Desenvolupament d'una DApp de crowdfunding sobre Ethereum

Ricard Alcaraz Mancebo

Resum– Aquest article presenta el disseny, desenvolupament i resultats d'una aplicació distribuïda, també anomenada DApp. En concret s'ha desenvolupat una DApp que permet crear campanyes de crowdfunding i a la vegada poder fer donacions a aquestes campanyes. Totes les transaccions que es realitzin seran gestionades per un Smart Contract sobre la xarxa d'Ethereum, que és una blockchain descentralitzada que permet la funcionalitat d'Smart Contracts. A més de l'Smart Contract s'ha desenvolupat una interfície web de forma que qualsevol usuari pugui interactuar amb aquest. Gràcies al fet que Ethereum és una blockchain descentralitzada permetrà que les donacions que es realitzin a aquestes campanyes es facin sense la necessitat de confiar en cap altra entitat que la mateixa infraestructura d'Ethereum. El mateix Smart Contract serà públic, estarà desplegat a la blockchain i se'n podra veure el seu funcionament intern.

Paraules clau– Cadena de blocs, Blockchain, Smart Contract, Ethereum, aplicació distribuïda, DApp, Crowdfunding

Abstract– This paper presents the design and results of a distributed application or DApp on the Ethereum network, this DApp will allow to create crowdfunding campaigns where every user will be able to donate if they want. An Smart Contract and a web interface has been developed, this way every user will be able to interact with this Smart Contract. Every transaction will be managed by this Smart Contract without needing any other entity and it also will be public, so everyone will be able to see it's functionality, this Smart Contract will be deployed in the Ethereum blockchain, which is a public decentralized blockchain that allows the Smart Contract functionality.

Keywords– Blockchain, Smart Contract, Ethereum, Crowdfunding, Distributed Application, DApp

1 Introducció

En l'actualitat la tecnologia blockchain és un concepte que causa molt interès tant en l'àmbit de la recerca com en l'àmbit empresarial, tot gràcies a la tecnologia que va presentar Satoshi Nakamoto amb el paper "Bitcoin: A Peer-to-Peer Electronic Cash System"[1]. Arran de la tecnologia Bitcoin, altres tecnologies basades en blockchain han anat sorgint, com és el cas d'Ethereum. Ethereum és considerada una blockchain de segona generació, sent Bitcoin de primera generació, aquesta a més de la possibilitat que ofereix Bitcoin de ser un sistema descentralitzat on poder realitzar transaccions monetàries entre usuaris, permet desenvolupar Smart Contracts [2]. Un Smart contract és en essència un programa que es desplegarà sobre la blockchain i que funcionarà com un contracte, en el qual les transaccions que passin per aquest contracte estaran subjectes a condicions, les quals estaran definides dintre de l'Smart Contract. Tot això sense la necessitat de cap altra entitat o actor que hi participi, tot aquest procés es realitzarà de manera descentralitzada i pública, tal com és la mateixa xarxa d'Ethereum. Ethereum utilitza el sistema Proof-of-Work, aquest és un sistema que tracta d'incentivar el correcte comportament dels usuaris a la xarxa i així fer-la més segura. Aquest és el sistema que s'utilitza per a minar els blocs a la xarxa donant rellevància a la capacitat de còmput del miner. Tot i que Ethereum vol canviar aquest sistema pel Proof-of-Stake en la seva version 2.0. En aquest sistema la seguretat s'aconsegueix a través d'escollir usuaris per a minar els blocs que siguin posseïdors de certa quantitat de monedes de la xarxa, en aquest cas Ethers, en aquest sistema no és tan rellevant la capacitat de còmput.

Els Smart Contracts obren moltes possibilitats per a la xarxa d'Ethereum com poden ser els NFT, els sistemes de

• E-mail de contacte: ricard.alcarazm@e-campus.uab.cat
• Menció realitzada: Tecnologies de la Informació
• Treball tutoritzat per: Jordi Herrera Joancomartí (DEIC)
• Curs 2020/21

DeFi o les DApps [3]. En concret aquest treball tracta del disseny, desenvolupament i resultats d'una DApp sobre la xarxa d'Ethereum. En aquest cas s'ha optat per fer una aplicació de crowdfunding. El motiu principal per l'elecció d'una aplicació de crowdfunding és els avantatges que ofereix que tot es realitzi de manera descentralitzada. D'aquesta forma s'elimina la necessitat de confiar en una entitat intermediària en la que confiem que tots els fons recollits per a la campanya s'enviïn al propietari de la campanya i no es desviïn, es perdin o l'intermediari se'ls emporti. Llavors aquesta aplicació garantirà que totes les donacions arribaran al seu destinatari si la campanya recull els fons que necessita, tot de forma transparent i pública. L'objectiu del treball és poder fer una aplicació funcional que es pugui desplegar sobre la xarxa d'Ethereum i que sigui segura. En el cas de que sigui necessari realitzar alguna modificació de l'Smart Contract, aquesta s'haurà de desplegar de nou sobre la xarxa a costa d'un cost de desplegament i modificar l'aplicació perquè interactuï amb el nou Smart Contract. El document tractarà d'explicar el desenvolupament d'aquesta aplicació, la metodologia utilitzada, la planificació, els resultats i les possibles millores que es podrien introduir, a més d'unes conclusions del treball.

2 ESTAT DEL ART

Com s'ha comentat prèviament aquesta DApp es desplegarà en la xarxa d'Ethereum i aquesta permet la funcionalitat dels Smart Contracts. La intenció d'Ethereum és la de ser un protocol per a crear aplicacions descentralitzades, oferint eines per a facilitar el desenvolupament d'aquestes. Essencialment ofereix una blockchain que permet executar un llenguatge de programació Turing-complet [4]. Ethereum també proveeix de xarxes de proves anomenades testnets. En el cas d'aquest treball, l'aplicació no s'ha desplegat sobre la xarxa principal d'Ethereum, si no que s'ha desplegat sobre la xarxa tesnet Kovan [5].

2.1 Smart Contracts d'Ethereum

Per a Ethereum un Smart Contract és un programa que es desplega a la xarxa d'Ethereum i que s'executa sobre la blockchain. Els Smart contracts són un tipus de compte d'Ethereum [6], és a dir que tenen el seu propi saldo i a més poden enviar transaccions sobre la xarxa. Els usuaris de la xarxa podran interactuar amb el contracte, de forma que podran cridar a funcions de l'Smart Contract. Aquestes funcions estaran definides i es basaran en condicions que permetin a l'usuari realitzar una acció o una altra.

Qualsevol usuari de la xarxa pot ser capaç de crear el seu Smart Contract i de desplegar el contracte. Per a poder desplegar un contracte serà necessari que l'usuari pagui en Gas [7]. El Gas en essència és una taxa que l'usuari paga per cada operació que el Smart Contract pot fer. Aquest Gas està relacionat amb la mateixa moneda d'Ethereum, l'Ether. Aquestes possibilitats que ofereix Ethereum s'han aprofitat els usuaris per a realitzar les seves pròpies aplicacions distribuïdes, també anomenades DApps.

2.2 DApps

Una aplicació descentralitzada, o DApp, és una aplicació creada en una xarxa descentralitzada que combina un Smart Contract amb una interfície d'usuari. Tots els Smart Contracts que es despleguen sobre la xarxa d'Ethereum són públics, de forma que són accessibles i transparents. Això significa que qualsevol desenvolupador els pot fer servir com a backend per a la seva interfície d'usuari. D'aquesta forma obtindrem una aplicació on el seu backend estarà desplegat sobre la blockchain d'Ethereum i una interfície d'usuari que podrà ser també desplegada de forma descentralitzada o es podrà optar per centralitzar aquesta interfície d'usuari.

La pàgina d'Ethereum descriu els beneficis que pot aportar el desenvolupament d'una DApp [8]. Una DApp sempre tindrà disponibilitat, ja que una vegada es desplegui sobre la xarxa no podrà caure el servei. També implica privadesa, perquè no és necessari autenticar-te per desplegar un Smart Contract. Tampoc és possible censurar cap Dapp, la integritat de la informació sempre es mantindrà i es pot verificar el comportament del Smart Contract, ja que com s'ha comentat prèviament, aquest és públic per a qualsevol usuari. Totes aquestes característiques tenen relació amb què el Smart Contract es desplega sobre una blockchain, llavors les seves característiques es mantindran per l'Smart Contract.

A més d'aquests beneficis també trobem unes implicacions que imposa el fet de desenvolupar una DApp. Per exemple el fet de mantenir la Dapp, ja que si aquesta es desplega sobre una blockchain, una de les característiques és la immutabilitat, per tant no serà possible modificar el Smart Contract una vegada sigui desplegat sobre la xarxa. També poden tenir problemes a l'hora de processar la informació, perquè tota aquesta informació haurà de ser incorporada a la xarxa per algun miner de la blockchain. Això pot comportar que es trigui un cert temps a processar la consulta que vulgui fer l'usuari, a més de les despeses en Gas que puguin comportar les transaccions que realitzi l'usuari.

3 OBJECTIUS

L'objectiu principal del projecte és desenvolupar una DApp que pugui funcionar sobre la xarxa d'Ethereum. Per a poder aconseguir aquest objectiu serà necessari realitzar dues aplicacions. Per una part una interfície web, la qual serà una aplicació web. I per l'altra banda un Smart Contract, que serà una aplicació en Solidity. Per completar aquests objectius s'han proposat una sèrie de tasques a completar, per a poder assolir l'objectiu del projecte. Per tant s'han fixat unes tasques que el projecte ha de complir per considerarlo finalitzat, aquestes tasques es van comentar en l'informe inicial del projecte. En l'apartat de desenvolupament d'analitza la feina que s'ha realitzat de les tasques següents:

- Registrar l'usuari.
- Crear una campanya de crowdfunding que es guardarà a l'Smart Contract.
- Donar fons a una campanya mitjançant l'Smart Contract.
- Mostrar informació sobre la campanya en la interfície d'usuari.

4 METODOLOGIA

La metodologia que es va seleccionar per al projecte va ser la metodologia Kanban. Aquesta metodologia està enfocada a portar a terme les tasques pendents mitjançant l'ús de targetes on escriurem les tasques i tres columnes o més. Les bàsiques són una columna per les tasques pendents, un altre per les que es troben en procés i altre per les finalitzades. Com a principals característiques podem destacar que aquesta metodologia permet visualitzar totes les tasques, tant els pendents com les que es troben en procés com les finalitzades. També permet prioritzar segons la importància i la urgència de la tasca i a més permet fer un seguiment del temps. Altres característiques són la transparència, el fet d'evitar tasques ineficients, tenir control sobre les tasques i la flexibilitat [9].

Seguint amb la metodologia s'han definit unes tasques obtingudes dels objectius que es van definir per al projecte. Seguidament es va confeccionar un taulell per mantenir el seguiment d'aquestes tasques d'acord amb la metodologia. A la Figura 1 es pot veure el taulell utilitzat, en aquest cas es mostra el taulell en les fases finals del projecte. Aquesta metodologia ha permès una alta flexibilitat a l'hora de desenvolupar el projecte. Les tasques definides donaven llibertat per a desenvolupar les diferents subtasques que es necessitaven per a completar la tasca.

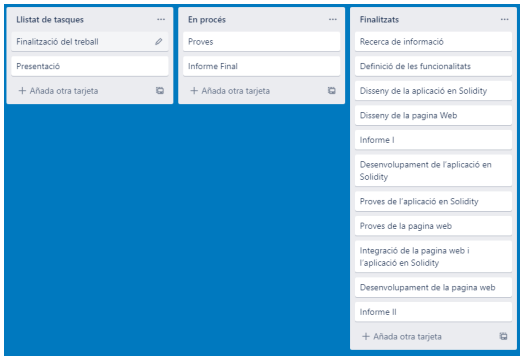

Fig. 1: Taulell Kanban

5 PLANIFICACIO

El projecte s'ha planificat primer pensant en les possibles tasques que s'haurien de resoldre i així complir amb la metodologia escollida. Les tasques que es van designar en l'informe inicial són les següents:

- La recerca d'informació.
- La definició de les funcionalitats.
- El disseny de la pàgina Web.
- El disseny de l'aplicació en Solidity.
- Fer l'informe I.
- Desenvolupament de l'aplicació en Solidity.
- Fer proves de l'aplicació en Solidity.
- El desenvolupament de la pàgina web.
- Fer proves de la pàgina web.
- Fer l'informe II.
- Realitzar la integració de la pàgina web i l'aplicació en Solidity.
- Fer proves generals.
- Finalitzar el treball.
- Fer l'informe Final.
- Fer la presentació del projecte.

Totes aquestes tasques es van ordenar per a poder realitzar un diagrama de Gantt com podem veure a la Figura 2.

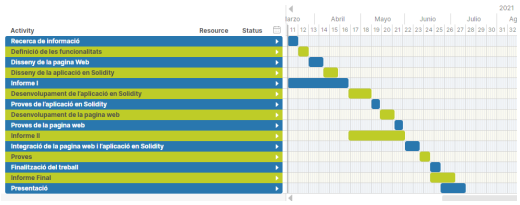

Fig. 2: Diagrama de Gantt

El plantejament inicial volia a dur a terme gairebé totes les tasques una darrere l'altra, prioritzant les que en un principi es consideraven més critiques per poder tenir marge de temps.

5.1 Modificacions

Durant el desenvolupament del projecte s'han realitzat unes quantes modificacions, perquè moltes tasques s'han realitzat de forma paral·lela. Gràcies a la metodologia Kanban aquestes modificacions s'han pogut realitzar sense gaires inconvenients. D'aquesta forma, tant el disseny com el desenvolupament de la interfície d'usuari com el de l'aplicació en Solidity s'han realitzat a la vegada. Era necessari tenir en compte el comportament de les dues per a la seva integració final.

6 DISSENY

En aquest subapartat s'explica de forma detallada en què ha consistit el disseny, tant l'aplicació web com de l'aplicació en Soldity.

6.1 Disseny de la pagina web

El disseny de la pàgina web ha consistit en dues parts, una del disseny gràfic de l'aplicació i l'altra del comportament que hauria de tenir. Per a això s'han realitzat diagrames de flux i de casos d'ús per a tenir en compte el diferent comportament que s'esperarà de l'aplicació.

Pel que fa al disseny gràfic, primer es va començar amb un esborrany només en HTML i CSS, per a tenir en compte on col·locar els diferents elements i quin aspecte se li volia donar a la pàgina web. La primera aproximació es pot observar a la figura 3. Respecte a la part de la base de dades amb la que contactarà el web no s'ha realitzat cap diagrama, perquè es farà ús d'una base de dades no relacional. Aquesta es va considerar una bona opció, ja que la quantitat d'informació que s'haurà de guardar a la base de dades no serà excessivament elevada ni era necessari relacionar-la entre les dades que s'emmagatzemin.

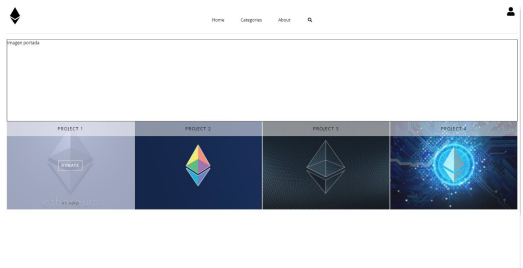

Fig. 3: Primera aproximacio del disseny de la interficie

6.2 Disseny de l'aplicacio en Solidity

Sobre l'aplicació en Solidity, es va realitzar conjuntament amb el disseny de la pàgina web, ja que els diferents diagrames que es van realitzar es tenia en compte el comportament de la pàgina web i en altres el comportament conjunt de les dues aplicacions. També es van tenir en compte les diferents bones pràctiques per codificar l'Smart Contract i es van intentar definir les diferents funcions necessàries a desenvolupar. Una vegada es va tenir identificades quines funcions serien necessàries, es van començar a realitzar una primera aproximació sobre la seva funcionalitat, gràcies al Remix IDE, que ofereix moltes eines de desenvolupament i proves aquest procés va resultar senzill.

6.3 Disseny general de l'aplicacio

A més dels dissenys respectius de l'aplicació web i de l'aplicació en Solidity, es va realitzar un diagrama general per a tenir una visió general del projecte. Tenint en compte les diferents parts que el formen, com es pot observar en la Figura 4, l'usuari serà el que interactua amb la interfície gràfica. Aquesta interfície contactarà amb un servei web que a la vegada tindrà una base de dades. Tot aquest servei web estarà desenvolupat amb JavaScript i Web3.js. Com a intermediari entre l'Smart Contract i el servei web es farà servir Metamask i per últim podem observar que aquest Smart Contract es trobarà a la blockchain d'Ethereum. Totes aquestes parts compondran la DApp que es vol desenvolupar per aquest projecte.

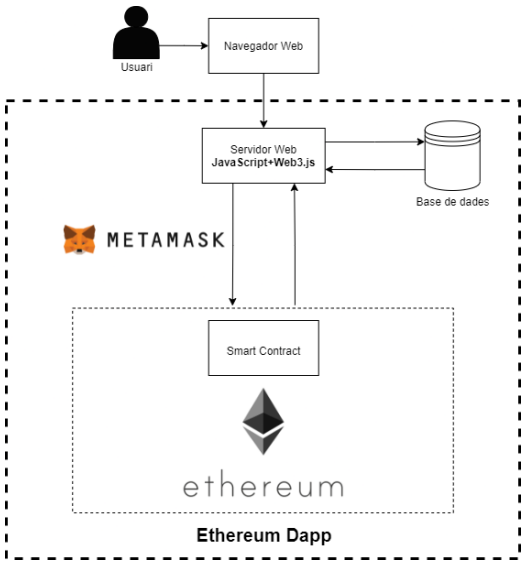

Fig. 4: Diagrama general

6.4 Problemes i solucions

El principal problema que va sorgir és relatiu a la planificació, on es van fer a la vegada els dos dissenys perquè semblava una decisió més adequada per al desenvolupament del projecte. Gràcies a fer servir la metodologia Kanban i la flexibilitat que permet no va resultar un problema gaire elevat.

7 Desenvolupament

En aquest apartat s'explicarà el desenvolupament de la pàgina web i de l'aplicació en Solidity, explicant també les tecnologies utilitzades i els problemes sorgits i les seves solucions.

7.1 Desenvolupament de la interfície web

Per a la interfície web s'ha fet servir Firebase, un servei de Google [10] com a hosting per la pàgina web i per a poder emmagatzemar la informació necessària. Firebase ofereix un servei de hosting, un server d'emmagatzematge i una base de dades no relacional entre altres. Pel que fa a la part de hosting, Firebase dóna les eines suficients per a poder pujar una pàgina web a la xarxa, oferint un domini [11] i un certificat TLS. L'emmagatzematge es farà servir per a poder emmagatzemar les imatges que els usuaris vulguin fer servir per a mostrar en les seves campanyes.

La base de dades servirà per a guardar la informació que no és necessària que es guardi a la blockchain. D'aquesta forma s'intentarà reduir el cost de guardar tota la informació de la campanya a la blockchain. Com menys informació es guardi a la blockchain menys costos serà per a l'usuari, llavors tant el títol com les descripcions de la campanya i la categoria seran guardades en aquesta base de dades. Tota aquesta part és un petit backend per a l'aplicació. S'ha optat fer-ho de forma centralitzada perquè el fet de desenvoluparho tot de forma descentralitzada afegia una complexitat que podria significar una falta de temps per a finalitzar el projecte.

Respecte a la part d'interfície gràfica d'usuari, s'ha fet servir el framework React.js [12] i com a mòdul gràfic s'ha optat per React Bootstrap [13]. D'aquesta forma es reduirà el temps per a fer una interfície gràfica que sigui una mica vistosa per a l'usuari. Tot i que inicialment es volia fer servir el framework Vue.js, aquesta opció es va descartar (veure les raons a l'Apartat 7.2). A més s'ha fet servir Web3.js, aquest mòdul serveix per a poder connectar la pàgina web amb la blockchain d'Ethereum, d'aquesta manera podrem recollir la informació i poder interactuar tant amb la blockchain com amb l'usuari. L'usuari necessitarà el proveïdor Metamask [14] en el navegador per a poder-se connectar amb la pàgina i amb l'Smart Contract.

A la interfície web podem trobar 5 pàgines diferents, la pàgina d'inici, la concreta per cada campanya, un formulari per a crear una nova campanya, una pàgina on mostrarà totes les donacions dels usuaris i un altre on mostrarà si té una campanya activa. Aquestes dues últimes permetran als usuaris poder recuperar els diners o recollir les donacions de la campanya si es compleix amb la quantitat fixada i és abans del temps fixat, a més d'una barra de navegació que trobarem a totes 5 pàgines.

S'ha tingut en compte la visualització de la pàgina web per a dispositius mòbils. Aquesta consideració s'ha tingut en compte pel cas en què en els dispositius mòbils es pugui fer un ús similar al que es fa en un navegador d'escriptori de Metamask o d'altre proveïdor. De moment no es pot fer servir el web per a realitzar donacions, però sí que és possible visualitzar la pàgina d'inici.

A la barra de navegació trobarem un enllaç a la pàgina inicial i un enllaç extern a la pàgina Etherscan on trobarem el Smart Contract. Etherscan ens ofereix la possibilitat de veure les transaccions en les quals intervé el Smart Contract, també podem veure la descompilació del codi del contracte. Seguint amb la barra de navegació trobem un botó que permet connectar Metamask amb la pàgina web. Internament aquesta connexió es fa amb el mòdul Web3.js que s'ha comentat prèviament. Una vegada connectat es comprovarà que la wallet de Metamask estigui en la xarxa adient, en aquest cas la xarxa testnet Kovan. Una vegada escollida la xarxa adient, la pàgina detectarà automàticament que hi ha un usuari vàlid amb una wallet de Metamask i mostrarà a la barra de navegació la seva adreça d'Ethereum i un logotip d'un usuari. Al clicar es desplegarà un menú que permetrà a l'usuari a anar al formulari de creació de la campanya, la barra de navegació es pot observar a la Figura 5.

Fig. 5: Barra de navegacio

A la pàgina d'inici es realitzarà una trucada a la base de dades de Firebase. Firebase proveeix dos tipus de bases de dades, la Firestore Database [15] i la Realtime Database [16], encara que totes dues tenen un funcionament similar. Totes dues són no-relacionals i en aquest projecte s'ha fet servir la Realtime Database. En aquesta base de dades s'emmagatzemarà la informació de les campanyes que es creïn. Això ho farem servir en aquesta pàgina inicial, on es mostraran totes les campanyes que s'han creat, com es pot veure a la Figura 6. A diferència del disseny inicial en aquest s'ha optat per una pàgina en mode fosc, s'ha considerat que actualment és una millor opció per a les pàgines web. Aquestes solen ser més vistoses i s'intenta evitar enlluernar excessivament a l'usuari.

A més la pàgina d'inici la podrà veure tothom encara que no tinguin Metamask. Tot i això es demanarà mitjançant una alerta, que s'instal·li. També es tindrà en compte que s'estigui utilitzant la xarxa correcta.

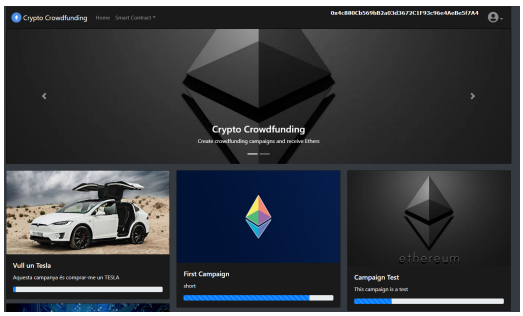

Fig. 6: Pantalla inicial

A la pàgina de cada campanya es fa servir una crida a la blockchain per a recollir la quantitat de fons que s'han recollit per a la campanya. Això significa que serà necessari que l'usuari tingui la wallet de Metamask connectada per a poder mostrar la informació de la campanya. També serà necessari perquè des d'aquí serà possible realitzar les donacions a la campanya amb un petit formulari on es demanarà la quantitat en Ethers que es vol donar a la campanya. Tot això ho podem veure a la Figura 7. En aquest cas no es mostrarà aquesta pàgina de forma correcta si no s'està connectat amb Metamask a la pàgina. Això és així perquè en aquesta pàgina es fa ús de crides a la blockchain per recollir informació, si no tenim Metamask, aquesta informació no seria possible d'aconseguir.

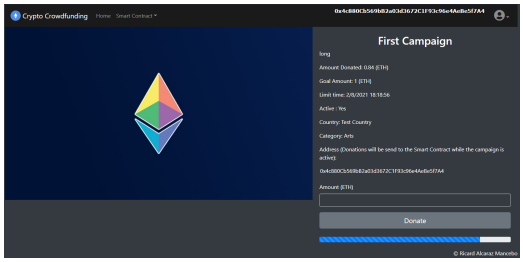

Fig. 7: Pantalla de una campanya

Sobre la pàgina de creació de les campanyes, aquesta consisteix en un formulari on es demanaran dades de la campanya. A partir d'aquest formulari serà possible la inserció de dades a la base de dades i algunes dades que aniran a la blockchain mitjançant l'Smart Contract. Es demanarà, un títol, una descripció curta, una descripció més llarga, la categoria a la qual es pensa que pertany, la quantitat que es vol aconseguir en Ethers, el país en el qual es troba el creador de la campanya i una imatge que descrigui la campanya. A més es mostrarà, però no es podrà modificar, l'adreça d'Ethereum del wallet de Metamask que tingui l'usuari connectat i la quantitat que s'ingressa en Ethers tindrà una correspondència en Dolars nord-americans com a referència. Per a fer això necessitem el valor actual del Ether en Dolars, per a realitzar això es fa una trucada a l'API de la pàgina Cryptocompare [17].

També es tindrà en compte que l'usuari pugui modificar les descripcions i la categoria de la campanya si és necessari. Les adreces només podran crear únicament una campanya, després no serà possible crear un altre amb la mateixa adreça, això s'ha trobat convenient perquè d'aquesta manera s'intenta assegurar la privadesa dels usuaris obligant-lo a fer servir una adreça nova cada vegada que vulgui crear una campanya. A més aquesta consideració disminuïa la complexitat del Smart Contract, que també significa que el cost sigui més baix a l'hora de desplegar-lo, podem veure el disseny a la Figura 8.

Per altra banda, si l'usuari està connectat amb Metamask, aquest tindrà accés a les donacions que ha realitzat. Si té una campanya activa, en aquestes es tindrà en compte si l'usuari pot recollir els diners. En el cas de la campanya, si s'arriba als fons necessaris en el temps límit, l'usuari serà capaç de recollir la donació. En el cas de donar, si la campanya no arriba als fons en el temps límit, l'usuari podrà recollir la seva donació. S'ha de tenir en compte que cada interacció de l'usuari en la que hi hagi alguna modificació a

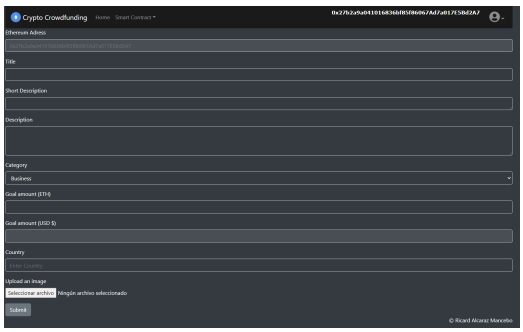

Fig. 8: Pantalla de creacio de la campanya

la blockchain implicarà un cost de transacció. Per tant quan creem la campanya, quan es realitzi una donació i a l'hora de retirar els diners, ja sigui per recollir els fons de la campanya com si es volen recuperar els diners de la donació, implicarà un cost de transacció que haurà de pagar l'usuari. Això ho podem observar a la figura 9. El cost de transacció variarà depenent de diferents factors, entre ells la velocitat a la qual es vol que la transacció sigui inclosa a la blockchain i el cost del Gas.

Fig. 9: Pagina de la campanya de l'usuari

7.1.1 Desenvolupament de l'aplicacio en Solidity

Per a l'aplicació en Solidity s'ha fet servir el llenguatge de programació anomenat Solidity. Solidity és el llenguatge principal per a desenvolupar Smart Contracts per a la blockchain d'Ethereum [18]. El llenguatge Solidity és un llenguatge de programació orientat a objectes per a escriure Smart Contracts. La compilació del programa serà en bytecode i serà executat en l'Ethereum Virtual Machine [19]. En la Figura 10 podem observar un exemple de codi en Solidity.

```
pragma solidity ^0.4.0;

contract SimpleStorage {
    uint storedData;

    function set(uint x) {
        storedData = x;
    }

    function get() constant returns (uint) {
        return storedData;
    }
}
```

Fig. 10: Exemple de codi en Solidity

S'han creat cinc funcions que seran essencials per al correcte funcionament de l'Smart Contract. Tenint en compte la informació mínima necessària a incloure a la blockchain i la informació que serà necessària recuperar de la blockchain. Hi ha diferents tipus de funcions que permet Solidity, en aquest projecte s'han fet servir de dos tipus. Aquestes són les funcions amb paràmetres i que permet retornar valors i les funcions view que prometen no modificar l'estat.

També s'han tingut en compte diferents bones pràctiques de seguretat per als Smart Contracts [20]. De forma que no sigui necessari en un futur desplegar un nou Smart Contract corregint errors i que només sigui necessari si es vol implementar altres funcions. Per això és necessari assegurar que el Smart Contract no tingui greus errors i que tingui la funcionalitat suficient perquè es pugui utilitzar.

En aquest projecte s'ha fet servir Remix IDE [21] per al desenvolupament de l'Smart Contract. Remix IDE és una aplicació web i de codi obert que es fa servir per al desenvolupament d'Smart Contracts, aquest ofereix moltes eines per a poder compilar i desplegar el teu Smart Contract tant a la xarxa mainnet, a les xarxes testnet. A més ofereix un entorn propi a l'aplicació per poder fer proves desplegant contractes i conté adreces amb fons per a realitzar proves.

Finalment s'ha realitzat un desplegament, en aquest cas s'ha realitzat en la testnet Kovan d'Ethereum. No s'ha fet servir la xarxa principal a causa dels elevats costos que implica desplegar un nou Smart Contract. S'ha fet una anàlisi de quin seria el cost aproximat d'aquest desplegament, fent ús de la pàgina Eth Gas Station [22]. Aquesta pàgina aproxima que el cost del desplegament d'un Smart Contract, en aquest cas s'ha aproximat que seria d'uns 97 dolars en el moment en el qual es va elaborar l'anàlisi. Aquest cost pot variar, depenent del preu de l'Ether i la quantitat d'usuaris que estan fent servir la xarxa en el moment.

7.2 Problemes i solucions

Durant el desenvolupament del projecte han sorgit diferents problemes, que s'han intentat tractar per seguir amb el desenvolupament. Un dels problemes inicials pel que fa al desenvolupament de la pàgina web és el fet que inicialment es tenia pensat fer servir el framework Vue.js. El problema va sorgir a l'hora de buscar exemples actualitzats de DApps desenvolupades amb Vue.js. Després d'una sèrie d'intents es va optar per fer servir el framework React.js. Aquest framework actualment és el més utilitzat per a desenvolupar DApps, això va significar trobar exemples més actuals de DApps desenvolupades amb aquest framework i a la vegada va simplificar el procés de desenvolupament d'aquest projecte perquè oferia solucions més actuals.

Un dificultat a mencionar és tot l'asincronisme que envolta a aquest projecte. Les insercions i la recollida d'informació de la base de dades, a través de les crides a les funcions han de ser asíncrones. Per tant s'han de tractar de forma que tota la informació s'arribi a mostrar per pantalla a l'usuari sense cap problema. També per aquest fet es va optar per Vue.js inicialment, però React.js també ofereix reactivitat dels elements que es mostren al web. Això significa que encara que es carregui la pàgina i no estigui llesta la informació, apareixerà en el moment en el qual arribi aquesta informació.

8 INTEGRACIO

La part de la integració consisteix a integrar les dues aplicacions desenvolupades, la pàgina web com a frontend i l'Smart Contract com a backend. Aquesta integració és possible gràcies al mòdul Web3.js que permet la interacció entre

la interfície d'usuari i l'Smart Contract. Per a fer que la integració funcioni serà necessari que l'usuari indiqui un proveïdor d'Ethereum, en aquest cas Metamask. La integració consistirà a indicar la direcció de l'Smart Contract que prèviament s'haurà de desplegar sobre una de les xarxes d'Ethereum i també haurem de proveir l'ABI del contracte. L'ABI serà la que permetrà aquesta interacció, ja que proveirà la informació de les funcions de l'Smart Contract i seguidament podrem interactuar amb l'Smart Contract. El mòdul Web3.js ofereix diferents formes d'interactuar amb l'Smart Contract, depenent de quin tipus de funció de l'Smart Contract sigui amb la que volem interactuar.

8.1 Problemes i solucions

Un dels principals problemes a l'hora de realitzar la integració és l'asincronisme. La crida a les funcions es realitza de forma asíncrona, per tant s'haurà de tenir en compte a l'hora de desenvolupar l'aplicació. En aquest cas també hi ha un punt més a considerar, perquè quan es realitza una transacció a la blockchain aquesta pot trigar un temps indeterminat del moment en què es realitza fins que es confirma a la blockchain. En aquest cas no s'ha tingut en compte aquest període i només s'ha tingut en compte el temps que es triga a fer la transacció per al comportament posterior a fer una donació o en crear una campanya.

9 Resultats respecte als objectius

En aquest apartat s'expliquen els resultats respecte a les tasques assignades per a completar els objectius. Tal com s'ha comentat prèviament en l'apartat d'objectius del projecte, aquestes tasques són les que s'han considerat necessàries per a considerar-lo finalitzat.

- **Registrar l'usuari:** Durant el desenvolupament de l'aplicació s'ha deixat de banda un registre complet de l'usuari. En un principi es tenia plantejat demanar més informació a l'usuari, com pot ser un email, una contrasenya, un nom d'usuari i una segona opció connectant-se amb Metamask. Finalment es va decidir que era millor oferir més privadesa a l'usuari i només oferir connectar-se amb Metamask. A més també aquesta opció resultava la menys complexa per a desenvolupar, que oferia un inici de sessió amb usuari i contrasenya. A l'hora de voler donar a una campanya o de crear-ne una seria necessari de totes formes fer ús de Metamask.

- **Crear una campanya de crowdfunding que es guardarà a l'Smart Contract:** Per a completar aquest objectiu era necessari tenir la interfície web, l'Smart Contract i la integració completada. Les campanyes que es vulguin crear es guardaran tant a la petita base de dades que tenim a Firebase com la informació bàsica de la campanya a la blockchain mitjançant l'Smart Contract. En concret en l'Smart Contract es guarda la quantitat d'Ethers que s'ha donat a la campanya, la quantitat que la campanya vol aconseguir, el temps límit, una variable per saber si s'ha creat una campanya amb l'adreça que s'està fent servir i una llista dels donants a la campanya amb la quantitat que ha donat. Tota aquesta informació es guardarà per cada campanya, i la identificarem amb l'adreça que es faci servir per a crear la campanya. Mitjançant la pàgina de creació de la campanya l'usuari haurà de donar tota la informació que demana el formulari, a més haurà de realitzar una transacció enviant aquesta informació l'Smart Contract.

- **Donar fons a una campanya mitjançant l'Smart Contract:** Aquest objectiu, com l'anterior, serà necessari que el projecte estigui en la seva fase final per a saber si s'ha pogut completar. En aquest cas de fer una donació a una campanya serà una funció de l'Smart Contract. Quan l'usuari interactuï amb la interfície per a donar a una campanya també ho farà amb l'Smart Contract mitjançant Metamask. L'usuari haurà d'introduir la quantitat d'Ether que vol donar, sempre que ell els tingui a la wallet de Metamask. Fer aquesta donació també comportarà un cost en Gas com a taxa, aquesta taxa serà la que Metamask doni per defecte. Una vegada confirmada la transacció el Smart Contract guardarà aquesta quantitat a l'espera que la campanya compleixi amb la quantitat que tingui com a objectiu, de forma que el creador de la campanya reculli els fons o pel contrari que l'usuari recuperi la donació en el cas en el qual la campanya no arribi a l'objectiu.

- **Mostrar informació sobre la campanya en la interfície d'usuari:** Aquest objectiu té dues parts, una el fet de mostrar la informació que es troba a la base de dades i una altra mostrar la informació que es troba l'Smart Contract. La informació que es troba a la base de dades la mostrarem en la pàgina d'inici. D'aquesta forma no necessitarem connectar-nos a la blockchain i així mostrar les imatges de les campanyes, el seu títol i una petita descripció de la campanya. En el cas que es vulgui més informació de la campanya haurem d'anar a la seva pàgina, en aquesta sí que serà necessari connectar-nos amb Metamask per a obtenir la informació de la campanya i poder donar a la campanya que vulguem. En aquesta pàgina obtindrem tota la informació que s'ha comentat en l'objectiu de crear una campanya.

10 Futures linies de treball

Aquest projecte compleix amb la funcionalitat essencial per a poder crear campanyes, fer donacions a aquestes campanyes i també poder treure els fons rebuts en les campanyes o donats en les campanyes que no arribin al seu objectiu. Però hi ha diferents millores per al futur del projecte que es tractaran d'explicar en aquest apartat. Les millores poden ser en l'Smart Contract o en la interfície web, per tant s'ha decidit dividir-ho en subapartats.

10.1 Millores l'Smart Contract

Una possible millora de l'Smart Contract seria el fet de poder automatitzar la recollida dels fons obtinguts o retornarlos als donants. Per a que això fos possible seria necessari estudiar si és viable fer servir l'ús d'Oracles d'Ethereum. Aquests permeten als Smart Contracts obtenir informació

que no es troba a la blockchain, d'aquesta manera una possibilitat seria que el Smart Contract tingues en compte la data actual que li proveís un Oracle i que el Smart Contract fos capaç de realitzar-ho automaticament. Això també significaria que l'Smart Contract necessitaria fons propis per a poder realitzar aquesta automatització. Una forma podria ser que l'Smart Contract s'emportés una comissió per cada campanya que és crees o per cada donació a una campanya. Aquesta seria una petita aproximació a una possible millora de l'Smart Contract. Una altra millora podria ser dedicar més temps a l'optimitzacio de l'Smart Contract, d'aquesta manera es podria intentar reduir els costos de desplegar-lo a la blockchain. Com s'ha comentat prèviament en el moment de l'anàlisi de quant podria costar, aquest era el voltant dels 97 Dòlars nord-americans. Optimitzant-lo potser es podria aconseguir alguna millora en aquest aspecte.

10.2 Millores a la interfície web

Respecte a la pàgina web, una possibilitat seria ampliar quins proveïdors per a interactuar amb la blockchain es poden fer servir. En aquest projecte només s'ha realitzat proves amb Metamask, però també existeixen altres proveïdors que es podrien fer servir. Una altra millora seria el fet de descentralitzar completament la pàgina web, això seria possible gràcies a IPFS. Aquesta és una xarxa peer-to-peer que permet emmagatzemar i compartir fitxers, fent ús d'adreces úniques als continguts d'aquesta xarxa. D'aquesta forma seria possible descentralitzar completament l'aplicació i es podria oferir un servei més confiable, ja que estaríem eliminant qualsevol entitat centralitzada. En aquest projecte encara hi ha una petita part centralitzada que és la base de dades que es fa servir, llavors aquesta part podria generar certa desconfiança i amb IPFS s'eliminaria. A la vegada es deixaria de fer ús de Firebase, que al final és un servei que ofereix Google, per tant eliminaríem qualsevol informació que Google estigues emmagatzemant sobre la pàgina o la base de dades. Una altra millora podria ser oferir a l'usuari una mica més de configuració del seu perfil si així volgués, com podria ser una imatge de perfil o informació que vulgui aportar el mateix usuari. Tot això seria opcional per a l'usuari sense obligar-lo a donar les seves dades. Una altra millora podria ser el fet de trobar algun proveïdor amb què es poguessin fer les crides per recollir informació sense que l'usuari estigui connectat amb Metamask. D'aquesta manera l'usuari no podria donar a les campanyes, però sí que podria accedir a tota la informació de la campanya.

10.3 Altres millores

A més de les possibles millores a la interfície web o l'Smart Contract també existeix la possibilitat de migrar l'aplicació a un altre blockchain. Ethereum actualment és la blockchain més popular per al desenvolupament de DApps, tot i això s'han de tenir en compte els diferents aspectes que poden fer que s'opti per la migració a un altre blockchain. Un d'aquests són els costos per a desplegar un Smart Contract, en el cas d'aquest projecte no s'ha desplegat en la xarxa principal. Però seria necessari si es vol que l'Smart Contract es fes servir, el desplegament tindria un cost d'aproximadament 97 USD. Com s'ha comentat existeix la possibilitat que es pugui optimitzar, però tot i així s'han de fer front a altres despeses com les taxes de transacció. Aquestes les hauria de fer front l'usuari i també varien en funció del moment en què es realitzi la transacció. Ethereum vol llançar la seva versió 2.0 on passaria del sistema actual de Proof-of-Work al sistema de Proof-of-Stake. Aquesta modificació podria significar una baixada significativa dels costos de desplegament i de transaccions, encara que seran necessaris uns anys per a contemplar aquesta possibilitat. Mentrestant es podria fer una anàlisi d'altres blockchain que permetin la funcionalitat de l'Smart Contract i que a la vegada permetin desenvolupar en Solidity. I aixi fer que no sigui necessari modificar l'Smart Contract i que les despeses en aquestes blockchains siguin més baixes que en Ethereum.

11 CONCLUSIONS

En vistes als resultats, podem determinar que aquest treball s'ha pogut completar satisfactòriament complint amb els objectius que es van marcar en un inici. Tot i la complexitat que implicava el projecte on el desenvolupament no és el tradicional per a una aplicació web i és necessari tenir present altres aspectes. Com pot ser tota la programació dels Smart Contracts i com aquests han d'interactuar amb la pàgina web.

Per al desenvolupament del projecte s'ha fet ús de la metodologia Kanban. Aquesta ha permès un alt grau de flexibilitat a l'hora de desenvolupar les tasques. A la vegada s'han determinat tasques que no fossin excessivament concretes ni generals, de forma que també s'afavoria aquesta flexibilitat.

La planificació del projecte ha anat variant durant el desenvolupament. Algunes tasques requerien una mica més de temps del previst i a la vegada altres s'han pogut completar abans. En general tenint en compte això els terminis s'han completat sense passar-se del temps límit per a la finalització del projecte.

Fent una visió més general del projecte, podem dir que s'han realitzat dues aplicacions. Una aplicació web desenvolupada amb el framework React.js i una aplicació en Solidity, a més de la integració d'aquestes dues aplicacions. També s'ha treballat en menor mesura amb una base de dades no relacional.

Finalment es considera satisfactori el resultat final del projecte, amb els objectius complets i amb possibilitats de millores futures. Personalment considero que aquest treball m'ha semblat interessant de realitzar i a la vegada he après coses sobre la interacció de les interfícies web amb els Smart Contracts. Aquest coneixement considero que em serà útil de cara al futur.

AGRAÏMENTS

Volia agrair a en Jordi Herrera Joancomartí pel seu suport resolent els dubtes sorgits i pel seu coneixement aportat al projecte. També volia agrair a les persones que s'han interessat pel meu projecte i m'han donat suport. Gràcies a tothom.

REFERÈNCIES

[1] "Bitcoin: A Peer-to-Peer Electronic Cash System,"bitcoin.org (juny 2021). [En línia]. Disponible: https://bitcoin.org/bitcoin.pdf

[2] "Ethereum Whitepaper,"ethereum.org (juny 2021). [En línia]. Disponible: https://ethereum.org/en/whitepaper/

[3] "What is Ethereum?," ethereum.org (juny 2021). [En línia]. Disponible: https://ethereum.org/en/what-is-ethereum/

[4] "Ethereum,"ethereum.org (juny 2021). [En línia]. Disponible: https://ethereum.org/en/whitepaper/#ethereum

[5] "Kovan Testnet The Fast and Reliable Ethereum Test Chain," Kovan Testnet (juny 2021). [En línia]. Disponible: https://kovan-testnet.github.io/website/

[6] "Ethereum accounts," ethereum.org (juny 2021). [En línia]. Disponible: https://ethereum.org/en/developers/docs/accounts/

[7] "Gas and fees," ethereum.org (juny 2021). [En línia]. Disponible: https://ethereum.org/en/developers/docs/gas/

[8] "Introduction to dapps," ethereum.org (juny 2021). [En línia]. Disponible: https://ethereum.org/en/developers/docs/dapps/

[9] "Metodología Kanban: en qué consiste y cómo utilizarla," APD España, 08-Jun-2021 (juny 2021). [En línia]. Disponible: https://www.apd.es/metodologia-kanban/

[10] Google (juny 2021). [En línia]. Disponible: https://firebase.google.com/

[11] Crypto Crowdfunding (juny 2021). [En línia]. Disponible: https://crypto-crowdfunding.web.app/

[12] "React – Una biblioteca de JavaScript para construir interfaces de usuario," – Una biblioteca de JavaScript para construir interfaces de usuario (juny 2021). [En línia]. Disponible: https://es.reactjs.org/

[13] "Bootstrap," react (juny 2021). [En línia]. Disponible: https://react-bootstrap.github.io/

[14] MetaMask (juny 2021). [En línia]. Disponible: https://metamask.io/

[15] "Cloud Firestore ; Firebase," Google (juny 2021). [En línia]. Disponible: https://firebase.google.com/docs/firestore

[16] "Firebase Realtime Database," Google (juny 2021). [En línia]. Disponible: https://firebase.google.com/docs/database

[17] "Cryptocurrency Prices, Portfolio, Forum, Rankings," CryptoCompare (juny 2021). [En línia]. Disponible: https://www.cryptocompare.com/

[18] "Solidity," Solidity (juny 2021). [En línia]. Disponible: https://solidity-es.readthedocs.io/es/latest/

[19] "Ethereum Virtual Machine (EVM)," ethereum.org (juny 2021). [En línia]. Disponible: https://ethereum.org/en/developers/docs/evm/

[20] Ethereumbook, "ethereumbook/ethereumbook," GitHub (juny 2021). [En línia]. Disponible: https://github.com/ethereumbook/ethereumbook/blob/develop/09smart-contracts-security.asciidoc

[21] "Ethereum IDE," Remix (juny 2021). [En línia]. Disponible: https://remix.ethereum.org/

[22] ETH Gas Station (juny 2021). [En línia]. Disponible: https://ethgasstation.info/