores de cada estado del ledger y cómo se ha modificado. Como su nombre indica se compone de una cadena de bloques (este es el motivo del nombre de la tecnología) cuyo primer bloque se llama “génesis block”. El resto de los bloques recogen las transacciones que suceden en la red, que se añaden a la cadena de bloques, lo que permite comprender el historial de cambios que han dado lugar al estado actual del mundo. La estructura de datos del blockchain es muy distinta a la del world state porque, una vez que es escrita, no puede modificarse; es inmutable.

2.3.2 DISEÑO DE LA NAVEGACIÓN

Como se puede apreciar en el mapa de navegación de la Figura 5, en el inicio de la aplicación aparece la opción de “loguearse”, donde el usuario introducirá el nombre de usuario y la contraseña. Una vez se haya realizado dicha acción se mostrará la pantalla donde se muestra el catálogo de películas disponibles para comprar. En esta sección se pueden hacer diferentes cosas como comprar una película o buscar una película.

Si se accede a la opción TUS PELÍCULAS, se muestran las películas que posee el usuario y en esta sección se podrán hacer diferentes acciones como crear una película, realizar una búsqueda de las películas según su título, eliminar una película y activar o desactivar su venta.

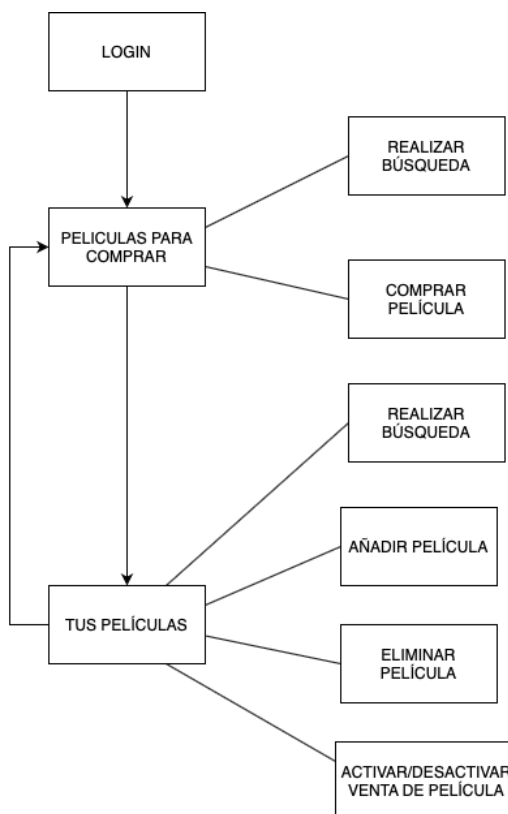


Figura 5: Mapa de navegación

Se puede ver un ejemplo de la navegación en el manual de usuario que se encuentra en el [Anexo IV](#).

2.4 Despliegue del sistema

Un primer trabajo de despliegue de cualquier sistema que opera con BlockChain es el establecimiento de la red con la que se va a trabajar. Si ésta ya existe, habrá que tenerla identificada para abordar los trabajos de despliegue de la siguiente sección. Si no es así, hay que llevar a cabo los siguientes trabajos.

Lo primero es descargar los archivos relacionados con la red de BlockChain [10] en la máquina local. De esta manera, se pueden encontrar los archivos necesarios para lanzar la red, diferentes tipos de SmartContracts, diversas aplicaciones en distintos lenguajes de programación para utilizar las funcionalidades de los SmartContracts etc. Posteriormente, hay que crear una sección nueva (asset-transfer-item), que es la relacionada con el proyecto, con su correspondiente SmartContract correctamente desarrollado.

Una vez que la configuración de la red ha sido creada, se procede a desplegarla en cada una de las máquinas locales mediante el comando `network.sh up createChannel-ca`. De esta manera se ha desplegado la red que se ha configurado anteriormente, en el caso

del proyecto, se despliega una red con un orderer, 2 organizaciones de las cuales cada una de ellas posee un nodo (peer0.org1 y pero.org2). Además de lo anterior, también se instala una Autoridad de Certificación (CA) por cada nodo de la red y por el orderer, para emitir certificados y poder realizar firmas electrónicas, y un canal que comunica a los miembros de la red llamado “mychannel”. La tecnología seleccionada ofrece varias configuraciones de red por defecto para algunos de los escenarios más habituales. Tras una revisión de las mismas y diversas pruebas que se han realizado sobre la configuración de la red ([Anexo III](#)), se ha optado por esta configuración que ofrece elementos suficientes para el sistema de pruebas de este TFG, además de dejar disponibles algunos componentes para experimentaciones específicas que quedarían fuera del alcance de este trabajo final de estudios.

```
// SmartContract provides functions for managing an Asset
type SmartContract struct {
    contractapi.Contract
}

// estructura de un Item
type Item struct {
    Id          string `json:"id"`
    Idint       int    `json:"idint"`
    Titulo      string `json:"titulo"`
    Anyo        string `json:"anyo"`
    Duracion    string `json:"duracion"`
    Pais        string `json:"pais"`
    Director    string `json:"director"`
    Actores     string `json:"actores"`
    Puntuacion  string `json:"puntuacion"`
    Foto        string `json:"foto"`
    Reminder    bool   `json:"reminder"`
    Propietario string `json:"propietario"`
}

//función Buy
func (s *SmartContract) Buy(ctx contractapi.TransactionContextInterface, id string, nuevoPropietario string) error {
    item, err := s.ReadItem(ctx, id)
    if err != nil {
        return err
    }
    item.Propietario = nuevoPropietario
    itemJSON, err := json.Marshal(item)
    if err != nil {
        return fmt.Errorf("failed to read the copy from world state: %v", err)
    }
    return ctx.GetStub().PutState(id, itemJSON)
}
```

Figura 6: Ejemplo de Smartcontract

El siguiente paso es desarrollar el SmartContract, que servirá para poder realizar una comunicación con la red de BlockChain y poder realizar cambios en ella. En el caso del proyecto, se ha desarrollado un SmartContract que contiene las funciones de crear una película, eliminarla, comprarla por otro usuario y poder realizar validaciones sobre si se permite realizar una operación. Este SmartContract ha sido realizado en go, pero se puede desarrollar en diferentes lenguajes como son javascript o typescript. En la Figura 6, se muestra un ejemplo de un SmartContract desarrollado en go, donde se puede ver la función *Buy*, donde se busca un ítem que se haya almacenado con el identificador que

se proporciona como parámetro de la función, se cambia el campo propietario por el proporcionado y se almacena nuevamente el ítem con la actualización realizada. Como se puede ver, la función Buy pertenece al SmartContract que corresponde y para realizar una modificación en la red de BlockChain, como puede verse en el último return de la función, se realiza una validación por parte de los integrantes de la red, donde si no se llega a un consenso entre ellos, aparecerá un error. Una vez se haya desarrollado el SmartContract, se procede a instalar en los nodos de la red mediante el comando `network.sh deployCC`.

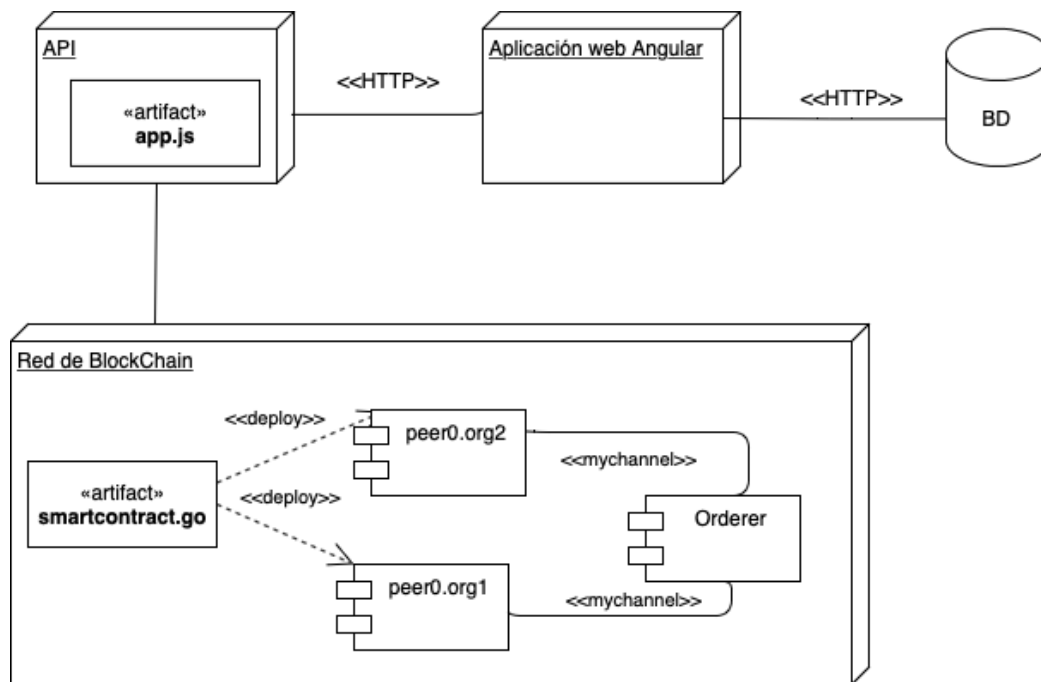


Figura 7: Diagrama de despliegue

A continuación, se debe desplegar la API que conecta la aplicación web con la red de BlockChain mediante el comando `node app.js`, y por último, lanzar la aplicación web. Para ello primero se debe desplegar el servidor que servirá como repositorio de películas, el servidor donde se encuentran los usuarios registrados y la aplicación que ha sido desarrollada mediante Angular a través del comando `npm start`. Siguiendo estos pasos, se pone en marcha la red e, introduciendo la dirección en la que se ha desplegado la aplicación web, se accede a la misma y ya se puede empezar a ejecutar las distintas funciones posibles.

2.5 Dimensión del trabajo realizado

Como ya se ha comentado, el proyecto se ha desarrollado con la empresa Inycom, con trabajo de media jornada.

La planificación inicialmente propuesta era:

- Aprovisionamiento de la infraestructura de tecnologías de la información y la comunicación (ICT) (mes 1, mes 2)
- Despliegue de la plataforma Blockchain (mes 1, mes 2, mes 3)
- Diseño de la API (mes 2)
- Implementación de la API (mes 2, mes 3, mes 4)
- Definición y desarrollo de los contratos inteligentes (mes 2, mes 3, mes 4)
- Testing unitario de la API en casos de uso (mes 3, mes 4, mes 5)
- Diseño de los interfaces básicos (mes 4)
- Desarrollo de los interfaces básicos (mes 4, mes 5)

Tareas	Mes 1	Mes 2	Mes 3	Mes 4	Mes 5
Aprovisionamiento de la infraestructura ICT					
Despliegue de la plataforma Blockchain					
Diseño de la API					
Implementación de la API					
Definición y desarrollo de los contratos inteligentes					
Testing unitario de la API en casos de uso					
Diseño de los interfaces básicos					
Desarrollo de los interfaces básicos					

Figura 8: Planificación inicial

Como se indica en el apartado 1 (aprovisionamiento de la infraestructura ICT), se tuvo que hacer un trabajo de aprendizaje sobre estas nuevas tecnologías. Primero, entendiendo los conceptos básicos y todas las nociones de su funcionamiento. Posteriormente, realizando todos los tutoriales que aparecen [7]. Finalmente, haciendo distintas pruebas como puede ser crear distintos peers y orgs, probar los distintos Smart contracts que hay, probar aplicaciones en distintos lenguajes, realizar modificaciones, entender los conceptos más profundos etc.

Al final, este trabajo de aprendizaje, investigación y pruebas se demoró más de lo previsto, de ese modo, se empezó a crear la plataforma del proyecto más tarde de lo que se había planeado y al final se ha conseguido realizar la aplicación con más carga al final del proyecto de la que se había previsto. Por lo tanto, se han realizado unas 350 horas de trabajo estipuladas más 100 horas de trabajo extra, con lo que hace un total de 450 horas.

Tiempo de dedicación al proyecto

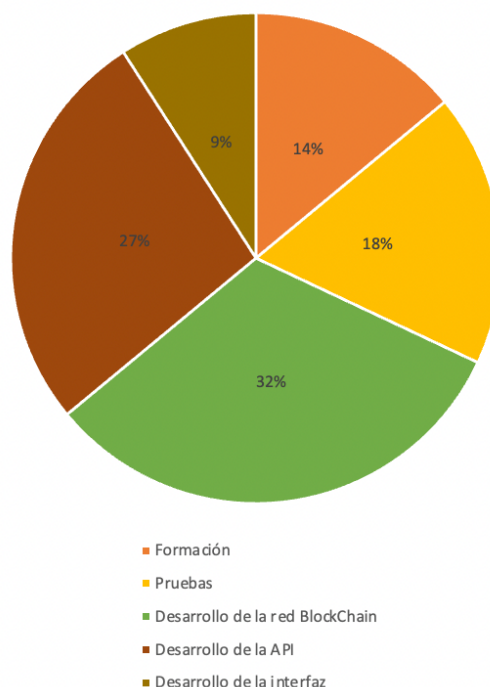


Figura 9: Gráfico de tiempo de dedicación al proyecto

La carpeta fabric-samples contiene 8168 ítems, de los cuales, no todos son utilizados en el trabajo, pero una parte considerable de ellos son empleados. Las líneas de los ficheros de la web, la API, y el SmartContract, que son los que han sido desarrollados principalmente, constan de 38.090 líneas. En la carpeta app-interface, donde se encuentra la aplicación desarrollada con Angular, se encuentran 44.552 ítems distintos.

2.6 Problemas encontrados

La tecnología involucra una gran cantidad de paquetes software, que a veces no funcionaban correctamente. La documentación, en ocasiones, no precisaba la solución necesaria para algunos errores que ocurrían, por lo que se ha tenido que buscar en distintas páginas cómo poder solucionar estos problemas. En el [Anexo II](#), se da más detalles sobre algunos problemas técnicos y cómo se consiguió solucionarlos.

Para realizar la conexión entre la aplicación web desarrollada con Angular y la API que se conecta con la red de BlockChain, se han encontrado algunos errores relacionados con CORS [14] por intentar acceder a una dirección de la cual no se tienen permisos.

3 LECCIONES APRENDIDAS Y CONCLUSIONES

3.1 Conocimientos adquiridos

El proyecto ha servido como una primera experiencia con una novedosa tecnología. Se ha visto por primera vez lo que es la tecnología BlockChain, entendiendo su importancia, utilidad, y el papel que puede jugar en su aplicación en un caso práctico concreto. Se han estudiado los elementos que componen la red, que eran complemente desconocidos previamente. Se ha aprendido a poder modificar la red, poder desarrollar un SmartContract y crear una aplicación que se pueda conectar con el SmartContract. Estamos ante una tecnología que en muchos foros se plantea como de gran futuro. Sin embargo, ni los conceptos de base, ni las propuestas tecnológicas para implementarlos que existen, se ven en ninguna de las asignaturas cursadas. Es por ello que me ha resultado un escenario completamente nuevo al que he tenido que adaptarme.

También se ha desplegado una API en un servidor, algo que se vio en un marco de trabajo de asignatura, pero que fue responsabilidad de otro integrante del equipo por lo que, realmente, el acceso a este problema fue de manera indirecta.

Se han utilizado nuevas tecnologías como Docker, Express, Angular e HyperLedger Fabric que no se habían utilizado antes. Además, se ha aprendido a conectar diferentes componentes a través de la API: mediante express y axios (en el caso de los archivos html) y mediante conexiones con el SmartContract (mediante el uso de “contract”) de la red de BlockChain.

Además, el hecho de trabajar en una empresa, y estar involucrado en un proyecto que pueda ser llevado a cabo en el mundo laboral, ha sido un gran aprendizaje.

3.2 Ideas Futuras

Este proyecto tiene margen de mejora para aumentar su escalabilidad. Por ejemplo, que sea desplegado en un servidor público, en vez de una máquina local, para que pueda ser accesible por todos los usuarios. También se podría mejorar la aplicación web, para que pueda tener funcionalidades más complejas. Todo esto ayudaría a aumentar el número de participantes de la red de BlockChain.

Realizando algunas modificaciones como las mencionadas con anterioridad, el proyecto podría tener un fin comercial. Además, los conocimientos adquiridos podrían ser

utilizados para realizar acciones similares, como podría ser, por ejemplo, el proyecto Surefish [6] o algún otro proyecto tanto de la empresa de Inycom o ajeno a ella.

3.3 Conclusiones

La parte que más complicada del proyecto ha sido familiarizarse con la tecnología, para poder llevar a cabo todos los cambios que han sido requeridos. También ha sido tediosa, en algunos casos, la tarea de intercambiar información entre los distintos componentes de la aplicación.

Se ha conseguido el principal objetivo del proyecto, que era el uso de la tecnología Blockchain para el control de la trazabilidad del producto y con ella, poder disminuir el fraude. El avance y la experimentación sobre esta tecnología tan presente a día de hoy y con gran proyección, ha resultado muy interesante y útil de cara a un futuro tanto laboral como personal.

Desde el principio se pensó que para que esto pudiera ser interactivo, en la etapa final del proyecto, se desarrollaría una aplicación web con una interfaz que permita interactuar a un usuario con las distintas funcionalidades que hay; y finalmente es lo que se ha conseguido.

Como **conclusión personal**, me ha gustado aprender todas estas tecnologías nuevas que desconocía, así como familiarizarme un poco más con la programación web. Además, la experiencia de tener un estrecho contacto con el mundo laboral y ver cómo funciona una empresa desde su interior, ha sido un conocimiento igual de valioso que el resto de aspectos aprendidos a nivel técnico.

4 BIBLIOGRAFÍA

- [1] Código Penal. <https://www.conceptosjuridicos.com/codigo-penal/>
- [2] Noticia sobre piratería.
<https://www.elmundo.es/cultura/2020/03/19/5e735dcd21efa0bb5b8b45cc.html>
- [3] Hyperledger Fabric.
<https://hyperledger-fabric.readthedocs.io/es/latest/whatis.html>
- [4] Organizaciones y nodos(peers).
<https://babel.es/es/Media/Blog/Agosto-2019/Hyperledger-Fabric-Conceptos-y-tipos-de-nodos>
- [5] Empresas que utilizan BlockChain
<https://blog.enzymeadvisinggroup.com/blockchain-2020-usos-y-aplicaciones>
- [6] Proyecto Surefish
<https://es.cointelegraph.com/news/european-project-surefish-to-use-blockchain-for-fish-traceability>
- [7] Tutoriales Hyperledger fabric
<https://hyperledger-fabric.readthedocs.io/es/latest/tutorials.html>
- [8] Rosetta <https://support.apple.com/es-es/HT211861>
- [9] Nicepage <https://nicepage.com>
- [10] Descarga de los archivos de BlockChain
<https://hyperledger-fabric.readthedocs.io/es/latest/install.html>
- [11] Explicación de Ledger
<https://hyperledger-fabric.readthedocs.io/es/latest/ledger/ledger.html>
- [12] Imagen estructura Hyperledger Fabric
<https://docplayer.es/130348840-Workshop-hyperledger-fabric.html>

[13] Referencias tomadas en la aplicación de Angular

<https://docplayer.es/130348840-Workshop-hyperledger-fabric.html>

<https://github.com/bezael/filterPipe>

<https://codingpotions.com/angular-login-sesion>

[14] Errores de Cors

<https://www.digital55.com/desarrollo-tecnologia/problemas-de-cors-y-algunas-soluciones-para-ionic/>

5 ANEXO I. ¿QUÉ ES BLOCKCHAIN?

BlockChain, como su nombre indica, consiste en una cadena de bloques conectados entre sí que contienen información. Cada bloque tiene su contenido, su identificador o hash y el hash del bloque anterior. Esta es la manera en la que los bloques de la cadena están conectan.

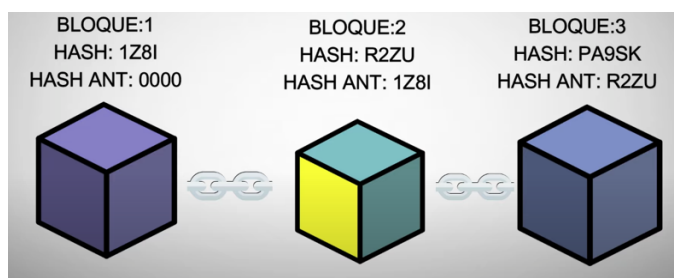


Figura 10: Bloques de BlockChain

El hash se genera según el contenido del bloque, por lo que, si se cambia el contenido de un bloque, cambia el hash, y esto haría que cambiaran todos los bloques siguientes. Por lo que, si sucede esto, la cadena queda invalidada. Es por esto por lo que es tan difícil de modificar el contenido de la cadena.

Además, los distintos usuarios que son partícipes en la red de BlockChain en cuestión, poseen una copia de esta. Los usuarios, además, dan la seguridad y la certificación de los documentos, ya que cuando se añade un nuevo bloque, este tiene que conseguir la validación que se obtiene mediante el consenso de los usuarios de la red.

De esta forma, si un usuario altera la información de su copia, toda la comunidad es consciente y, su copia quedaría anulada.

-Smart Contract

Otro elemento importante son los “SmartContracts”, también llamados “Chaincodes”, que son programas informáticos, que pueden ser desarrollados en distintos lenguajes de programación como go, javascript, typescript etc.

Estos programas se encargan de la ejecución de acuerdos que han sido configurados y son utilizados para añadir información a la cadena de BlockChain.

-Canales

La plataforma de Blockchain que ha sido elegida es Hyperledger Fabric [3]. Hyperledger Fabric posee canales.

En los canales, los usuarios que participan en una red de Fabric establecen una subred donde cada miembro tiene visibilidad de un conjunto de transacciones en particular. De esta manera, solo los nodos que participan en un canal pueden acceder al SmartContract y a los datos transados, conservando la privacidad y la confidencialidad ambos [3].

-Nodos, orderers y orgs

Además, la red está compuesta por distintos nodos, también llamados peers [4], y organizaciones. Hay distintos tipos de nodos en los que destaca el Orderer y los anchor peers. Cada nodo tiene una copia del “ledger”, que se compone de transacciones y datos (Transaction log y state DB).

El **orderer** se encarga de la comunicación; de mantener la consistencia entre los distintos nodos de la red y de asegurar el orden de transacciones. Los orderers se comunican con los **anchor peers**, que son un tipo especial de nodos puesto que son los únicos conocidos fuera de su organización.

Las **organizaciones** se componen de un conjunto de peers. Cada organización debe tener al menos un anchor peer; y el resto de nodos que no sean ni anchor peer ni orderer, no tienen comunicación con ningún otro nodo que no esté en su propia organización.

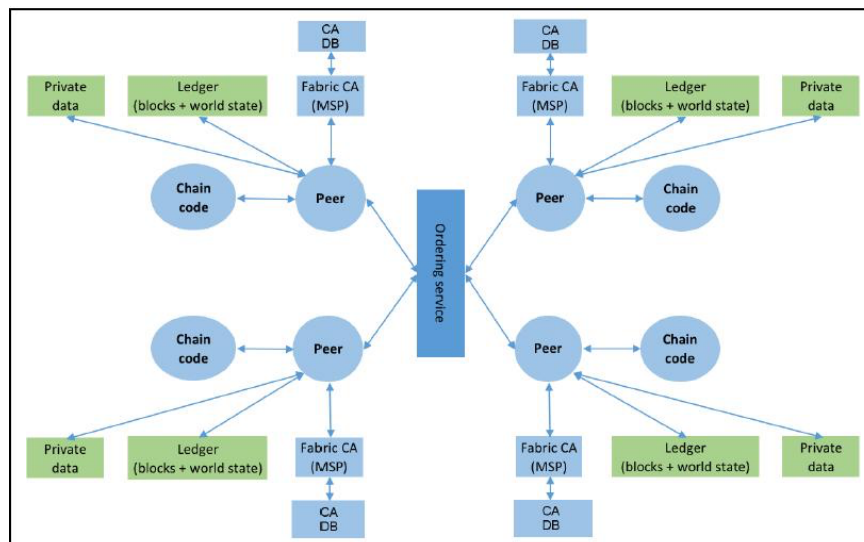


Figura 1: Ejemplo de la comunicación interna de una red de Hyperledger Fabric [12]

6 ANEXO II. PROBLEMAS DETALLADOS

- Al crear la red, aparecía el siguiente error:

```
ERROR: manifest for hyperledger/fabric-orderer:latest not found: manifest unknown:
manifest unknown
```

También ocurría lo mismo con fabric-orderer y fabric-peer. Esto se debe a que no existe la etiqueta “latest” en la mayoría de las imágenes fabric, por lo tanto, hay que especificar cuál es la versión de la imagen de la que se quiere hacer “pull”. Esto se consigue introduciendo los siguientes comandos:

```
docker pull hyperledger/fabric-orderer:2.3.2
```

```
docker tag hyperledger/fabric-orderer:2.3.2 hyperledger/fabric-orderer:latest
```

Lo mismo serviría para fabric-orderer y fabric-peer cambiando fabric-orderer por los mencionados anteriormente. En el caso mostrado, se usa la versión 2.3.2 debido a que es la última establecida, pero podría emplearse otra versión si se desea.

- También fue un problema la versión de la máquina desde la que se despliega la red, puesto que es un Mac con procesador Apple M1, pero se reconocía como si fuera un procesador Intel, con lo cual generaba ciertos errores.

Para ello, se tuvo que instalar Rosetta [8] y además, se empleó el siguiente comando para indicar que el procesador utilizado era de tipo AMD (Advanced Micro Devices) y no ARM (Advanced RISC Machine):

```
docker pull hyperledger/fabric-ca --platform=linux/amd64
```

- Otro problema que se encontró al principio fue que al cerrar la red mediante el comando `./network.sh down`, había contenedores que no se cerraban, quedándose abiertos, y generaban conflictos después al levantar nuevamente la red. Por lo que se emplearon los siguientes comandos para forzar que estos se cerraran:

```
docker stop $(docker ps -a -q)
```

```
docker rm $(docker ps -a -q)
```

- Con el despliegue de la API, se encontraron algunos problemas más.

Primero, aparecía un error en la web debido a que la dirección es `file:///Users...` en vez de `http` o `https`. Por lo que, lo primero que se hizo fue alojar la web en un dominio, después se pensó en algo más sencillo y se instaló una extensión web de Chrome



llamada *"Web Server for Chrome"*. Pero lo que se acabó haciendo finalmente fue, mediante la utilización de express, desplegar la API en un puerto (3000 en este caso).

7 ANEXO III. AÑADIR PEERS Y ORGS

Para **añadir una organización** a la red de BlockChain se pueden seguir los siguientes pasos:

- En el directorio `fabric-samples/test-network/configtx/configtx.yaml` hay que duplicar la sección de `OrgX`, cambiando el nombre de la organización duplicada por el nombre de la nueva. Además, se debe añadir la nueva organización en la parte inferior del fichero en la sección de “Organizations”.
- Crear el fichero `fabric-samples/test-network/organizations/cryptogen/crypto-config-orgX.yaml`. Se puede duplicar alguno de los ficheros ya creados en este directorio y cambiar el nombre de la organización duplicada por el de la nueva.
- Crear un nodo para la nueva organización, ya que una organización tiene que estar formada por al menos un nodo (más abajo se detalla cómo crear un nodo).
- En el fichero `fabric-samples/test-network/organizations/ccp-generate.sh`, crear una sección con la nueva organización y sus puertos correspondientes.
- En el fichero `fabric-samples/test-network/network.sh`, añadir en la función `createOrgs()` lo correspondiente con la nueva organización (se puede ver cómo se hace con las organizaciones ya definidas, duplicarlo y poner el nombre de la organización correspondiente) y añadir `createOrgX` en las líneas posteriores de código.
- Clonar la carpeta `fabric-samples/test-network/organizations/fabric-ca/orgX` y modificar el archivo del interior introduciendo el nombre de la nueva organización.
- Modificar el fichero que se encuentra en la dirección `fabric-samples/test-network/organizations/fabric-ca/registerEnroll.sh`, creando la función `createOrgX()` e introduciendo el nombre de la organización que va a ser creada.

Cuando se habla de `OrgX`, se hace referencia al nombre que se desee dar a la organización, podría ser `Org1`, `Org2`, `Org3` etc. Lo mismo sucede más adelante cuando se escribe `peerX`.

Para **añadir un peer** dentro de una organización se pueden seguir los pasos que se muestran a continuación:

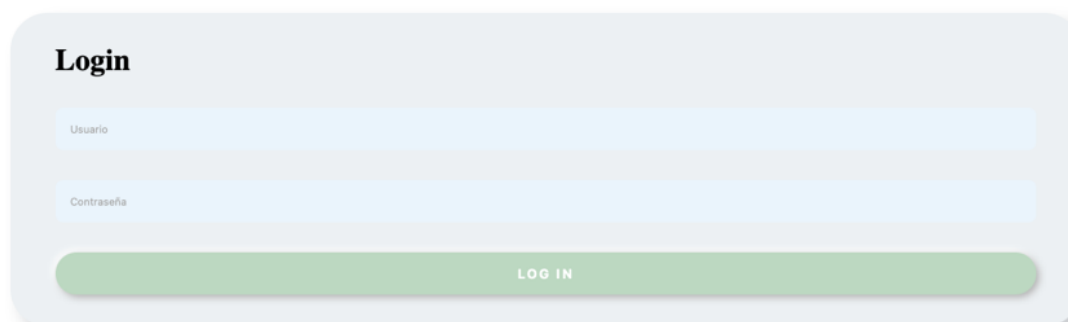
- Modificar del fichero `fabric-samples/test-network/organizations/cryptogen/crypto-config-orgX.yaml` la primera variable `Count` por el número de peers que se quieran crear en dicha organización. De esta forma, se crea el fichero `fabric-`

`samples/test-network/organizations/peerOrganizations/org1.example.com
/peers/peerX.orgX.example.com.`

- Para que se pueda añadir el peer al iniciar la red, hay que modificar el fichero `fabric-samples/test-network/docker/docker-compose-test-net.yaml` añadiendo los correspondientes peers. Se puede duplicar la información de un peer, cambiando el nombre del peer duplicado por el del nuevo, y se cambia el puerto de esta información. También es necesario añadir el puerto correspondiente en la parte inferior y superior del fichero.

8 ANEXO IV. MANUAL DE USUARIO

Lo primero con lo que se encuentra un usuario al iniciar sesión, como se puede ver en la Figura 11, es con la sección de login, donde se pide introducir el nombre de usuario y la contraseña.

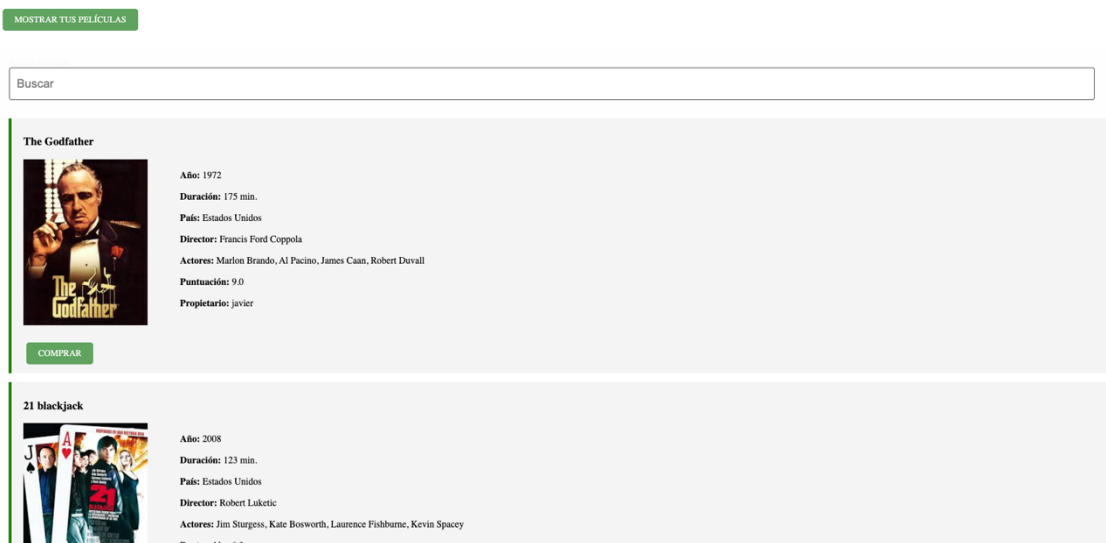


The login form is titled "Login" and is contained within a light gray rounded rectangle. It features two input fields: "Usuario" (User) and "Contraseña" (Password), both with light blue borders. Below these fields is a green button with the text "LOG IN" in white capital letters.

Figura 11: Sección de login

Una vez se haya iniciado sesión, se muestran las películas que el usuario dispone para comprar, como muestra la Figura 12, mostrando las películas donde su propietario no coincida con el nombre de usuario que se ha introducido previamente. Si se pulsa el botón de comprar una película, esa película pasará a ser de la propiedad de el usuario que la ha comprado, y en el campo de el propietario aparecerá el nombre de dicho usuario. Como estamos en la sección de comprar películas, la película que ha sido comprada ya no se mostrará en esta página. También se puede realizar una búsqueda de las películas según su título en la opción que hay arriba de buscar, de tal manera que se mostrarán las películas cuyo título coincida con el texto introducido.

PELÍCULAS DISPONIBLES PARA COMPRAR



The interface for purchasing movies includes a green button labeled "MOSTRAR TUS PELÍCULAS" at the top. Below it is a search bar with the placeholder text "Buscar". The main content area displays two movie cards. The first card is for "The Godfather", showing its poster, release year (1972), duration (175 min), country (Estados Unidos), director (Francis Ford Coppola), cast (Marlon Brando, Al Pacino, James Caan, Robert Duvall), rating (9.0), and owner (javier). A green "COMPRAR" button is at the bottom of the card. The second card is for "21 blackjack", showing its poster, release year (2008), duration (123 min), country (Estados Unidos), director (Robert Luketic), cast (Jim Sturgess, Kate Bosworth, Laurence Fishburne, Kevin Spacey), and rating (6.3).

Figura 12: Sección de películas para comprar

Si se pulsa el botón de arriba a la izquierda MOSTRAR TUS PELÍCULAS, se avanzará a la sección donde se muestran las películas que posee el usuario.

TUS PELÍCULAS

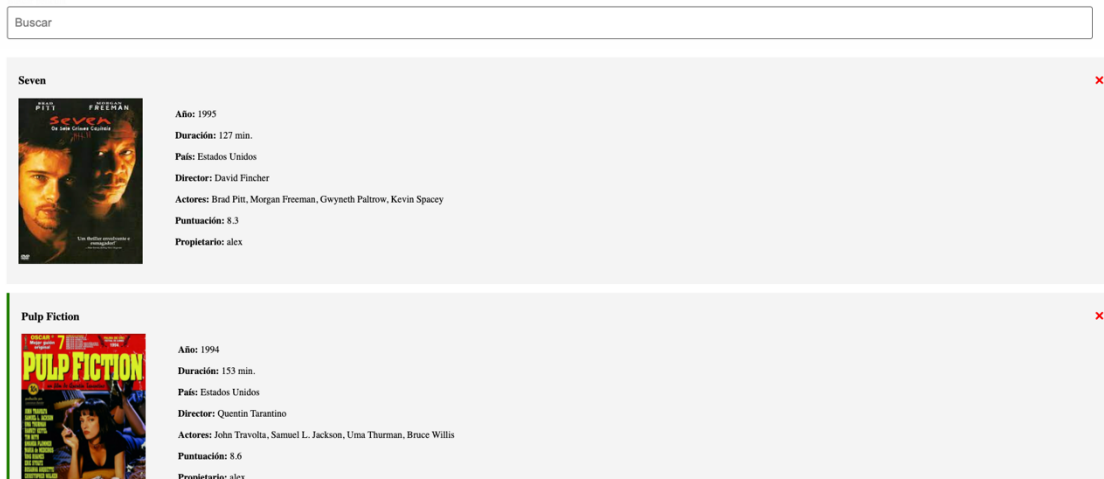
[MOSTRAR PELÍCULAS DISPONIBLES PARA COMPRAR](#)[AÑADIR](#)

Figura 13: Sección de tus películas

En la sección donde se muestran las películas del usuario (Figura 13) se pueden hacer diferentes acciones. Como sucedía en la sección de películas para comprar, se puede realizar una búsqueda según el título de las películas en esta sección también, funcionando de la misma manera que en la sección anterior.

Si se presiona el icono de la cruz roja que se sitúa en la parte superior derecha de una película, la película es eliminada directamente y no aparecerá en la sección de las películas del usuario ni en la opción de comprar de otros usuarios. En esta sección, las películas se pueden poner en venta o quitarlas de la venta, ambas acciones se realizarán haciendo doble clic sobre la película que se desee, de tal forma que, si la venta de la película está activada, aparecerá en la sección de películas para comprar por otros usuarios y si esta opción está desactivada no les aparecerá, y si la opción de venta está activada se mostrará una raya verde en la parte izquierda de la película. Como puede verse en el ejemplo de la Figura 13, la primera película tiene la opción de venta desactivada y la segunda activada.

Si se presiona el botón AÑADIR que está en la parte superior de la sección a la derecha se procederá a añadir una nueva película y aparecerán una serie de campos que el usuario debe introducir para crear la película. (Figura 14)

TUS PELÍCULAS

MOSTRAR PELÍCULAS DISPONIBLES PARA COMPRAR

CERRAR

Título
Año
Duración
País
Director
Actores
Puntuación
Foto

GUARDAR PELÍCULA

Buscar

Seven



Año: 1995

Duración: 127 min.

País: Estados Unidos

×

Figura 14: Sección añadir película

Después de añadir los campos de la nueva película, si se pulsa el botón que aparecerá en la parte inferior de la sección, GUARDAR PELÍCULA, se creará una película nueva con los campos introducidos donde el propietario será el usuario que la ha creado y aparecerá en las secciones correspondientes. El único campo obligatorio es el título y la película tendrá por defecto la opción de venta activada. Si se pulsa el botón rojo de la parte superior a la derecha, CERRAR, desaparecerán los campos para crear la nueva película y se mostrará nuevamente lo mostrado en la Figura 13. Si se pulsa el botón superior de la izquierda MOSTRAR PELÍCULAS PARA COMPRAR, se avanzará a la sección de la Figura 12, donde se muestran las películas que un usuario tiene disponibles para comprar.

9 ANEXO V. CONTROL DE LA TRAZABILIDAD

Uno de los principales objetivos del proyecto es poder llevar un seguimiento de la trazabilidad de un producto y registrar todas las fases por las que ha transcurrido dicho producto. Para poder demostrarlo con un ejemplo, se ha realizado una compra de una película que posee el identificador “item9”, donde esta película, que pertenece a alex, es comprada por Javier. En las imágenes mostradas se puede ver cómo esta acción queda registrada, donde se muestra que la compra es efectuada (Figura 17), que el ítem en una primera instancia pertenecía a alex (Figura 15), y que después de que se realice la compra, pertenece a Javier (Figura 16).

De esta manera se puede consultar los procesos por los que ha pasado un producto y los diferentes cambios que ha tenido.

```
1,
{
  "id": "item9",
  "Idint": 9,
  "Titulo": "21 blackjack",
  "Anyo": "2008",
  "Duracion": "",
  "Pais": "Estados Unidos",
  "Director": "Robert Luketic",
  "Actores": "Jim Sturgess, Kate Bosworth, Laurence Fishburne, Kevin Spacey",
  "Puntuacion": "6.3",
  "Foto": "https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcSDwFjgeVKLg4G",
  "Reminder": true,
  "Propietario": "alex"
}
```

Figura 15: Ítem9 con alex como propietario

```
{
  "id": "item9",
  "Idint": 9,
  "Titulo": "21 blackjack",
  "Anyo": "2008",
  "Duracion": "",
  "Pais": "Estados Unidos",
  "Director": "Robert Luketic",
  "Actores": "Jim Sturgess, Kate Bosworth, Laurence Fishburne, Kevin Spacey",
  "Puntuacion": "6.3",
  "Foto": "https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcSDwFjgeVKLg4G",
  "Reminder": true,
  "Propietario": "javier"
}
```

Figura 16: Ítem9 después de que lo compre Javier

```
--> EMPIEZA LA CREACIÓN DEL ITEM item23
--> ACABA LA CREACIÓN DEL ITEM item23
--> EMPIEZA LA CREACIÓN DEL ITEM item21
--> ACABA LA CREACIÓN DEL ITEM item21
--> EMPIEZA LA CREACIÓN DEL ITEM item20
--> ACABA LA CREACIÓN DEL ITEM item20
--> EMPIEZA LA CREACIÓN DEL ITEM item17
--> ACABA LA CREACIÓN DEL ITEM item17
--> EMPIEZA LA CREACIÓN DEL ITEM item0
--> ACABA LA CREACIÓN DEL ITEM item0
--> EMPIEZA LA CREACIÓN DEL ITEM item19
--> ACABA LA CREACIÓN DEL ITEM item19
Loaded the network configuration located at /Users/omist/go/src/github.com/
Built a CA Client named ca-org1
Built a file system wallet at /Users/omist/go/src/github.com/omist/fabric-
An identity for the admin user already exists in the wallet
An identity for the user appUser already exists in the wallet
--> EMPIEZA LA COMPRA DEL ITEM item9
--> ACABA LA COMPRA DEL ITEM item9
```

Figura 17: log donde se puede ver la compra del ítem9