Entrades NFT per a esdeveniments: Implementació de transaccions segures mitjançant Smart Contracts de Ethereum

Treball Final de Grau 2024/25

Informe Inicial

UAB
Universitat Autònoma de Barcelona

Lucas Dalmau Garcés – 1636290

ÌNDEX

Informació preliminar

Motivació

Durant la carrera, s'han cursat diverses assignatures que han ofert diferents perspectives i coneixements rellevants en informàtica. No obstant això, a mesura que avançava amb assignatures com: *Informació i Seguretat* i *Fonaments de la Tecnologia de la Informació*, em van descobrir l'ús de la criptografia i vaig observar com s'utilitza en allò que per a mi era encara un món desconegut que em cridava l'atenció: La tecnologia de la cadena de blocs i la seva implementació en les criptomonedes. Sobretot com aquesta tecnologia s'utilitzava per aconseguir un sistema descentralitzat allunyat dels sistemes tradicionals als quals estem acostumats. És per això que quan aquest any vaig veure l'oportunitat de fer l'assignatura de Blockchain, em vaig inscriure.

En aquella assignatura vaig adquirir coneixements sobre el seu funcionament i ús. Un aspecte interessant sobre l'assignatura és el fet que culmina amb un projecte final on fas una mena de recerca sobre una tecnologia relacionada amb la Blockchain. En aquest punt, vaig seleccionar els tokens no fungibles (NFT), amb el qual vaig aprendre sobre l'estàndard que regeix aquesta tecnologia i el seu ús dintre d'Ethereum. És en aquest moment quan el tutor ens va parlar de com era possible que no s'hagués estandarditzat aquesta tecnologia per la gestió d'entrades gràcies a les propietats que té i és justament quan vaig pensar que aquesta podia ser una bona idea per a un treball perquè em proposaria tot un repte de combinar tecnologies apreses amb altres que no he vist mai. I això va ser el que em va motivar a iniciar aquest treball.

Fonts inicials: Una primera recerca

Aquestes setmanes he dedicat temps en tractar de modelar aquest treball i entendre quins reptes em suposarà.

Primerament, he estat llegint els meus apunts de l'assignatura de Blockchain per poder tenir més presents els coneixements sobre la xarxa d'Ethereum, com el funcionament de les transaccions, el Gas[1], els Smart Contracts, la generació de blocs a Ethereum, els mecanismes de consens, etc.

Una vegada tinc una idea una mica més àmplia, he començat a indagar en coses més específiques, per exemple, com funciona el Wallet de Metamask, per entendre una mica més quines són les seves capacitats i limitacions. Durant la cerca, he trobat informació sobre com connectar Smart Contracts amb Wallets, com fer les transaccions, etc. M'ha resultat interessant trobar que l'empresa darrere de Metamask dona facilitats als desenvolupadors per connectar-se a qualsevol xarxa com les Testnet o fins i tot per poder integrar-se amb xarxes locals per poder fer proves amb els Smart Contracts.

Tota aquesta recerca d'informació, em va plantejar quin IDE seria el més adient per aquest projecte. Personalment, he adquirit una amplia experiència amb Visual Studio Code, així que una tecnologia que permeti utilitzar-ho tindria un pes considerable a l'hora d'elegir.

[1] **Gas**: Unitat de mesura que representa el cost computacional i esforç que es requereix per executar una operació o transacció a la xarxa de Etehreum.

Vaig veure que la pròpia empresa de Metamask té un article justament sobre aquesta qüestió i planteja dues eines interessants:

- **Hardhat**: En un primer cop he observat que té una gran comunitat i té una extensió al Visual Studio Code per poder començar a fer projectes amb Solidity des de zero. A més, compta amb una guia per a principiants que pot ser de gran utilitat. Està basat en Node.js que és un Framework que conec àmpliament, ja que ens ho van ensenyar l'any passat a l'assignatura de Sistemes i Tecnologies Web i es pot integrar amb VSCode.

- **Foundry**: És relativament nou, està basat en Rust, que és un llenguatge de programació semblant a C i C++ que no he fet mai. També hi ha guies per a principiants i es pot integrar amb VSCode, cosa que fa que sigui un candidat vàlid.

Com a candidat addicional, dintre de l'assignatura de Blockhain que he cursat, el tutor ens va ensenyar un IDE Online anomenat Remix IDE que a més permet desplegar els Smart Contracts fàcilment per poder-los testejar.

Com a línia futura per al següent Informe, investigaré més a fons quina d'aquestes eines s'adapta millor al treball i a les meves necessitats.

Objectius i abast del treball

Si bé el que es vol reflectir en aquest projecte és la creació d'una solució basada en la Blockchain d'Ethereum per la compravenda d'entrades, això es pot aconseguir a diferents nivells, ja que estrictament parlant, un Smart Contract que posseeixi un diccionari d'adreces amb un array d'identificadors, compliria perfectament la missió i objectiu d'aquest treball. En contra part, tampoc es pot excedir la càrrega de treball, perquè hi ha una gran varietat de opcions complementàries i requisits que es podrien posar a aquest, fent inviable la seva realització en el temps establert. És per aquesta raó que considero que els objectius s'han de categoritzar en dos tipus: Aquells els quals permeten un correcte funcionament i acompliment del propòsit del treball (els objectius primaris), i els objectius els quals implementar-los poden augmentar el valor del treball, però impliquen una càrrega de treball encara desconeguda.

Objectius Primaris

- Permetre la compra d'entrades NFT en estoc.
- Permetre la venda d'entrades NFT.
- Permetre la consulta d'entrades NFT a la venda al mercat amb el seu preu.
- Permetre la compra d'entrades NFT que estiguin venent altres usuaris.
- Establir mesures preventives contra monopoli d'entrades, revenda massiva, inflació de preus, venda d'entrades ja bescanviades i compravenda d'entrades quan l'esdeveniment ha terminat.
- Complir amb els estàndards de seguretat per a evitar possibles vectors d'atac.

- Complir amb els estàndards dels NFT com el ERC721[2] i el ERC165[3]
- Poder consultar la legitimitat d'una entrada.

Una vegada desenvolupat el Smart Contract amb aquestes característiques, caldria desenvolupar una aplicació web que pugui interactuar amb el Smart Contract de forma que faci de pont/interfície entre el Wallet del usuari i el Smart Contract.

Objectius Secundaris

- Implementació avançada del codi perquè les transaccions siguin eficients pel que fa a la despesa de Gas.
- Establiment de rols com verificadors (un verificador seria l'operari encarregat de verificar la pertinença de l'entrada i bescanviar-la per l'accés a l'esdeveniment).
- Creació de codis QR per a una gestió d'accés als esdeveniments més fluida.
- Creació del token URI i implementació a un IPFS per afegir més informació a l'entrada (nom de l'esdeveniment, data i hora, preu de compra, informació postvenda...).
- Llençar el Smart Contract a una Testnet d'Ethereum sortint així del entorn local.

Plantejament de la metodologia

La metodologia per poder aconseguir el meu objectiu és senzilla, ja que tenim dues parts ben diferenciades:

- El '**Backend'**, que en aquest context seria el Smart Contract a desenvolupar, perquè s'encarregaria de tota la lògica de la compravenda dels NFT que fan d'entrades.

- El '**Frontend'**, que seria l'aplicació web. Aquesta ha de poder interactuar amb el Smart Contract i cridar les funcions que hi implementi perquè qualsevol usuari pugui comprar i vendre les entrades NFT sense haver d'interactuar directament amb el Smart Contract. En ser un aplicatiu web que involucra la interacció amb un Smart Contract l'anomenaré Descentralized Application, ja que és un terme més acurat a causa de la naturalesa de l'aplicació que incorpora l'ús d'un contracte intel·ligent d'una xarxa descentralitzada com Ethereum.

Una vegada definides aquestes dues parts, aprofundeixo una mica més en la metodologia per poder abordar aquest treball:

[2] **ERC-721**: Conjunt de regles establertes per a que es puguin manipular els NFT entre aplicacions i Smart Contracts .

[3] **ERC-165**: Regla que defineix un mètode per a que els Smart Contrats detectin i publiquin quins altres estàndards compleixen.

Smart Contract

Per aquesta part, és imprescindible aprendre Solidity, per poder aprendre-ho tractaré de desenvolupar Smart Contracts petits per anar provant les diferents característiques per posar en pràctica la teoria recollida al llibre “Mastering Ethereum: Building Smart Contracts and DApps”.

A més he fet una recerca d'altres fonts per poder aprendre i he trobat una pàgina que ensenya Solidity a partir d'exemples que pot ser una gran idea per guiar-me més en l’aprenentatge.

Una vegada après els conceptes importants tractaré de fer una recerca més exhaustiva sobre eines, llibreries i recursos que em puguin ser útils per al desenvolupament del Smart Contract.

Descentralized application

Per aquesta segona part, inicialment em centraria en la recerca d'informació sobre projectes en DApps per veure quines són les millors pràctiques, quins són els Frameworks / llenguatges més utilitzats de forma que, una vegada tingués una idea formada sobre com plantejar aquesta part del projecte, poder enfocar-me en l’aprenentatge sobre l'eina i el llenguatge que sigui més convenient.

De preferència, voldria fer servir vue.js com Framework per la Dapp perquè l’hem vista a classe i la corba de treball seria més senzilla, ja que tinc una certa experiència, però, no descarto que, si hi trobo algun altre Framework que sigui més usat o que tingui més suport per a projectes com l'exposat aquí, acabi aprenent un altre llenguatge o un Framework.

Una vegada acabat aquesta etapa d'aprenentatge començaria a fer la Descentralized App adaptant-la al Smart Contract perquè aquesta pugui fer-ne ús i que es pugui connectar a un Wallet com Metamask per poder fer transaccions funcionals.

Planificació del treball

Per a poder desenvolupar-ho tot de manera organitzada i aprofitant la segmentació del treball que he fet en l’apartat anterior. Es planificarà de la següent manera:

Primera Fase: Coincidint amb la 2a sessió de seguiment i el Lliurament informe de progrés I previst pel 13 d'abril de 2025 seria la data límit per haver après i desenvolupat el Smart Contract amb tots els objectius primaris i els objectius secundaris que aportin més al projecte i siguin factibles de fer.

Segona Fase: Aprofitant la data de la 3a sessió de seguiment i el lliurament de l'informe de progrés II, entre la finalització del segon lliurament i aquest tercer lliurament (25 maig de 2025) la meva proposta es haver après i desenvolupat la Descentralized App.

Tercera Fase: Coincidint amb la 4a sessió de seguiment i la Proposta d’informe final(15 juny de 2025), faria els últims retocs al treball per terminar d'encaixar el Smart Contract amb l'aplicació descentralitzada i tenir el projecte plenament funcional amb, com a mínim, tots els requisits primaris.

Per tenir una vista més específica de les tasques i la seva planificació, es disposa de la següent taula:

La taula comença en el dia 1 que es compta a partir del 15 de febrer i termina al dia 119 al 15 de juny coincidint amb la Proposta d'informe final.

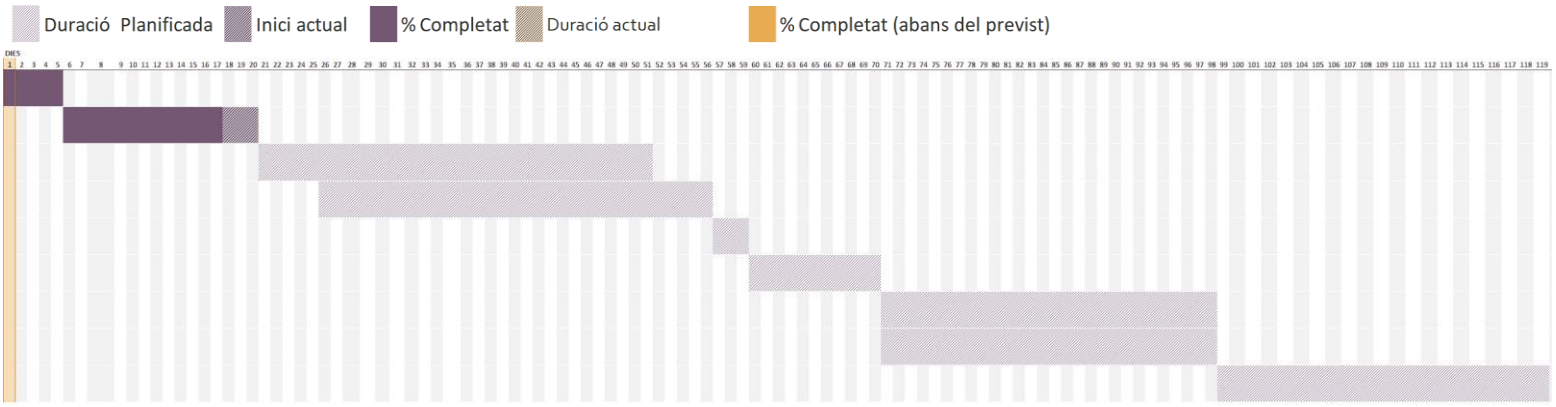
L'Inici Planificat és el dia en què es té planejat començar una determinada tasca, de moment, aquests dies coincideixen amb la columna d'Inici Real de les tasques que s'han començat. Les tasques encara no començades tenen un Inici Real de 0.

Respecte a la Duració Planificada, és el temps assignat a una tasca. En cas que la tasca sembli que trigarà més o menys (o té un començament diferent de l'Inici planificat) es veurà reflectit en la columna de Duració Actual. És per això que de moment les tasques tenen la mateixa Duració Planificada i Duració Actual.

TASQUES	INICI PLANIFCIAT	DURACIÓ PLANIFICADA	INICI REAL	DURACIÓ ACTUAL	PERCENTATGE COMPLETAT
Investigar sobre els Smart Contracts	1	5	1	5	100%
Aprendre Solidity	6	15	6	15	80%
Desenvolupar el Smart Contract	21	31	0	31	0%
Testejar el Smart Contract	26	31	0	31	0%
Investigar sobre la Dapp	57	3	0	3	0%
Aprendre un llenguatje per la Dapp	60	11	0	11	0%
Desenvolupar la Dapp	71	28	0	28	0%
Testejar la Dapp	71	28	0	28	0%
Acabar d'unficar Smart Contract i Dapp	99	21	0	21	0%

Per acabar, s'ha afegit una última columna en la qual es mostrarà el progrés de les tasques (Percentatge Completat).

Per veure de manera mes visual la taula, es disposa del següent diagrama de Gantt[4] :



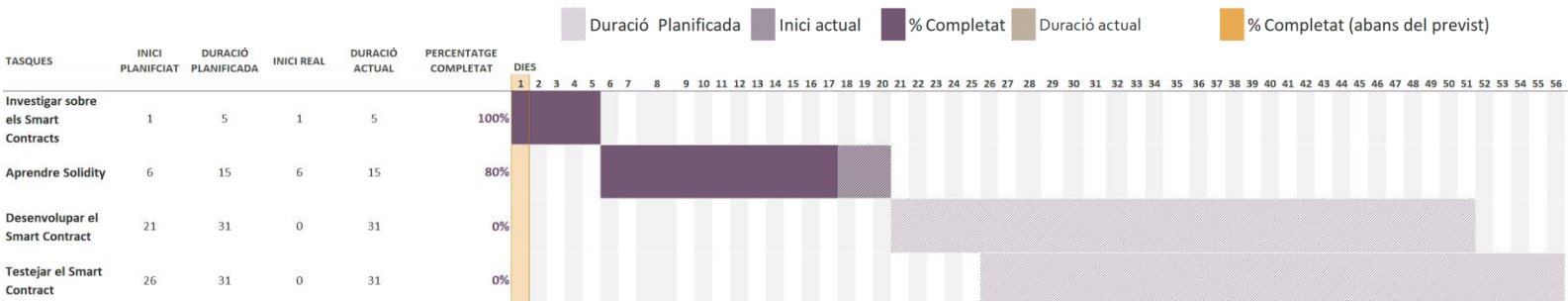
[4] La Duració actual no és visible per si sol, ja que de moment se superposa amb la Duració planificada.
D'altra banda, el %Completat(abans del previst) només serà visible si una tasca comença abans de la data planificada.

Anàlisi de la Primera Fase:

Aquesta fase comença el 27 de febrer de 2025 i acaba el 13 d'abril de 2025

Les Dates planificades per a cada tasca són:

- Investigar sobre els Smart Contracts (15 de febrer – 19 de febrer): Aquesta part ja ha estat completada durant la realització d'aquest informe, i es veurà reflectit al següent informe.
- Aprendre Solidity (20 de febrer – 6 de març): Aquesta tasca està quasi acabada seguint la planificació. Tracta d'entendre la lògica i metodologia per al desenvolupament del Smart Contract a la vegada que és desenvolupen petits Smart Contracts de prova per consolidar l'après i es veurà reflectit al següent informe.
- Desenvolupar el Smart Contract (7 de març – 6 d'abril): Tasca on es desenvoluparà el Smart Contract seguint els objectius establerts en aquest informe.
- Testejar el Smart Contract (14 de març – 13 d'abril): Comença dos dies després del començament del desenvolupament, ja que amb aquest marge es deixa prou temps per acostumar-se al projecte i desenvolupar les primeres funcions essencials.



Observació gràfica amb el diagrama de Gantt sobre la Primera Fase.

Bibliografia

ANDREAS M. ANTONOPOULOS; GAVIN WOOD PH.D. Mastering Ethereum : Building Smart Contracts and DApps. Sebastopol, CA: O'Reilly Media, 2018. Disponível em: https://research.ebsco.com/linkprocessor/plink?id=fcbc8282-5c2e-34a7-bde8-805e7ab76a41. Acesso em: 15 fev. 2025.

Proof-of-stake (PoS) [en línea], (sin fecha). ethereum.org. [Consultado el 15 de febrero de 2025]. Disponible en: https://ethereum.org/en/developers/docs/consensus-mechanisms/pos/

Blocks [en línea], (sin fecha). ethereum.org. [Consultado el 15 de febrero de 2025]. Disponible en: https://ethereum.org/en/developers/docs/blocks/

How to add a custom network RPC | MetaMask Help Center [en línea], (sin fecha). MetaMask Help Center | MetaMask Help Center. [Consultado el 15 de febrero de 2025]. Disponible en: https://support.metamask.io/configure/networks/how-to-add-a-custom-network-rpc/

Hardhat vs Foundry | MetaMask News [en línea], (sin fecha). The Ultimate Crypto Wallet for DeFi, Web3 Apps, and NFTs | MetaMask. [Consultado el 16 de febrero de 2025]. Disponible en: https://metamask.io/news/developers/hardhat-vs-foundry-choosing-the-right-ethereum-development-tool/

Hardhat for Visual Studio Code | Ethereum development environment for professionals by Nomic Foundation [en línea], (sin fecha). Hardhat | Ethereum development environment for professionals by Nomic Foundation. [Consultado el 16 de febrero de 2025]. Disponible en: https://hardhat.org/hardhat-vscode/docs/overview

Foundry Book [en línea], (sin fecha). Introduction - Foundry Book. [Consultado el 16 de febrero de 2025]. Disponible en: https://book.getfoundry.sh/

Remix - Ethereum IDE [en línea], (sin fecha). Remix - Ethereum IDE. [Consultado el 16 de febrero de 2025]. Disponible en: https://remix.ethereum.org/

Solidity by Example [en línea], (sin fecha). Solidity by Example. [Consultado el 18 de febrero de 2025]. Disponible en: https://solidity-by-example.org/