

DApp de compraventa y alquiler de inmuebles

Héctor Montesinos Parra

Escola Politècnica Superior d'Enginyeria de Vilanova i la Geltrú

Grado en Ingeniería Informática

Resumen

Históricamente, el sector inmobiliario se ha visto afectado por la complejidad de sus procesos, sumado a una falta de transparencia, debido a un alto número de documentos e intermediarios participantes en dichos procesos.

Actualmente, la rompedora tecnología Blockchain se abre camino en diferentes sectores, debido a ciertos factores que aporta, cómo transparencia, inmutabilidad y trazabilidad.

En el presente proyecto se pretende crear una plataforma web descentralizada, lo que se conoce como una DApp (Decentralized Application) en un entorno blockchain, la cual permita agilizar procesos burocráticos, eliminando intermediarios, tiempos de espera y costes de gestión asociados a la compraventa o alquiler, además de acercar a los usuarios a las aplicaciones descentralizadas, brindándoles una buena experiencia de usuario.

Para llevar a cabo esta prueba de concepto, se realizará una investigación dentro del campo de la tecnología Blockchain, estudiando todo su potencial y características, dónde se utilizará un stack de herramientas y tecnologías que actualmente están a la vanguardia, detallando cada una de ellas, para posteriormente, llevar a cabo la implementación de la DApp.

1. Introducción

El presente proyecto consiste en el desarrollo de una plataforma web dedicada a la compraventa y alquiler de inmuebles, de manera totalmente descentralizada, gracias a la tecnología Blockchain, creando así lo que actualmente se conoce como una DApp (Decentralized Application).

Esta prueba de concepto, pretende resolver problemas existentes en el sector inmobiliario para adquirir o alquilar inmuebles, entre los que se pueden destacar la falta de verificación de la propiedad, lentitud en los procesos burocráticos con un gran número de intermediarios participando en dichos procesos, pagos de altas cantidades en costes notariales, alta probabilidad de existencia de errores humanos o la posibilidad de manipulación de transacciones.

Además de los puntos mencionados en el párrafo anterior, las plataformas dedicadas a la búsqueda de inmuebles, están faltas de procesos estandarizados, pudiendo contener datos inexactos, incompletos y desactualizados, respecto a otras plataformas del mismo sector.

2. Contexto

Una blockchain, o cadena de bloques pública, es comparable a un libro de contabilidad público, el cual almacena un historial de transacciones entre partes, en conjuntos de datos llamados bloques.

Una cadena de bloques es una red peer-to-peer (P2P), dónde los usuarios que participan se denominan nodos, los cuales todos actúan como cliente y servidor, ya que no hay ningún elemento central que actúe como servidor.

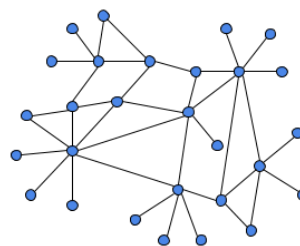


Figura 1. Red descentralizada

Las cadenas de bloques siguen algunos principios como inmutabilidad, transparencia, integridad, veracidad, confiabilidad o anonimato.

Volviendo a la analogía del libro de contabilidad público, todas las transacciones ocurridas dentro de una cadena de bloques, son públicas, y cualquiera puede acceder a ellas y consultar su información.

3. Objetivos

El objetivo principal del presente proyecto es crear una plataforma web descentralizada para el sector inmobiliario, que corra en una cadena de bloques pública, dando así el poder de las gestiones a los usuarios involucrados, los cuales pueden ser dos completos desconocidos y a la vez, realizar transacciones seguras entre ellos, al estar auditadas por los llamados contratos inteligentes.

Para lograr el objetivo de crear la plataforma web descentralizada, de aquí en adelante “DApp”, será necesario cumplir los siguientes puntos:

- **Desarrollo de contratos inteligentes:** Son programas autónomos que definen un acuerdo público. Están programados para realizar tareas autónomas, una vez estén desplegados en la cadena de bloques, por lo tanto, aseguran los acuerdos registrados en él.
- **Despliegue de los contratos en la cadena de bloques:** Una vez programados los contratos, serán desplegados en una cadena de bloques pública, para que cualquier

usuario pueda acceder a ellos, y que, posteriormente, las transacciones ocurridas en dicho contrato, sean inmutables y públicas para que cualquier las pueda consultar. Para llegar a este fin se deberá realizar un estudio para escoger la cadena de bloques más adecuada para su despliegue.

- **Desarrollo de un entorno gráfico:** Para que los usuarios puedan interactuar con la plataforma se quiere implementar un entorno gráfico amigable, que proporcione una buena experiencia de usuario y de fácil usabilidad.

La creación de la plataforma, quiere abarcar la venta y alquiler de propiedades, así como el alquiler de propiedades de manera tokenizada.

La tokenización es el proceso por el cual se emite un token, o activo digital, para representar un activo real. En este caso, cada token hace referencia a una fracción de un inmueble a alquilar, dónde cada token será igual a una habitación, no pudiéndose alquilar de manera tokenizada, una propiedad con menos de dos habitaciones.

A los usuarios que utilicen la plataforma se les quiere brindar las siguientes opciones:

- **Registro:** Pese a que, como se ha comentado, en la cadena de bloques es posible realizar transacciones seguras de manera anónima, en el caso de uso principal de este proyecto, es necesario asegurar los datos de las partes implicadas, por lo que, los usuarios deberán darse de alta en la plataforma, introduciendo algunos datos personales. Dicha información deberá validar que, en caso de querer publicar una propiedad, el usuario que la publique es el propietario de la misma.
- **Login y acceso:** Los usuarios deberán acceder a la plataforma para poder publicar, adquirir o alquilar propiedades, dónde se vincularán sus datos personales a una cuenta blockchain que los identificará dentro de la plataforma, y, en consecuencia, dentro de la cadena de bloques.
- **Publicar propiedades:** Los usuarios logueados podrán publicar propiedades, siempre que se verifique que son los titulares.
- **Adquirir o alquilar propiedades:** Los usuarios logueados podrán consultar todos los datos de las propiedades, obteniendo datos de contacto del propietario, por si quisieran agendar una visita, y como no, realizar la transacción de compra / alquiler desde la misma plataforma.
- **Consultar datos personales:** Todo usuario logueado podrá consultar su información personal, además de ver en todo momento qué propiedades y tokens tiene en su poder, desde el mismo apartado.

4. Organización

En cuanto a organización, el presente proyecto tuvo como fecha de matriculación el 22 de septiembre de 2021 y como fecha de entrega el 31 de enero de 2022.

Se realizó una estimación inicial de horas, en base a la carga de trabajo estimada, de aproximadamente 450 horas, dónde se incluían todas las tareas referentes al proyecto.

Se ha seguido una organización con sprints semanales, haciendo uso de metodologías ágiles, pese a que, al ser un proyecto individual, éstas han diferido ligeramente a las clásicas. Se ha hecho uso de un tablero KanBan, mediante el entorno web Trello [1], que ha facilitado la perspectiva de tareas pendientes, en proceso y completadas.

La planificación ha sido englobada en cuatro bloques principales:

- **Estudio y gestión:** Incluye todo el trabajo transparente que hay detrás del proyecto: Reuniones con la directora, seguimiento de las metodologías ágiles y del KanBan, estudio de las tecnologías a utilizar y aprendizaje de las mismas.
- **Especificación y diseño:** Planteamiento de las funcionalidades de las que dispondrá el proyecto, detallando los actores implicados en cada caso de uso y una visión general del proyecto.
- **Documentación:** Hace referencia a todo el material redactado que se aporta en la entrega del presente proyecto.
- **Implementación:** Desarrollo del código de la plataforma, se realiza un seguimiento mediante un repositorio de GitHub [2], el cuál permite llevar un control de las versiones gracias a Git [3].

Tras la planificación, se ha realizado un diagrama de Gantt inicial, y al final del proyecto se realizará otro para observar las diferencias para aprender de cara a futuros proyectos.

5. Especificación

La arquitectura del proyecto completo, gira entorno a la cadena de bloques. En este caso, no estamos ante una plataforma web tradicional alojada en un servidor, con una base de datos centralizada, por tanto, los datos quedan almacenados en la cadena de bloques, dentro de los contratos inteligentes desplegados.

El usuario interactúa con la plataforma mediante un navegador web, dónde puede acceder al entorno gráfico de la DApp. Posteriormente, debe disponer de una cuenta blockchain, la cual identifica al usuario en la DApp mediante una dirección de clave pública, que corresponde a un hash de 20 bytes.

Para enlazar dicha cuenta con la plataforma, el usuario debe conectar una billetera blockchain, lo cual es posible mediante ciertos plugins o extensiones de navegador. La billetera en cuestión permite encapsular cuentas blockchain en su interior. Cuando el usuario conecta una cuenta a la plataforma, ya puede realizar transacciones en la cadena de bloques, puesto que ya está identificado y ya forma parte de la red como usuario.

Para almacenar cierto contenido o información extra, como imágenes o documentos, externa a los contratos inteligentes, se hace uso de IPFS [4], una red de almacenamiento distribuida, P2P, la cual permite subir contenido y que quede almacenado de manera distribuida entre nodos. La información almacenada es direccionable por contenido, lo que significa que es accesible mediante un hash identificador llamado CID (Content ID), por lo que siempre está disponible.

La información de la cadena es totalmente interna, esta no tiene acceso al mundo exterior, creando lo que llamamos

“entorno on-chain”. En ocasiones es necesario acceder a información del exterior, o “entorno off-chain”, para lo cual se hace uso de los llamados oráculos.

Un oráculo es un puente entre la cadena de bloques y el mundo real, el cual actúa como servicio interno de la cadena, con el objetivo de trasladar información de una parte a otra.

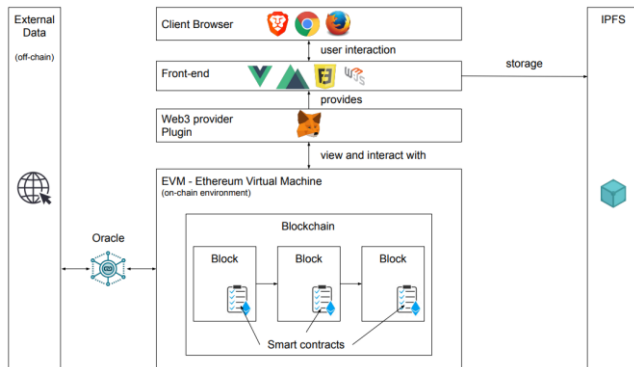


Figura 2. Arquitectura de la DApp

6. Tecnologías

Después de introducir las tecnologías utilizadas en el diagrama de arquitectura (figura 3), a continuación, se van a comentar en detalle:

- **Blockchain:** Pieza fundamental del proyecto, es la base de datos distribuida dónde se despliega la plataforma. Sin embargo, existen diferentes tipos de cadenas de bloques, las cuales hay que conocer para saber cuál escoger:

- **Públicas:** Cualquier persona tiene acceso a ella, ya sea como participante, mediante un nodo, o simplemente como usuario. Para participar, los usuarios habitualmente han de descargarse un software, instalarlo y conectarse a la red. No hay ninguna entidad central detrás y toda la información es pública, desde el código del software, hasta las transacciones ejecutadas en ella. Es económicamente autosuficiente, puesto que suelen contar con su propio “token” o “criptodivisa”, sobre las cuales se realizan las transacciones y los pagos de comisiones de las mismas.

Ejemplo: Bitcoin [5]

- **Privadas o permissionadas:** Están controladas por una entidad central, que da permisos a usuarios y participantes para acceder a la red.

Ejemplo: Hyperledger [6]

- **Federadas o híbridas:** Están controladas por varias entidades centrales, que suelen ser organizaciones o entidades gubernamentales. Similarmente a las privadas, dan autorización a los miembros que pueden participar en ellas.

Ejemplo: Alastria [7]

Para este proyecto, no se va a depender de ninguna organización, con el objetivo de seguir los principios de la descentralización, y crear una plataforma lo más abierta posible, para dar el máximo control a los usuarios, por lo cual, se va a desarrollar sobre una cadena de bloques pública.

Las cadenas de bloques públicas están divididas en tres generaciones:

- **1ª Generación:** Establece un método de realizar transacciones P2P con divisas digitales, sin necesidad de un intermediario como una entidad bancaria, siguiendo un protocolo de consenso llamado Proof-of-Work (PoW).

En las redes que utilizan el protocolo PoW, a los participantes de la red se les llama “mineros”, puesto que, mediante supercomputadoras tratan de resolver cálculos matemáticos con el objetivo de encontrar un hash concreto, en base a la entrada de todos los datos encapsulados en un bloque. Como recompensa a esta labor con alto gasto computacional, si consiguen encontrar el hash objetivo, son recompensados con el token nativo de la red.

- **2ª Generación:** También utilizan el protocolo de consenso PoW, introduciendo los contratos inteligentes, gracias a contar con un sistema Turing complete [8], lo cual permite programar cualquier tipo de operación, y no sólo realizar transacciones de divisas sencillas. El hecho de poder implementar estructuras de datos permite crear aplicaciones que corran en la cadena de bloques. En esta segunda generación nace Ethereum [9], primera blockchain de segunda generación Turing complete.

- **3ª Generación:** Nacen para mejorar algunos problemas de la 2ª generación, como la escalabilidad y la velocidad de transacciones. Utilizan un protocolo de consenso distinto, llamado Proof-of-Stake (PoS), en el cual los participantes no actúan como mineros, sino que son incentivados por la cantidad de token nativo de la red del que son poseedores.

La blockchain escogida para implementar el proyecto es **Ethereum**, debido a qué, al ser una blockchain de 2ª generación permite el despliegue de aplicaciones en su interior, y que las cadenas de bloques de 3ª generación todavía están en pleno desarrollo, por lo que no cuentan todavía con las herramientas necesarias para facilitar el testing, la implementación y el posterior despliegue de aplicaciones.

- **IPFS:** El protocolo InterPlanetary File System, tiene un propósito similar al de blockchain, crear una red de computadoras o nodos, de alcance global, en este caso con el objetivo de permitir el almacenamiento de información de manera descentralizada, consiguiendo así una red P2P global. A diferencia del conocido HTTP, IPFS crea una red permanente y distribuida, utilizando un sistema de búsqueda por contenido, mediante un hash CID que identifica a cada archivo compartido en la red, el cual se almacena de manera fraccionada en uno o varios nodos, dependiendo de su tamaño. Para participar en la red

mediante un nodo, los usuarios deben descargar un cliente software, el cual los provee de un nodo local.

Ethereum

La cadena de bloques sobre la cual se despliega el proyecto es Ethereum, una blockchain pública que permite el desarrollo de aplicaciones, pudiendo desplegar contratos inteligentes en ella y que hace uso del token Ether (símbolo ETH), como medio de pago dentro de la red.

Las transacciones en una blockchain se realizan entre cuentas, las cuales se guardan en billeteras software.

En Ethereum existen dos tipos de cuentas blockchain:

- **Cuentas de usuario:** Identifican a un usuario, como su nombre indica, están formadas por un par criptográfico de clave pública y clave privada. Se pueden generar a través de librerías mediante entradas de datos, las cuales genera una clave privada de 64 caracteres hexadecimales. Posteriormente, en base a dicha clave privada, se aplica un algoritmo criptográfico asimétrico de curva elíptica (ECC), el cual se basa en funciones matemáticas simples de calcular en una dirección y muy difíciles de revertir. Del resultado obtenido, se obtiene una dirección de clave pública mediante los últimos 20 bytes de la clave pública y añadiendo 0x al inicio.
- **Cuentas de contrato:** Identifican a un contrato inteligente, en este caso, poseen una dirección hexadecimal de 42 caracteres, la cual se le asigna al contrato una vez es desplegado en la red, en base a los datos de entrada de la dirección de clave pública del creador del contrato y del número de transacciones enviadas desde esa misma dirección.

7. Implementación

La implementación del proyecto se divide en tres bloques:

- **Blockchain:** Red dónde se despliega la DApp.
- **Cliente Web:** Entorno mediante el cual los usuarios interactúan con la plataforma.
- **Intermediarios:** Herramientas y aplicaciones que permiten interconectar ambas partes o que sirven de complemento para alguna de ellas.

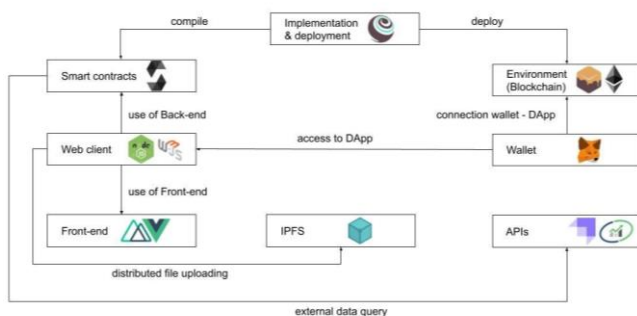


Figura 3. Arquitectura por tecnologías

Blockchain

Se han creados dos contratos inteligentes:

- **Auth:** Permite validar a usuarios en la plataforma, dándoles opción de registrarse, loguearse y desloguearse.
- **Properties:** Permite a los usuarios logueados publicar propiedades, dándoles opción de vender, alquilar propiedades. En caso de ser titular de una propiedad que no ha sido vendida o alquilada, el usuario puede eliminarla de la plataforma.

Los contratos inteligentes han sido desarrollados mediante el lenguaje de programación Solidity [10], el cual fue creado para dicho propósito.

El entorno sobre el que se han desplegado los contratos inteligentes de la DApp es Truffle [11], un framework que permite configurar una cadena de bloques de Ethereum en un entorno de desarrollo local, lo cual es vital, ya que un despliegue en la red principal de Ethereum no podría ser revertido, y la aplicación quedaría ya desplegada para siempre. Truffle provee por cada red configurada de 10 cuentas de usuario con 100 Ethers de balance en cada una, para poder hacer tests.

Una vez Truffle está configurado, se ha hecho uso de Ganache [12], un entorno gráfico dónde es posible importar la red de desarrollo configurada con Truffle, y así poder ver de forma cómoda, información interesante como los bloques minados, la información que se incluye en las transacciones o en los contratos inteligentes.

La plataforma en ocasiones deberá consultar datos del entorno off-chain, o mundo exterior, para lo cual se hará uso de Chainlink [13], un proyecto de oráculo descentralizado que corre en la blockchain de Ethereum, que permite a los contratos inteligentes consultar y obtener datos del entorno off-chain. Los datos que se quieren obtener del mundo exterior son los siguientes:

- **Precio del Ether:** Los pagos dentro de la blockchain de Ethereum se realizan mediante la criptomoneda Ether, el token nativo de la red. Por tanto, los pagos en la plataforma se hacen también con Ether. El Ether es una divisa, cuyo precio fluctúa constantemente contra las divisas tradicionales como el euro o el dólar, por lo que es necesario saber en todo momento cual es su precio en euros, para poder mantener el valor en Ethers, de las propiedades publicadas en la DApp.
- **Verificación del registro de la propiedad:** Para aplicar una seguridad lo más alta posible, sólo se permite publicar propiedades a aquellos usuarios hacia los cuales se haya verificado que son los titulares de las mismas. Para ello, se consulta el registro de la propiedad vigente. El registro de la propiedad es una institución del estado español, dónde se inscriben los inmuebles para saber quien es el propietario, y sus derechos y cargas sobre una propiedad. Por tanto, en el momento en que un usuario vaya a publicar una propiedad, se comprobará que el par <Referencia catastral, Datos usuario> introducido desde la plataforma, coincida con los datos que hay en el registro de la propiedad.

Otro propósito de la plataforma, es el de generar un NFT [14] (Non-fungible token), para todos aquellos usuarios que alquilen una propiedad de manera tokenizada, para así,

además de quedar reflejado en el contrato inteligente que han adquirido un token, que reciban un objeto digital que simule la fracción de la propiedad que están alquilado.

Un NFT, hace referencia a un estándar de tokens de la red, llamado ERC-721. A diferencia del ETH, el cuál corresponde al estándar ERC-20. La diferencia entre ellos, reside en que, los estándares ERC-20, cómo el Ether, son divisibles o fraccionables en partes más pequeñas, en el caso del Ether, se pueden dividir hasta 10^{18} veces, siendo el Wei su unidad mínima. Por tanto, un usuario puede enviarle a otro, por ejemplo, 0.001 ETH, en cambio, un token ERC-721, es indivisible y limitado, por tanto, su valor intrínseco reside en su rareza, debido a qué su generación o acuñamiento, se realiza manualmente mediante un contrato inteligente.

Metamask

Para poder interactuar con la DApp desde un entorno gráfico y poder realizar transacciones, son necesarias cuentas blockchain, las cuales viven en billeteras software.

La billetera utilizada en este proyecto es Metamask [15], un software que posee extensión de navegador y aplicación multiplataforma para smartphones y que permite configurar una billetera en la red de Ethereum, además de permitir configurar otras redes.

En este caso, se ha configurado la red local de desarrollo configurada mediante Truffle en Metamask, lo cual permite utilizar la billetera con 10 cuentas que nos provee Truffle en la DApp.

Front-end web

Para brindar a los usuarios una buena experiencia de usuario en la plataforma, se ha desarrollado una interfaz web, o front-end, puesto que Truffle provee de una consola de comandos, pero ese no es un entorno fácil ni idóneo para que los usuarios interactúen con la DApp.

Las tecnologías utilizadas para su desarrollo son las siguientes:

- **Vue.js [16]:** Framework escrito en JavaScript, centrado en front-end, que trabaja con componentes, dónde cada componente encapsula código de HTML, CSS y JavaScript, en un único archivo, lo cual permite proyectos fáciles de escalar, además de poder reemplazar componentes de manera sencilla.
- **Nuxt.js [17]:** Framework escrito en JavaScript, basado en Vue.js, permite crear proyectos con una estructura de directorios predefinida, además de mejorar algunas configuraciones de Vue.js como el enrutamiento entre componentes.
- **Web3.js [18]:** Librería de JavaScript que permite conectar el front-end con los contratos inteligentes desplegados en la blockchain.
- **Node.js [19]:** Entorno open-source que ejecuta código JavaScript en el lado del servidor, sin utilizar el navegador. Aloja el código que interconecta el front-end con la blockchain, haciendo uso de las librerías Web3.js y TruffleContract [20] para lograr este propósito.

IPFS

Desde el front-end, se utilizará el protocolo IPFS para almacenar ciertos ficheros, cómo las imágenes de las propiedades que los usuarios suban a la hora de publicar las mismas, cómo las propias imágenes de perfil de los usuarios, además de alojar los contratos de compraventa o alquiler, que generará la DApp automáticamente en el momento de venta o alquiler de una propiedad.

Datos externos

Cómo se ha detallado en el apartado “Ethereum”, se quiere realizar una consulta al registro de la propiedad para verificar que los usuarios son titulares de las propiedades, en el momento de su publicación. Sin embargo, y tras una investigación, se ha comprobado que la consulta de esta información, es un servicio privado del Estado, y además de pago, por tanto, no tenemos la posibilidad de obtener tal información. Por este motivo, para esta prueba de concepto, se ha construido una API, mediante Strapi [21], un framework de Node.js). Esta API simula el registro de la propiedad, ya que cuenta con una base de datos con una colección llamada “Property”, la cual almacena datos de propiedades y sus titulares, por lo que, la DApp puede consumir los datos almacenados en ella.

8. Navegación

Las pantallas de navegación de las que disponen los usuarios desde el entorno gráfico son las siguientes:

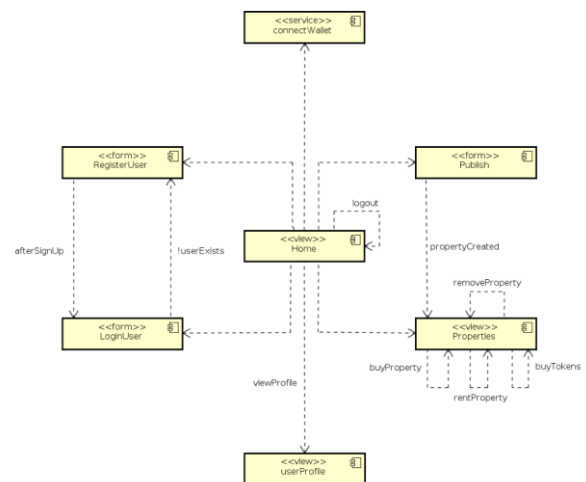


Figura 5. Diagrama de pantallas de navegación

En la “figura 5”, también se aprecia la relación que hay entre ellas, y desde cual se puede enlazar a otra.

Brevemente, se va a describir a cada una de ellas:

Home

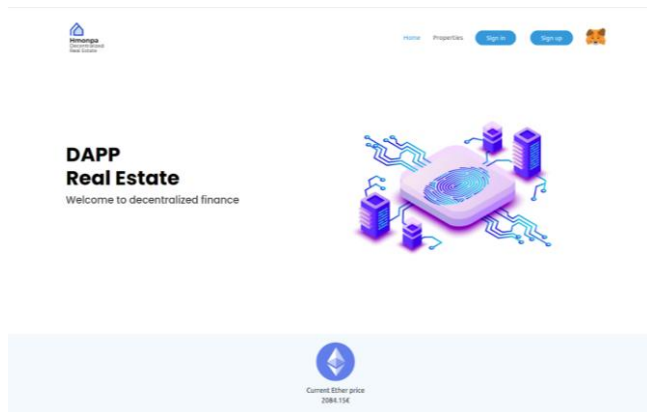


Figura 6. Pantalla de inicio

Es la pantalla inicial, muestra el precio del Ether, el cual se actualiza de manera reactiva cada segundo, permite a los usuarios conectar una billetera, consultar las propiedades publicadas, registrarse e iniciar sesión.

Sign up

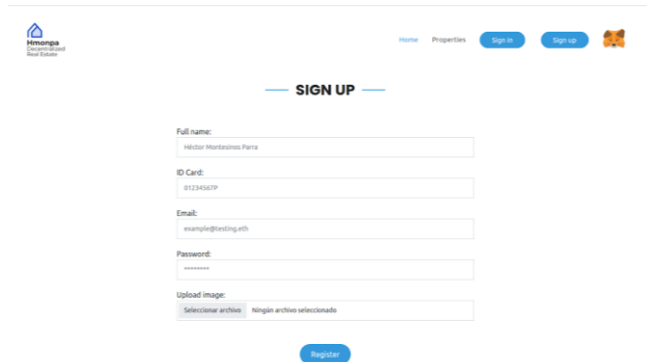


Figura 7. Pantalla de registro de usuarios

Permite a los usuarios registrarse en la plataforma, para ello, antes deben tener conectada una cuenta mediante la billetera de Metamask, ya que la clave pública es lo que les identifica dentro de la plataforma.

Sign in

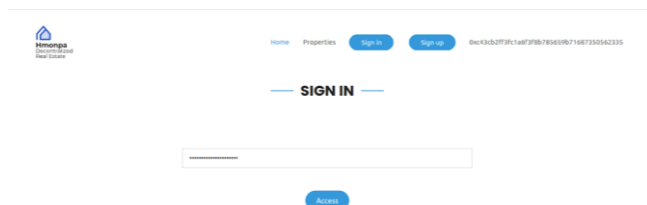


Figura 8. Pantalla de inicio de sesión

Una vez registrados, a los usuarios se les redirige a la pantalla de inicio de sesión, dónde en la parte superior derecha, pueden observar la clave pública que les identifica, por tanto, sólo deben introducir su contraseña.

Una vez inician sesión, aparece una opción en esa misma parte superior derecha, que permite a los usuarios consultar sus datos personales, además de las propiedades o tokens que poseen en la plataforma:

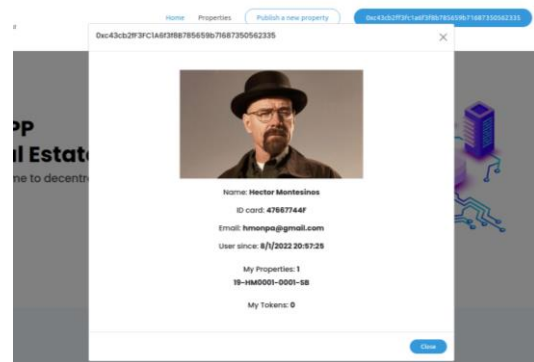


Figura 9. Modal detalles de usuario

Publish

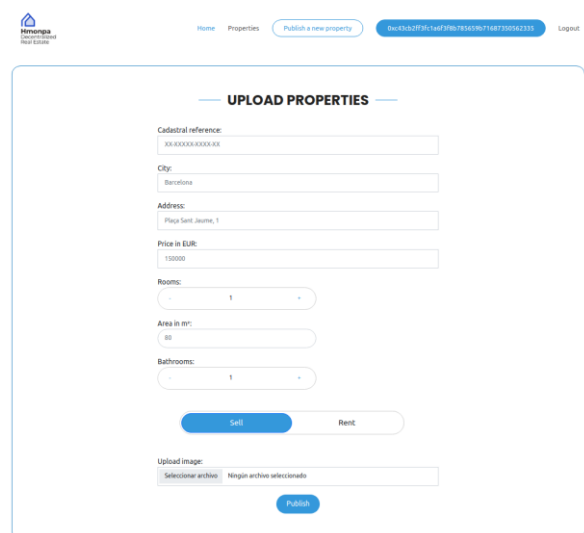


Figura 10. Pantalla de publicación de propiedades

Permite a los usuarios publicar propiedades en el objetivo de venderlas, alquilarlas o alquilarlas de manera tokenizada.

Properties

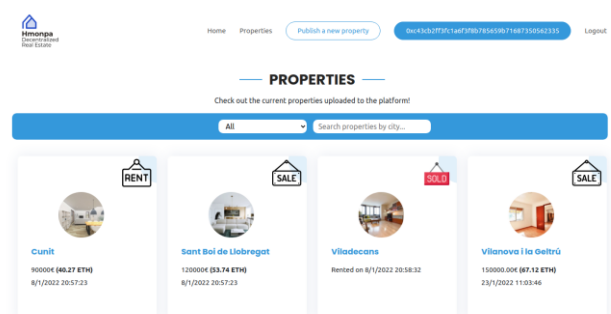


Figura 11. Pantalla de propiedades

Los usuarios pueden consultar las propiedades publicadas en la DApp, pudiendo filtrar por las publicadas por ellos mismos o por ciudad.

Al hacer clic sobre alguna de ellas, aparece un modal dónde pueden consultar los datos de la propiedad, los datos de contacto del propietario y la opción de adquirirla o alquilarla:

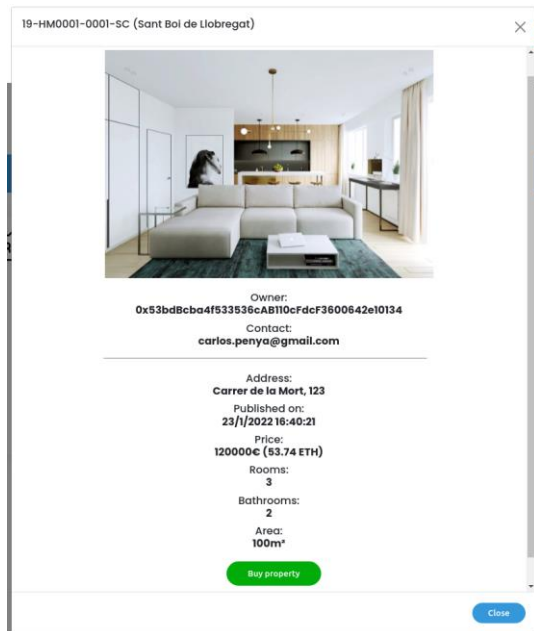


Figura 12. Modal de propiedad en venta

En el momento de realizar una compra o alquiler, una vez el pago se ha realizado al propietario, se genera el contrato de compraventa o alquiler, respectivamente, el cual se almacena en la red IPFS, y a los usuarios se les brinda la posibilidad de descargarlo en PDF, acceder al enlace del contrato en IPFS y almacenar el hash CID del mismo:



Figura 13. Modal con información del contrato generado



Figura 14. Visualización de contrato de compraventa en IPFS

9. Conclusiones

Cumplimiento de objetivos

El objetivo de este proyecto ha sido la exploración en el campo de la tecnología blockchain con tal de crear una prueba de concepto que pueda ayudar a mejorar el ámbito del sector inmobiliario en el futuro.

Durante el desarrollo del proyecto han surgido algunos contratiempos, cómo se podía esperar ya que es algo a tener en cuenta cuando se plantea un proyecto.

En algunos casos se ha tenido que cambiar el planteamiento inicial, pero llegando finalmente a obtener el mismo objetivo.

Personalmente considero que finalmente se cuenta con una plataforma web descentralizada que cumple con lo que se espera de ella, aportando a los usuarios que la utilicen una transparencia y veracidad en sus operaciones, al estar respaldadas por un smart contract y sin depender de un número desconocido de intermediarios, además de facilidad de uso, algo que no es lo habitual en este tipo de aplicaciones descentralizadas, ya que es un sector innovador que está comenzando, dónde la experiencia de usuario es todavía muy mejorable.

Conclusiones personales

Inicialmente el planteamiento del proyecto fue complicado, debido a que, el hecho de trabajar de manera individual y no en grupo, hace que el proyecto se pueda plantear de manera subóptima, y que se tarde más en corregir o encarrilar un posible error de planificación. Pese a ello, estoy satisfecho con mi trabajo, considero que he organizado mi trabajo diario de una manera óptima, siguiendo una estructura de sprints semanales, mediante un KanBan de seguimiento que me ha permitido ir cumpliendo objetivos a lo largo del proyecto y mejorar mi eficiencia a la hora de organizarme.

Desde el punto de vista tecnológico, he aprendido muchos conceptos de la tecnología blockchain, que al inicio del proyecto creía, o sospechaba, que funcionaban de una manera, que posteriormente he visto que estaba equivocado. También he aprendido a programar en Solidity, así como el uso de las herramientas expuestas en el proyecto, además de mejorar mi nivel de programación con JavaScript y conocer algunos de sus frameworks, dónde todo el código ha quedado almacenado en un repositorio de GitHub propio.

Futuras mejoras

El proyecto en su estado actual, cumple con los objetivos fijados en el inicio, y sirve como prueba de concepto de una plataforma que, con algunos cambios podría existir y ser utilizable por usuarios reales.

Sin embargo, un punto interesante a implementar, sería poder aportar liquidez a los usuarios, de manera que puedan adquirir o alquilar propiedades, mediante transacciones periódicas. Actualmente, el hecho que impide llevar a cabo este punto, es la falta de liquidez por parte de la plataforma.

Si la plataforma estuviese respaldada, teniendo fondos suficientes, en el momento de ejecutar transacciones entre

usuarios, la parte compradora, ejecutaría la compra o alquiler de un inmueble, y la plataforma abonará su totalidad al propietario. Posteriormente, la parte compradora podría ir abonando periódicamente, por ejemplo, mes a mes, la equivalencia del importe total a una dirección de clave pública dónde el propietario sea la plataforma.

En este contexto, seguiría existiendo una descentralización, debido a que las condiciones estarían especificadas en un smart contract, cómo ahora, pero se les daría soporte a los usuarios a la hora de realizar transacciones de grandes importes, cómo sabemos que son los de los inmuebles.

Otro contratiempo para desplegar la plataforma y que fuese utilizada de manera real, es el hecho de consultar el registro de la propiedad, puesto que se tratan de datos privados y, pese que la tecnología blockchain cada vez está recibiendo más adopción por parte de organizaciones y gobiernos, este punto difícilmente cambiará y dichos datos seguirán siendo privados, por lo que, habría que aplicar otro tipo de autenticación de datos en un futuro, cómo por ejemplo, la subida de documentos acreditativos de la propiedad a la plataforma, de manera que no sean visibles, pero que permitan a los usuarios publicar propiedades de manera segura y verificada.

10. Agradecimientos

A mi tutora Eva Marín, por aceptarme este tema para mi proyecto final de grado y por sus comentarios durante el seguimiento del proyecto.

A mi pareja, quién me ha visto trabajar, pegado al ordenador día y noche y siempre ha estado ahí para brindarme su apoyo incondicional.

A mi compañero y amigo Javier Delgado, por escucharme y darme ánimos, interesándose y dándome feedback del estado del proyecto en todo momento.

A los fundadores de mi empresa, Nectios, los cuales, además de mostrar su interés acerca del proyecto, han sido flexibles con mi horario durante estos meses, dándole prioridad a mi formación.

Por último, a mis amigos de toda la vida y a mi familia, por preocuparse por mí, durante el transcurso del proyecto.

Referencias

- [1] Trello: <https://trello.com/es>
- [2] GitHub: <https://github.com/hmonpa/TFG>
- [3] Git: <https://git-scm.com/about>
- [4] IPFS: <https://ipfs.io/#why>
- [5] Bitcoin: <https://bitcoin.org/bitcoin.pdf>
- [6] Hyperledger: <https://www.hyperledger.org/>
- [7] Alastria: <https://www.alastria.io/>
- [8] University of Virginia: https://www.cs.virginia.edu/~robins/Turing_Paper_1936.pdf
- [9] Ethereum: <https://ethereum.org/es/>
- [10] Solidity: <https://docs.soliditylang.org/en/v0.8.0/>
- [11] Truffle: <https://trufflesuite.com/truffle/>
- [12] Ganache: <https://trufflesuite.com/ganache/>
- [13] Chainlink: <https://chain.link/>
- [14] Ethereum: <https://ethereum.org/en/nft/>

- [15] Metamask: <https://metamask.io/>
- [16] Vue.js: <https://es.vuejs.org/v2/guide>
- [17] Nuxt.js: <https://v3.nuxtjs.org/concepts/introduction>
- [18] Web3.js: <https://web3js.readthedocs.io/en/v1.7.0/>
- [19] Node.js: <https://nodejs.org/es/about>
- [20] NPM - TruffleContract: <https://www.npmjs.com/package/truffle-contract>
- [21] Strapi: <https://strapi.io/>