



IMPLEMENTACIÓ DE MODELS CONCEPTUALS  
DEDUCTIUS AMB BLOCKCHAIN I BASES DE  
DADES RELACIONALS

TREBALL DE FI DE GRAU EN ENGINYERIA INFORMÀTICA  
Facultat d'Informàtica de Barcelona, UPC

*Aniol Carbó I Marchuet*

Director: Joan Antoni Pastor Collado

Especialitat: Enginyeria del Software

Quadrimestre de primavera 2022

## **Resum**

Aquest treball de final de grau té com a objectiu demostrar l'aplicació dels Models Conceptuals Deductius a través de la combinació de blockchain i bases de dades relacionals. La proposta dels MCD's va ser introduïda per Antoni Olivé l'any 1989, quan encara no existia la tecnologia blockchain. Aquesta tecnologia presenta un seguit de semblances amb la proposta d'Olivé, i en aquest projecte es demostra la possible implementació pràctica d'aquest concepte per a obtenir un sistema d'informació funcional.

## **Resumen**

Este trabajo de fin de grado tiene como objetivo demostrar la aplicación de los Modelos Conceptuales Deductivos a través de la combinación de blockchain y bases de datos relaciones. La propuesta de los Modelos Conceptuales Deductivos fue introducida por Antoni Olivé el año 1989, cuando aún no existía la tecnología blockchain. Esta tecnología presenta varias similitudes con la propuesta de Olivé, y en este proyecto se demuestra la posible implementación práctica de este concepto para obtener un sistema de información funcional.

## **Abstract**

This final degree project has the purpose of demonstrating the application of the Deductive Conceptual Models through the combination of blockchain and relational databases. The Deductive Conceptual Models were introduced by Antoni Olivé on year 1989, before the blockchain technology was created. This very same technology presents multiple similarities with Olivé's proposal, so in this project we prove the possible practical implementation of this concept in order to accomplish a functional information system.

# Índex de continguts

<b>1</b>	<b>Introducció</b>	<b>7</b>
1.1	Context	7
1.1.1	Conceptes	7
1.1.2	Problema a resoldre	9
1.1.3	Stakeholders	10
1.2	Justificació	10
1.3	Abast	11
1.3.1	Objectius	11
1.3.2	Competències tècniques	12
1.3.3	Obstacles i riscos	12
1.3.4	Requisits funcionals	13
1.3.5	Requisits no funcionals	13
1.4	Metodologia i rigor	14
<b>2</b>	<b>Planificació</b>	<b>17</b>
2.1	Descripció de tasques	17
2.1.1	Gestió de projecte	18
2.1.2	Sprint 1	19
2.1.3	Sprint 2	21
2.1.4	Sprint 3	22
2.1.5	Memòria final	24
2.2	Estimació de tasques	24
2.3	Diagrama de Gantt	25
2.4	Gestió de riscos	28
<b>3</b>	<b>Pressupost</b>	<b>29</b>
3.1	Identificació dels costos	29
3.1.1	Recursos materials	29
3.1.2	Recursos humans	29
3.2	Estimació dels costos	31
3.2.1	Recursos genèrics	31
3.2.2	Cost de contingència	32
3.2.3	Cost d'imprevistos	33
3.2.4	Pressupost final	33
3.3	Control de gestió	33
<b>4</b>	<b>Informe de sostenibilitat</b>	<b>35</b>
4.1	Dimensió econòmica	35
4.2	Dimensió mediambiental	35
4.3	Dimensió social	36

<b>5</b>	<b>Disseny</b>	<b>37</b>
5.1	Arquitectura . . . . .	37
5.1.1	Frontend . . . . .	38
5.1.2	Backend . . . . .	38
5.1.3	Blockchain . . . . .	38
5.1.4	Bases de dades . . . . .	40
5.2	Domini . . . . .	40
5.2.1	Model de classes . . . . .	40
5.2.2	Casos d'ús . . . . .	41
<b>6</b>	<b>Desenvolupament</b>	<b>49</b>
6.1	Interacció amb Smart Contracts . . . . .	49
6.1.1	Aplicació web Angular . . . . .	51
6.2	Obtenció i interpretació de les dades . . . . .	52
6.2.1	Obtenció de les transaccions . . . . .	53
6.2.2	Interpretació de les dades . . . . .	56
6.3	Consulta de dades . . . . .	59
6.3.1	Vistes . . . . .	61
<b>7</b>	<b>Anàlisi de rendiment</b>	<b>63</b>
7.1	Temps de resposta . . . . .	63
7.1.1	Transaccions blockchain . . . . .	63
7.1.2	Consulta de dades . . . . .	65
7.2	Possibles optimitzacions . . . . .	66
<b>8</b>	<b>Conclusions</b>	<b>68</b>
8.1	Resultat final . . . . .	68
8.1.1	Solució desenvolupada . . . . .	68
8.1.2	Punts forts del sistema . . . . .	69
8.1.3	Punts febles del sistema . . . . .	70
8.2	Propostes de futur . . . . .	71
8.3	Integració de coneixements . . . . .	72
8.3.1	Coneixements aplicats d'altres assignatures . . . . .	72
8.3.2	Grau de satisfacció de les competències tècniques . . . . .	73
8.3.3	Aprenentatges obtinguts . . . . .	76
<b>9</b>	<b>Desviaments i imprevistos</b>	<b>77</b>
9.1	Variacions respecte la planificació inicial . . . . .	77
9.2	Variacions respecte el pressupost inicial . . . . .	78
<b>10</b>	<b>Annex</b>	<b>79</b>
<b>11</b>	<b>Referències</b>	<b>83</b>
11.1	Bibliografia . . . . .	83
11.2	Webgrafia . . . . .	83

## Índex de figures

1	Representació gràfica de la relació entre blocs . . . . .	9
2	Representació gràfica d'un sprint en la metodologia Scrum . . . . .	14
3	Tauler de gestió de Azure DevOps . . . . .	16
4	Diagrama de Gantt . . . . .	27
5	Diagrama de l'arquitectura del sistema . . . . .	37
6	Vista principal de Metamask . . . . .	39
7	Vista principal de l'eina Ganache . . . . .	40
8	Model de classes . . . . .	41
9	Vista de confirmació de Metamask per a executar el mètode d'un Smart Contract . . . . .	50
10	Definició d'estructures de dades del Smart Contract . . . . .	51
11	Vista "Manage" de l'aplicació web . . . . .	51
12	Vista "Add product" de l'aplicació web . . . . .	52
13	Exemple d'informació codificada sobre l'execució d'un mètode del contracte . . . . .	53
14	Exemple de decodificació d'un mètode del contracte . . . . .	54
15	Events definits en el contracte . . . . .	55
16	Exemple d'informació codificada sobre els events emesos en un mètode del contracte . . . . .	55
17	Exemple de decodificació dels events emesos en un mètode del contracte . . . . .	56
18	Exemple de definició del mètode <i>AddProduct</i> en el fitxer ABI . . . . .	58
19	Exemple d'estructura i contingut d'una taula de la base de dades . . . . .	59
20	Vista "Explore" de l'aplicació web . . . . .	59
21	Vista "Product quantity" de l'aplicació web . . . . .	60
22	Vista "Transactions by price" de l'aplicació web . . . . .	61
23	Vista "quantities", definida per a facilitar el càlcul de quantitats . . . . .	62
24	Query SQL que s'aplica sobre la vista "quantities" . . . . .	62
25	Nombre de transaccions en la xarxa Ethereum (mensual) . . . . .	64
26	Nombre d'adreces actives en la xarxa Ethereum (mensual) . . . . .	65
27	Paràmetres de configuració de la xarxa Rinkeby . . . . .	79
28	Script de desplegament de contractes . . . . .	80
29	<i>Output</i> d'èxit del <i>script</i> de desplegament d'un contracte . . . . .	80
30	Temps de resposta obtinguts en la blockchain . . . . .	81
31	Temps de resposta obtinguts en la base de dades . . . . .	81
32	Temps de resposta obtinguts en el <i>server</i> . . . . .	82

## Índex de taules

1	Estimació de tasques . . . . .	25
2	Riscos i mitigacions . . . . .	28
3	Costos materials . . . . .	29
4	Salari dels rols implicats en el projecte . . . . .	30

5	Costos humans . . . . .	31
6	Costos genèrics . . . . .	32
7	Cost dels riscos identificats . . . . .	33
8	Pressupost final . . . . .	33
9	Cas d'ús "Afegir producte" . . . . .	42
10	Cas d'ús "Comprar producte" . . . . .	42
11	Cas d'ús "Reposar estoc" . . . . .	43
12	Cas d'ús "Eliminar producte" . . . . .	43
13	Cas d'ús "Modificar preu d'un producte" . . . . .	44
14	Cas d'ús "Consultar quantitat d'un producte en una data" . . . . .	44
15	Cas d'ús "Consultar preu d'un producte en una data" . . . . .	45
16	Cas d'ús "Consultar productes afegits segons quantitat" . . . . .	45
17	Cas d'ús "Consultar productes venuts segons quantitat" . . . . .	46
18	Cas d'ús "Consultar productes reposats segons quantitat" . . . . .	46
19	Cas d'ús "Consultar productes afegits segons preu" . . . . .	47
20	Cas d'ús "Consultar productes venuts segons preu" . . . . .	47
21	Cas d'ús "Consultar productes reposats segons preu" . . . . .	48
22	Temps de resposta de les transaccions blockchain . . . . .	63
23	Temps de resposta de les consultes de dades . . . . .	66
24	Satisfacció de competències tècniques . . . . .	76
25	Diferències de costos temporals causats pels canvis de planificació . . . . .	77

# 1 Introducció

## 1.1 Context

El projecte "Implementació de Models Conceptuals Deductius amb Blockchain i BD's relacionals" correspon al meu Treball de Fi de Grau per a concloure els meus estudis en Enginyeria Informàtica, impartit a la Universitat Politècnica de Catalunya. Aquest treball pertany a l'especialització d'Enginyeria del Software, de manera que pretén abordar un seguit de conceptes treballats en aquesta menció. L'objectiu del projecte és explorar l'ús de tecnologies Blockchain i bases de dades relacionals com a plataformes per a definir, implementar i utilitzar sistemes d'informació especificats com a Models Conceptuals Deductius, segons les propostes publicades pel professor Antoni Olivé, procurant fer servir benchmarks comparatius per a analitzar el rendiment del sistema.

### 1.1.1 Conceptes

Els conceptes principals que tracta aquest estudi són:

**Models conceptuals:** En els models conceptuals convencionals, es defineixen composicions de conceptes de manera que s'inclouen les entitats importants i les relacions entre aquestes. Les classes definides tenen com a objectiu emmagatzemar un tipus concret d'informació, seguint el format definit en aquella classe. Seguint la nomenclatura de l'article de Jaume Sistac i Antoni Olivé [7], aquests s'anomenen IB (Information Base). L'informació que es disposa d'aquestes entitats en un moment concret és l'Information Base en el moment  $t$ , anomenat també  $IB(t)$ . A més d'aquest component "base" (com el nom indica) es sol definir també un esquema de transicions. Aquest té l'objectiu d'emmagatzemar tots els events que causen la transició de dades d'una taula a una altra.

El problema principal d'aquest enfocament és que obliga al dissenyador del model a fer una decisió primerenca respecte els continguts del IB (la qual s'assumeix que correspon a la informació que s'emmagatzema a la base de dades). Els factors que influeixen en la decisió de la informació que s'hauria d'emmagatzemar depenen de la implementació, fet que dificulta fer-ho durant l'etapa del disseny.

**Models conceptuals deductius:** Aquest concepte va ser introduït per Antoni Olivé, i el principal article en què desenvolupa aquesta proposta és "On the design and implementation of information systems from deductive conceptual models" [4], publicat l'any 1989. Aquest exposa que el principal objectiu d'aquesta proposta és solucionar el problema que presenten els models conceptuals convencionals, tal i com s'explica en l'apartat anterior, delegant així tota la responsabilitat a la capa de lògica i menys a la capa de control. Els models conceptuals deductius es basen en l'importància d'emmagatzemar events o transaccions, en lloc d'objectes. Degut a que

durant l'etapa de disseny d'un model conceptual no podem saber quin tipus de consultes es faran sobre el model, es decideixen emmagatzemar tots aquells events que succeeixin en el sistema. Per exemple, en un model que tingui com a entitat principal el concepte "empleats", es decideix guardar tots els events relacionats: altes, baixes, suspensions, trasllats, etc. D'aquesta manera, quan es vulgui consultar qualsevol informació respecte aquesta entitat, es podran deduir els resultats que desitjem a través d'operacions d'interpretació i classificació.

El fet d'emmagatzemar events atribueix un paper fonamental al "temps", és a dir, al moment d'ocurrència d'aquests fets. Tota la informació (i) té un instant associat  $T(i)$ , que indica l'instant d'ocurrència o de coneixement d'aquella informació. També juguen un paper molt important les restriccions, les quals són les "encarregades" de filtrar i classificar les dades per obtenir els resultats indicats. Al guardar "transaccions" d'una forma tant genèrica, perdem organització de la informació, de manera que necessitem un conjunt de regles o restriccions que filtrin aquesta informació segons les dades que es volen obtenir.

**Bases de dades relacionals:** Una base de dades relacional és un conjunt de taules i relacions entre aquestes, formades per files i columnes. Les files corresponen als registres, i les columnes als atributs que contenen aquests registres. Cada registre, a més d'un identificador únic, té normalment un valor per cada atribut, de manera que es simplifica la relació entre les dades. Això proporciona un accés fàcil i intuïtiu a la informació emmagatzemada.

Un concepte clau d'aquesta tecnologia són les vistes, les quals són molt importants per a aquest projecte. Una vista és un conjunt de resultats d'una consulta emmagatzemada en les dades. És una consulta, però que es presenta com una taula, virtual, a partir d'un conjunt de la resta de taules de la base de dades. A més, presenta la mateixa estructura que una taula convencionals: files i columnes.

**Blockchain:** La blockchain és, fonamentalment, una base de dades distribuïda, que està compartida pels nodes d'una xarxa. Això garanteix la fidelitat i la seguretat del registre de dades sense la necessitat de tercers. La principal diferència que presenta la blockchain respecte una base de dades convencional és la forma en què s'estructuren les dades. La informació s'emmagatzema en blocs, els quals contenen conjunts de dades. Els blocs tenen una capacitat limitada, de manera que quan un bloc està ple, es "tanca" i es relaciona amb el bloc anterior. D'aquesta manera es forma una cadena de blocs (blockchain), i s'aconsegueix un registre precís de tots els esdeveniments amb el timestamp en què ha succeït cadascun.



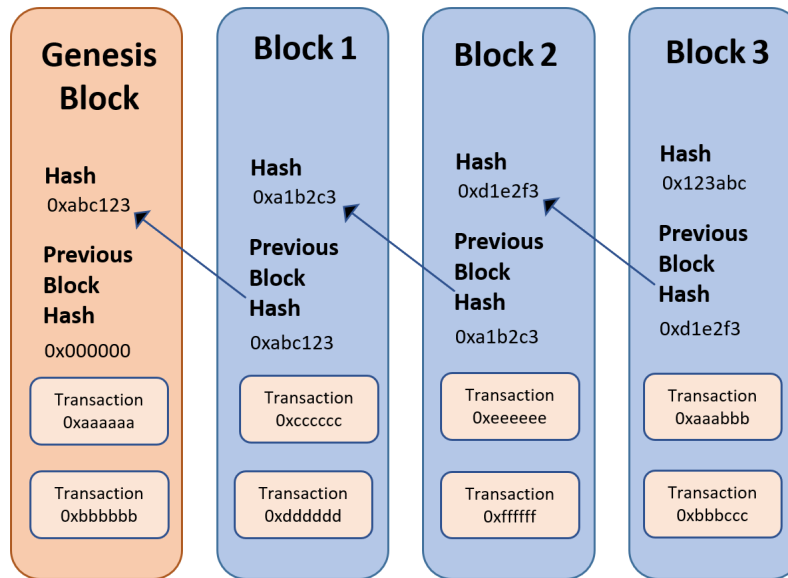


Figura 1: Representació gràfica de la relació entre blocs

### 1.1.2 Problema a resoldre

Tal i com s'explica en l'apartat dels models conceptuals, aquests presenten un problema de disseny. S'han de prendre un gran nombre de decisions respecte el model conceptual que es vol utilitzar abans de conèixer les necessitats que es tindran en el processament de la informació. No és fins que es posa en pràctica un sistema que es poden definir amb claredat quin tipus d'informació es requereix. En la majoria de casos, això sol comportar haver d'aplicar canvis respecte el model de dades inicial, perquè la seva estructura o les dades que es persisteixen no satisfan les necessitats de l'organització. Per exemple, utilitzant un model conceptual convencional per a emmagatzemar els empleats d'una empresa, aquest disposarà només de l'estat actual del sistema, és a dir, els empleats actuals. Només es pot consultar informació referent a events passats si aquesta informació ha sigut emmagatzemada, com podrien ser els empleats que ja no treballen a l'empresa. Aquestes necessitats solen aparèixer un cop es posa en funcionament el sistema, de manera que es perd aquesta informació si no s'ha implementat una funcionalitat que les enregistri. Aquest projecte pretén utilitzar la blockchain per tal d'emmagatzemar transaccions, de manera que sempre es disposarà d'un registre de tots els events que han succeït.

En l'actualitat, cada vegada són més les empreses que utilitzen l'anàlisi de dades, també anomenat com "Business Intelligence". Aquest terme fa referència a l'ús d'estratègies i eines per transformar la informació en coneixement, amb l'objectiu de millorar els processos i les preses de decisions de l'empresa. Així

doncs, s'intensifica la necessitat de crear un sistema d'informació que millori els processos d'anàlisi de dades, per tal d'evitar la modificació del model de dades una vegada ja està dissenyat, i tot el que això comporta si el sistema es troba en una etapa avançada de desenvolupament.

Considerant les característiques que ofereix la blockchain i els problemes que suggeria Antoni Olivé en la seva proposta, aquesta tecnologia podria ser una solució per a implementar els "Models Conceptuals Deductius". Així doncs, l'estudi consisteix en desenvolupar un sistema d'informació utilitzant blockchain i bases de dades relacionals, per tal de simular un model conceptual deductiu.

### 1.1.3 Stakeholders

Els actors implicats en el projecte, ja sigui de manera directa o indirecta, són:

- Desenvolupador: El projecte serà desenvolupat íntegrament per l'Aniol Carbó, l'autor d'aquest Treball de Fi de Grau.
- Director del TFG: el professor que supervisa aquest projecte és en Joan Antoni Pastor, qui té un ampli coneixement de l'àmbit dels sistemes d'informació i coneix en detall l'objecte d'estudi.
- Antoni Olivé: és l'autor de la proposta "Models Conceptuals Deductius", de manera que aquesta implementació l'involucra en el projecte.
- Investigadors de recerca en Information Systems i Software Engineering: degut a que aquest projecte és de caire exploratori, els investigadors dels àmbits que es tracten en el treball es poden considerar stakeholders.

## 1.2 Justificació

Així doncs, aquest projecte es basa en utilitzar el concepte teòric dels "Models Relacionals Deductius" i crear un nou sistema d'informació utilitzant la blockchain i les bases de dades relacionals.

L'objectiu d'aquest nou sistema és utilitzar la blockchain, que ofereix un registre de transaccions, juntament amb les bases de dades relacionals i les vistes, que permeten la classificació de la informació en funció de les nostres necessitats. Tal i com s'explica en la secció dels models conceptuals deductius, aquests es basen en emmagatzemar events, de manera que es pot tenir un control molt més precís de la informació i del seu recorregut. Això és una característica que presenta molt similarment la tecnologia blockchain, ja que aquesta consisteix en el registre de transaccions de manera segura i descentralitzada.

Pel que fa al processament de la informació, les vistes poden oferir una solució molt eficient per filtrar les dades desitjades. Cal tenir en compte que el registre de la informació en la blockchain significa disposar de les dades de manera, la qual només està indexada per l'identificador (hash) de la transacció. És per això que una vista, ja sigui materialitzada o no, pot aportar la possibilitat de

filtrar i processar la informació emmagatzemada. Tal i com es menciona durant el treball, aquest concepte va ser introduït per un professor de la FIB, Antoni Olivé. És per això que els projectes teòrics o de recerca existents sobre aquest tema són tots procedents de la FIB, tots ells fruit d'un desenvolupament de tesi amb la col·laboració d'Antoni Olivé. Les tesis trobades sobre aquest concepte són:

- Sistac Planas, J. (1987). Construcció automàtica de prototipus de sistemes d'informació a partir de models conceptuals deductius amb el llenguatge DADES. Universitat Politècnica de Catalunya.
- Costal, D. (1992). A Method for Reasoning about Deductive Conceptual Models of Information Systems. Universitat Politècnica de Catalunya.
- Quer, C. (1993). Object Interaction Object-Oriented Deductive Conceptual Models. Universitat Politècnica de Catalunya.

Aquestes tres tesis tracten sobre els models conceptuals deductius, i contenen enfocaments tant de caire teòric com pràctic. Tot i així, les semblances amb aquest projecte són escasses, ja que en aquest hi juguen un gran paper tecnologies com la blockchain i les bases de dades relacionals, els quals no apareixen en les tesis trobades. El que sí que es podran utilitzar són alguns dels exemples que s'hi mencionen, sempre citant-los adequadament en el projecte.

L'objectiu final d'aquest projecte és, doncs, dissenyar un sistema d'informació basat en els models conceptuals deductius i que combina aquestes dues tecnologies per tal d'aprofitar els beneficis que aporta cadascuna.

## 1.3 Abast

### 1.3.1 Objectius

Pel que fa a la part tècnica del projecte, els objectius principals són:

1. Comprendre amb profunditat el concepte dels "Models Conceptuals Deductius" per tal d'identificar tots els beneficis que aporta respecte un model conceptual convencional, i així dissenyar un sistema d'informació que aporta un benefici real.
2. Aprofitar les capacitats que ofereix la blockchain per tal d'aplicar-ho als models conceptuals deductius, una proposta que va néixer abans que la blockchain i, per tant, no es comptava amb una tecnologia d'aquestes característiques.
3. Dissenyar un sistema d'informació funcional que combini la blockchain i les BD's relacionals per tal d'aprofitar els beneficis que aporta cadascuna, i aportar una nou enfocament respecte els models conceptuals deductius.

Altres sub objectius del projecte, d'un caire més personal, són:

1. L'exploració profunda de la tecnologia blockchain per tal d'aprofitar-ne tot el seu potencial. Al ser una tecnologia tan recent, encara té molt recorregut per davant i un ampli ventall de possibilitats on pot ser aplicada.
2. L'optimització de la part referent a les bases de dades per tal d'obtenir els mínims temps de resposta. Haver cursat l'especialitat d'enginyeria del software em permet poder indagar en aquest aspecte del projecte, i centrar-me en detalls tècnics que puguin millorar el rendiment del sistema.

### **1.3.2 Competències tècniques**

Aquest projecte pertany a l'especialitat d'Enginyeria del Software, i les principals competències tècniques escollides per a tractar en profunditat en aquest TFG són:

- CES1.2: Donar solució a problemes d'integració en funció de les estratègies, dels estàndards i de les tecnologies disponibles.
- CES1.5: Especificar, dissenyar, implementar i evaluar bases de dades.
- CES1.7: Controlar la qualitat i dissenyar proves en la producció de software.

### **1.3.3 Obstacles i riscos**

Els obstacles que dificulten la realització del projecte o que l'aparten d'un desenvolupament ideal són:

1. Limitacions de temps: tot i ser un projecte en el que m'agradaria dedicar-hi el màxim de temps possible, només disposo d'un quadrimestre per a desenvolupar-lo per complet. Fer una bona planificació serà un aspecte crucial per a mitigar aquest risc.
2. Desconeixença del tema tractat: tot i que el projecte tracta conceptes de l'enginyeria del software, n'hi ha dos els quals no en tinc un ampli coneixement. El primer és el blockchain, ja que tot i el meu interès personal per aquesta tecnologia, no en tinc molta experiència. El segon són els "Models Conceptuals Deductius", una proposta portada a la FIB a mans d'Antoni Olivé, però que no es tracta en cap assignatura del grau.
3. Documentació limitada: respecte els "Models Conceptuals Deductius", és un concepte teòric poc conegut, de manera que pot ser complicat trobar-ne informació a Internet. Tot i així, la UPC disposa de varis articles i tesis que tracten sobre aquest concepte.

### 1.3.4 Requisits funcionals

Els requisits funcionals principals del sistema són:

- Inserció de registres: l'usuari ha de poder, en tot moment, crear nous registres i que aquests s'emmagatzemin correctament en el sistema.
- Esborrat de registres: l'usuari ha de poder, en tot moment, esborrar registres existents i que cadascuna d'aquestes accions quedi persistent al sistema.
- Modificació de registres: l'usuari ha de poder, en tot moment, modificar registres existents i que cadascuna d'aquestes accions quedi persistent al sistema.
- Consulta de dades: l'usuari ha de poder, en tot moment, consultar les dades emmagatzemades en els sistema, les quals han de ser presentades de forma clara i entenedora. A més, l'usuari ha de poder aplicar filtres sobre aquesta consulta, com poden ser paràmetres de temps (a partir d'una data, fins a una data, etc) o d'estat (registres eliminats, registres modificats, etc).

### 1.3.5 Requisits no funcionals

Els requisits no funcionals que s'han d'oferir en el sistema són:

- Rendiment: el sistema ha de garantir que es poden processar volums de dades suficientment grans com per poder demostrar la seva utilitat. A més, ha de poder oferir les seves funcionalitats amb temps de resposta prou ràpids com perquè l'ús del sistema sigui útil i còmode.
- Disponibilitat: els usuaris del sistema han de poder accedir a totes les funcionalitats en qualsevol moment.
- Integritat de dades: degut a que el producte a desenvolupar és un sistema d'informació, la integritat de dades és crucial. S'ha de controlar de forma estricta quina és la informació que es permet introduir als usuaris per evitar la inserció de dades incorrectes. A més, s'han d'evitar situacions de corrupció o pèrdua de dades, ja que una d'aquestes situacions pot significar un gran problema per al sistema i tots els seus usuaris.
- Compatibilitat: un dels objectius del projecte és desenvolupar un sistema d'informació que sigui útil i funcional en un cas real, i això implica que sigui compatible amb altres sistemes i infraestructures. L'adaptació a utilitzar aquest sistema ha de ser el més còmode possible, és per això que s'ha d'oferir una bona compatibilitat amb la resta d'elements de la infraestructura on es vulgui incorporar.

- Robustesa: davant d'un esdeveniment "anormal" en el sistema, s'ha de garantir que el sistema segueix funcionant. És a dir, que davant d'un error en algun dels components del sistema, s'ha d'evitar que això suposi un error en tot el sistema per complet i ha d'intentar oferir el màxim de funcionalitat possible a l'usuari.

## 1.4 Metodologia i rigor

Degut al temps limitat del qual es disposa per a realitzar projecte, la gestió del temps i de les tasques ha de ser la més òptima possible. Tot i no ser un projecte en equip, es desenvoluparà utilitzant una metodologia Agile, la qual dóna flexibilitat i agilitat al procés de desenvolupament perquè permeten una detecció ràpida de possibles riscos i obstacles. La metodologia escollida és Scrum, degut a que hi he treballat anteriorment en entorns de treball reals i estic acostumat a seguir aquest mètode.

El nucli principal d'aquesta metodologia de treball és l'sprint. Es tracta de dividir la feina del projecte en diferents blocs, els quals han de ser funcionals, i completar-los en la durada establerta per l'sprint. D'aquesta manera, s'aconsegueix un increment de valor en el producte que s'està construint i, a més, es pot valorar el treball assolit fins al moment ja que el producte és funcional i pot rebre *feedback* dels stakeholders.

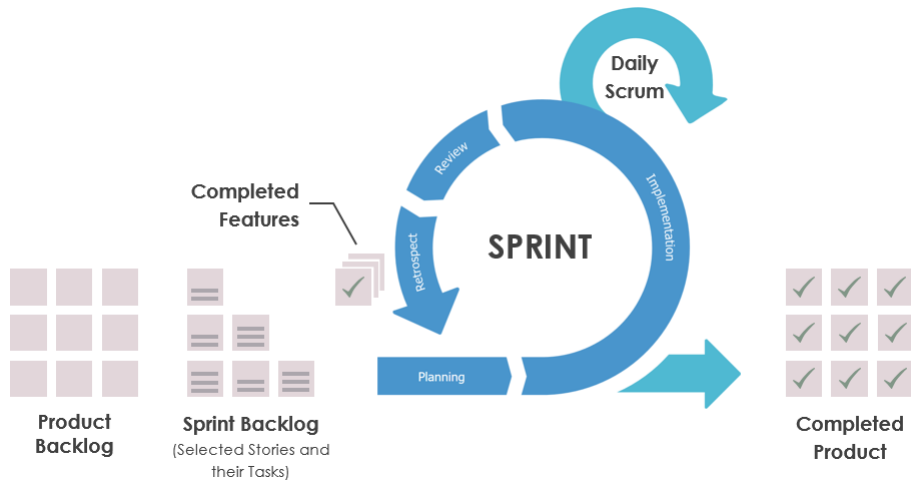


Figura 2: Representació gràfica d'un sprint en la metodologia Scrum

En aquest cas, s'han establert reunions mensuals amb el director del projecte, en Joan Antoni, de manera que aquests seran considerats els "deadlines" dels sprints d'aquest projecte. En aquestes reunions, es realitzarà la retrospectiva de

l'sprint realitzat, i la planificació del següent. Tot i així, si durant el desenvolupament del projecte veiem que la durada dels sprints és massa llarga, establirem més reunions, per exemple de forma bimensual. Pel que fa a les altres reunions que s'haurien de realitzar al llarg de l'sprint, com poden ser les "Daily standups" o les "Revisions d'sprint", aquestes reunions tenen com a finalitat compartir i conèixer el progrés de cada membre de l'equip del projecte. Aquest no és un treball en equip, sinó que hi treballa només l'autor amb la cooperació del director, de manera que s'eliminen les reunions de *feedback* tant recurrent. Tot i així, qualsevol reunió que es realitzi amb el director de forma paral·lela a les reunions previstes, es pot considerar una "Revisió d'sprint" i es pot utilitzar com a tal. Un dels elements que també es veuen modificats respecte l'Scrum convencional són els rols. En aquest cas, l'autor del projecte, l'Aniol Carbó, rep dos rols:

1. "Scrum Master": gestiona la divisió del treball i és el responsable de que s'assoleixin les tasques dins els períodes establerts. S'ocupa també del tauler Scrum, on es defineixen totes les històries d'usuari en forma de targetes.
2. Equip de desenvolupament: realitza el treball necessari en cada sprint per a poder obtenir un increment de valor i acabar amb un producte funcional.

Per l'altra banda, el director del projecte, Joan Antoni Pastor, es pot considerar el "Product owner", ja que la seva mirada estarà sempre posada en els resultats que s'han d'obtenir i serà l'encarregat de marcar els objectius de forma clara i acordada amb l'autor del treball. Pel que fa a la resta d'elements d'Scrum, com són les històries d'usuari, les tasques, els backlogs o la "definition of done", es realitzaran de manera estàndard.

L'eina escollida per a la gestió del projecte és Azure DevOps (veure Figura 3), ja que conté un conjunt de serveis que serveixen tant per al desenvolupament com per la gestió de tasques. Al tenir aquests dos components en la mateixa plataforma, es poden relacionar les històries d'usuari i les tasques dels taulers amb els canvis de codi corresponents, juntament amb la gestió de repositoris utilitzant Git. També disposa de pipelines per a poder aplicar tècniques de CI/CD.

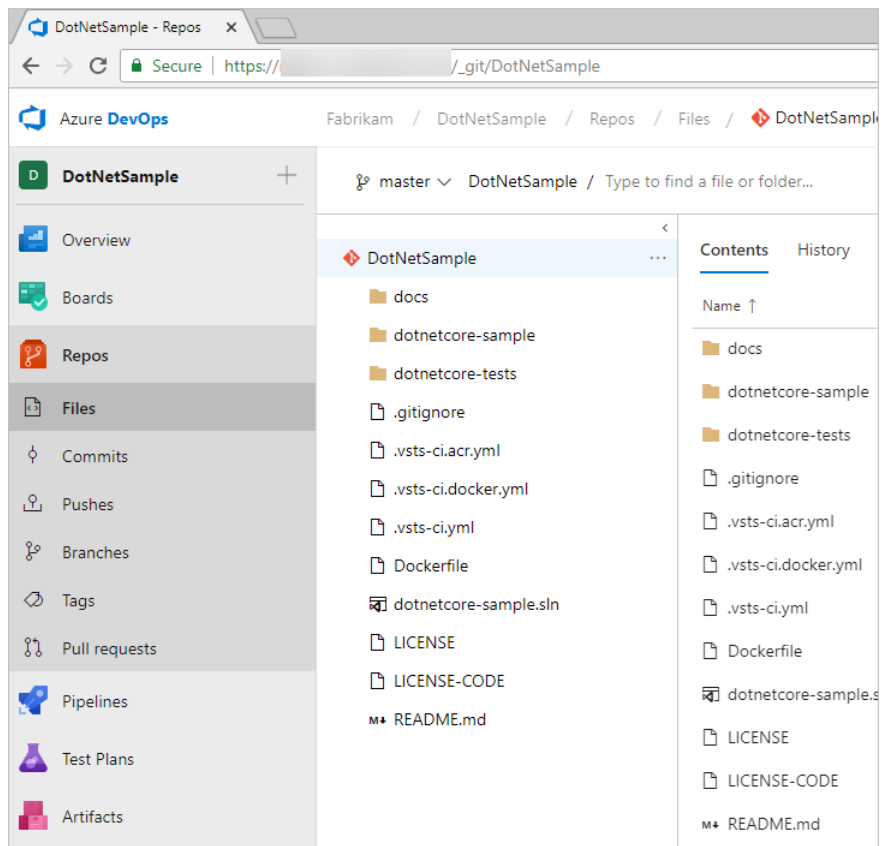


Figura 3: Tauler de gestió de Azure DevOps

Les eines descrites apareixen en aquesta imatge d'exemple, on a la part esquerra podem veure els apartats de: "Boards", que correspon als taulers de gestió de tasques; "Repos", que correspon a la gestió de repositoris i "Pipelines", que correspon a les eines d'integració contínua. Disposar de totes aquestes eines unificades en una mateixa plataforma és el motiu pel qual he decidit utilitzar Azure DevOps.



## 2 Planificació

Tal i com s'explica en l'apartat de metodologia i rigor, el projecte es desenvolupa utilitzant la metodologia Agile. Aquesta considera que, abans dels sprints, on es realitza el desenvolupament del projecte, hi ha d'haver una etapa d'incepció. Aquesta etapa té com a objectiu identificar els objectius i els riscos del projecte, la planificació temporal, la creació del tauler Agile i les històries d'usuari, etc. És per això que, des del dia 21 de febrer fins al 21 de març, es considera que el projecte es troba en l'etapa d'incepció. Pel que fa als sprints, on sí que es desenvolupa un producte funcional, es delimiten les següents dates:

- Sprint 1: del 22 de març al 22 d'abril
- Sprint 2: del 23 d'abril al 23 de maig
- Sprint 3: del 24 de maig al 24 de juny

D'aquesta manera, el projecte està dividit en 4 etapes, totes amb una durada d'un mes. Cada sprint té un conjunt de tasques assignades, de manera que s'espera tenir-les acabades en la data final de l'sprint. A més, s'ha de realitzar també la redacció de documentació corresponent a la feina feta durant l'sprint. Al final de cada etapa, es fa una reunió amb el director del treball i, a més, es fa una retrospectiva per tal de valorar si totes les tasques han estat completades o si cal modificar la planificació inicial degut a possibles contratemps. La data prevista d'entrega és el 27 de juny, per tant el més probable és que sobrin varis dies entre la finalització del tercer sprint i l'entrega del projecte, que serviran per a preparar la memòria final. El total d'hores de dedicació previstes és aproximadament 500, tot i que la xifra exacta es detalla en l'apartat d'estimació de tasques.

### 2.1 Descripció de tasques

Per tal de definir les tasques de forma detallada, aquestes es divideixen en blocs segons l'etapa en la que està previst que siguin realitzades. A més, es classifiquen també segons el tipus de feina que comporten:

- Gestió de projecte (GP): engloba tots els elements de gestió, tant la preparació prèvia del projecte com es activitats de la metodologia Agile.
- Desenvolupament (DEV): tota la feina corresponent al desenvolupament del projecte, principalment tasques de disseny i de programació.
- Documentació (DOC): referent a les tasques de redacció.
- Construcció (CN): engloba tant les tasques d'instal·lació d'eines i softwares necessaris com les tasques de desplegament d'aplicacions o serveis.
- Recerca (RC): investigació, lectura i comprensió en detall dels conceptes tractats en aquest projecte.

Es consideren també els tres rols mencionats a l'apartat de metodologia: "Scrum Master", "Equip de desenvolupament" i "Product Owner".

### 2.1.1 Gestió de projecte

- GP1: Definició de l'abast i contextualització
  - Descripció: Es contextualitza el treball i es descriuen els objectius d'aquest, delimitant així l'abast del projecte. Es defineixen també possibles riscos i s'explica la metodologia que s'utilitzarà per a l'organització i el desenvolupament.
  - Duració: 20h
  - Dependències:
  - Recursos humans: Scrum Master
  - Recursos materials: Ordinador, connexió a Internet, MikTex, Visual Studio Code
- GP2: Planificació temporal
  - Descripció: S'explica la delimitació de les etapes del projecte, les tasques a realitzar i el temps estimat de cadascuna. També es defineix com s'actua davant de possibles riscos.
  - Duració: 20h
  - Dependències: GP1
  - Recursos humans: Scrum Master
  - Recursos materials: Ordinador, connexió a Internet, MikTex, Visual Studio Code
- GP3: Gestió econòmica i sostenibilitat
  - Descripció: Es calculen els costos materials, humans, de riscos i de contingències que suposa el projecte. També s'analitza l'impacte del desenvolupament del sistema a nivell social, econòmic i mediambiental.
  - Duració: 20h
  - Dependències: GP2
  - Recursos humans: Scrum Master
  - Recursos materials: Ordinador, connexió a Internet, MikTex, Visual Studio Code
- GP4: Revisió de la documentació
  - Descripció: Crear un document unificat amb tota la documentació elaborada fins el moment, per tal d'aconseguir una definició de projecte completa.

- Duració: 5h
- Dependències: GP3
- Recursos humans: Scrum Master
- Recursos materials: Ordinador, connexió a Internet, MikTex, Visual Studio Code
- GP5: Preparació del tauler Agile
  - Descripció: Posar en funcionament un tauler Agile i crear totes targetes corresponents a les històries d'usuari. D'aquesta manera està preparat per l'inici del primer sprint, on ja es porten a terme tasques de desenvolupament.
  - Duració: 10h
  - Dependències: GP4
  - Recursos humans: Scrum Master
  - Recursos materials: Ordinador, connexió a Internet, Azure DevOps

### 2.1.2 Sprint 1

- RC1: Investigació i recerca
  - Descripció: Indagar en els conceptes descrits en els articles i tesis obtinguts per tal de definir amb claredat la relació entre aquest projecte i els models conceptuals deductius.
  - Duració: 25h
  - Dependències:
  - Recursos humans: Equip de desenvolupament
  - Recursos materials: Ordinador, articles/tesis necessaris
- RC2: Preparació d'exemples
  - Descripció: Es defineixen els exemples que s'utilitzaran en els sistema per a poder demostrar-ne el funcionament i la seva utilitat davant d'un cas real.
  - Duració: 10h
  - Dependències: RC1
  - Recursos humans: Equip de desenvolupament
  - Recursos materials: Ordinador, articles/tesis necessaris
- DEV1: Disseny del model de dades
  - Descripció: Dissenyar el model de dades utilitzat en el sistema d'informació a desenvolupar.

- Duració: 15h
- Dependències: RC2
- Recursos humans: Equip de desenvolupament
- Recursos materials: Ordinador, connexió a Internet, eina de notació de models
- DEV2: Disseny de les consultes de dades
  - Descripció: Pensar possibles casos d'ús referents al model de dades dissenyat per tal de definir les consultes que es podran aplicar sobre les dades del sistema.
  - Duració: 15h
  - Dependències: DEV1
  - Recursos humans: Equip de desenvolupament
  - Recursos materials: Ordinador, connexió a Internet, eina de notació de models
- CN1: Instal·lació de software
  - Descripció: Instal·lar totes les eines i softwares necessaris per al desenvolupament del projecte.
  - Duració: 10h
  - Dependències:
  - Recursos humans: Equip de desenvolupament
  - Recursos materials: Ordinador, connexió a Internet
- DEV3: Desenvolupament dels Smart Contracts
  - Descripció: Desenvolupar el codi dels Smarts Contracts per al registre de les operacions i transaccions en la xarxa blockchain.
  - Duració: 35h
  - Dependències: DEV2
  - Recursos humans: Equip de desenvolupament
  - Recursos materials: Ordinador, connexió a Internet, Ganache
- DOC1: Documentació de l'sprint 1
  - Descripció: Redactar tota la documentació corresponent a la feina realitzada durant l'sprint 1.
  - Duració: 15h
  - Dependències:
  - Recursos humans: Scrum Master

- Recursos materials: Ordinador, connexió a Internet, MikTex, Visual Studio Code
- GP6: Retrospectiva de l'sprint 1
  - Descripció: Evaluar si s'han realitzat les tasques programades per a l'sprint i fer els ajustaments necessaris si hi han desviacions.
  - Duració: 5h
  - Dependències:
  - Recursos humans: Scrum Master, Product owner
  - Recursos materials: Ordinador, connexió a Internet, Azure DevOps

### 2.1.3 Sprint 2

- DEV4: Desenvolupament de l'aplicació web
  - Descripció: Desenvolupar una aplicació web per a simular l'ús d'una interfície que permet a l'usuari la gestió de les dades.
  - Duració: 35h
  - Dependències:
  - Recursos humans: Equip de desenvolupament
  - Recursos materials: Ordinador, connexió a Internet, Webstorm IDE
- DEV5: Interacció amb els Smart Contracts
  - Descripció: Programar la interacció/comunicació amb l'Smart Contract i les seves funcions.
  - Duració: 30h
  - Dependències: DEV3, DEV4
  - Recursos humans: Equip de desenvolupament
  - Recursos materials: Ordinador, connexió a Internet, Ganache
- CN2: Desplegament dels Smart Contracts
  - Descripció: Desplegar els Smart Contracts desenvolupats en una xarxa blockchain de proves "testnet".
  - Duració: 20h
  - Dependències: DEV5
  - Recursos humans: Equip de desenvolupament
  - Recursos materials: Ordinador, connexió a Internet, Ganache
- DEV6: Obtenció de dades de la xarxa blockchain
  - Descripció: Establir una connexió amb la xarxa blockchain i obtenir-ne les dades a demanda.

- Duració: 25h
- Dependències: CN2
- Recursos humans: Equip de desenvolupament
- Recursos materials: Ordinador, connexió a Internet, Webstorm IDE
- DOC2: Documentació de l'sprint 2
  - Descripció: Redactar tota la documentació corresponent a la feina realitzada durant l'sprint 2.
  - Duració: 15h
  - Dependències:
  - Recursos humans: Scrum Master
  - Recursos materials: Ordinador, connexió a Internet, MikTex, Visual Studio Code
- GP7: Retrospectiva de l'sprint 2
  - Descripció: Evaluar si s'han realitzat les tasques programades per a l'sprint i fer els ajustaments necessaris si hi han desviacions.
  - Duració: 5h
  - Dependències:
  - Recursos humans: Scrum Master, Product owner
  - Recursos materials: Ordinador, connexió a Internet, Azure DevOps

#### 2.1.4 Sprint 3

- DEV7: Disseny i creació de les taules
  - Descripció: Dissenyar i crear les taules de la base de dades relacional utilitzada per a emmagatzemar dades del sistema.
  - Duració: 30h
  - Dependències:
  - Recursos humans: Equip de desenvolupament
  - Recursos materials: Ordinador, connexió a Internet, SGBD
- DEV8: Desenvolupament de les vistes
  - Descripció: Desenvolupar la part corresponent a les vistes utilitzant bases de dades relacionals.
  - Duració: 25h
  - Dependències: DEV7
  - Recursos humans: Equip de desenvolupament
  - Recursos materials: Ordinador, connexió a Internet, SGBD

- CN3: Desplegament de la base de dades
  - Descripció: Dissenyar i crear les taules de la base de dades relacional utilitzada per a emmagatzemar dades del sistema.
  - Duració: 10h
  - Dependències: DEV7
  - Recursos humans: Equip de desenvolupament
  - Recursos materials: Ordinador, connexió a Internet, SGBD
- DEV9: Interacció amb la base de dades
  - Descripció: Establir una connexió amb la base de dades i desenvolupar per les crides per a poder obtenir informació a demanda.
  - Duració: 25h
  - Dependències: CN3, DEV8
  - Recursos humans: Equip de desenvolupament
  - Recursos materials: Ordinador, connexió a Internet, SGBD
- DEV10: Mostrat de dades en l'aplicació web
  - Descripció: Desenvolupar les crides necessàries per tal de mostrar la informació demanada per l'usuari des de l'aplicació web.
  - Duració: 20h
  - Dependències: DEV9
  - Recursos humans: Equip de desenvolupament
  - Recursos materials: Ordinador, connexió a Internet, Webstorm IDE
- DOC3: Documentació de l'sprint 3
  - Descripció: Redactar tota la documentació corresponent a la feina realitzada durant l'sprint 3.
  - Duració: 15h
  - Dependències:
  - Recursos humans: Scrum Master
  - Recursos materials: Ordinador, connexió a Internet, MikTex, Visual Studio Code
- GP8: Retrospectiva de l'sprint 3
  - Descripció: Evaluar si s'han realitzat les tasques programades per a l'sprint i fer els ajustaments necessaris si hi han desviacions.
  - Duració: 5h
  - Dependències:
  - Recursos humans: Scrum Master, Product owner
  - Recursos materials: Ordinador, connexió a Internet, Azure DevOps

### 2.1.5 Memòria final

- DOC4: Finalització de la memòria
  - Descripció: Unificar tota la documentació redactada durant el projecte i consolidar la memòria final.
  - Duració: 20h
  - Dependències:
  - Recursos humans: Scrum Master
  - Recursos materials: Ordinador, connexió a Internet, MikTex, Visual Studio Code

## 2.2 Estimació de tasques

Després de llistar totes les tasques planificades amb la seva respectiva descripció, a la Taula 1 es pot veure, de forma gràfica, la distribució de les tasques i la càrrega de treball associada a cada etapa del projecte. A més, s'han associat colors per a cada etapa del projecte, que després s'utilitzaran per representar-les en el Diagrama de Gantt (Figura 4).



Tasca	Codi	Duració (h)	Dependències
Incepció	-	75	-
Definició de l'abast i contextualització	GP1	20	-
Planificació temporal	GP2	20	GP1
Gestió econòmica i sostenibilitat	GP3	20	GP2
Revisió de la documentació	GP4	5	GP3
Preparació del tauler Agile	GP5	10	GP4
<b>Sprint 1</b>	-	<b>130</b>	-
Investigació i recerca	RC1	25	-
Preparació d'exemples	RC2	10	RC1
Disseny del model de dades	DEV1	15	RC2
Disseny de les consultes de dades	DEV2	15	DEV1
Instal·lació de software	CN1	10	-
Desenvolupament dels Smart Contracts	DEV3	35	DEV2
Documentació de l'sprint 1	DOC1	15	-
Retrospectiva de l'sprint 1	GP6	5	-
<b>Sprint 2</b>	-	<b>130</b>	-
Desenvolupament de l'aplicació web	DEV4	35	-
Interacció amb els Smart Contracts	DEV5	30	DEV3, DEV4
Desplegament dels Smart Contracts	CN2	20	DEV5
Obtenció de dades de la xarxa blockchain	DEV6	25	CN2
Documentació de l'sprint 2	DOC2	15	-
Retrospectiva de l'sprint 2	GP7	5	-
<b>Sprint 3</b>	-	<b>130</b>	-
Disseny i creació de les taules	DEV7	30	-
Desenvolupament de les vistes	DEV8	25	DEV7
Desplegament de la base de dades	CN3	10	DEV7
Interacció amb la base de dades	DEV9	25	CN3, DEV8
Mostrat de dades en l'aplicació web	DEV10	20	DEV9
Documentació de l'sprint 2	DOC3	15	-
Retrospectiva de l'sprint 2	GP8	5	-
<b>Memòria final</b>	-	<b>20</b>	-
Finalització de la memòria	DOC4	20	-
<b>TOTAL</b>	-	<b>485</b>	-

Taula 1: Estimació de tasques

### 2.3 Diagrama de Gantt

A la Figura 4 hi ha representada de forma gràfica la distribució temporal de les tasques dins el període de realització d'aquest treball. Tal i com es menciona a la introducció de la planificació temporal, es considera com a data inicial el 21 de febrer i data final el 27 de juny, i durant aquest període es distribueixen les tasques en quatre etapes. La integració dels sprints en el diagrama és en la

dimensió temporal, tal i com s'explica a [2]. Tot i així, com que les tasques estan ordenades per ordre de realització, es podria considerar que els sprints es troben en la pròpia llista de tasques. És per aquest motiu que l'assignació de tasques en el diagrama és molt semblant a cada sprint, seguint una forma esglaonada.

Cal mencionar que, tot i que en la Taula 1 es faci referència a una cinquena etapa per la memòria final, aquesta no apareix en el Diagrama de Gantt. Això és degut a que la càrrega de treball sol acumular-se al final del projecte, de manera que considerem l'últim sprint fins la data final, i la finalització de la memòria haurà d'estar inclosa en aquesta última etapa.

Pel que fa a la documentació, és una pràctica habitual assignar-la a totes les setmanes de l'sprint, per a poder anar redactant a mesura que es desenvolupa la part tècnica. Tot i així, en aquest cas s'ha assignat només a la última setmana ja que considero més òptim centrar-se en el desenvolupament de les tasques DEV i CN durant l'sprint. Així doncs, la última setmana sol quedar assignada a tasques de gestió (GP) i documentació (DOC), de manera que al documentar tota la feina realitzada, és més pràctic evaluar si s'han realitzat les tasques tal i com s'esperava.

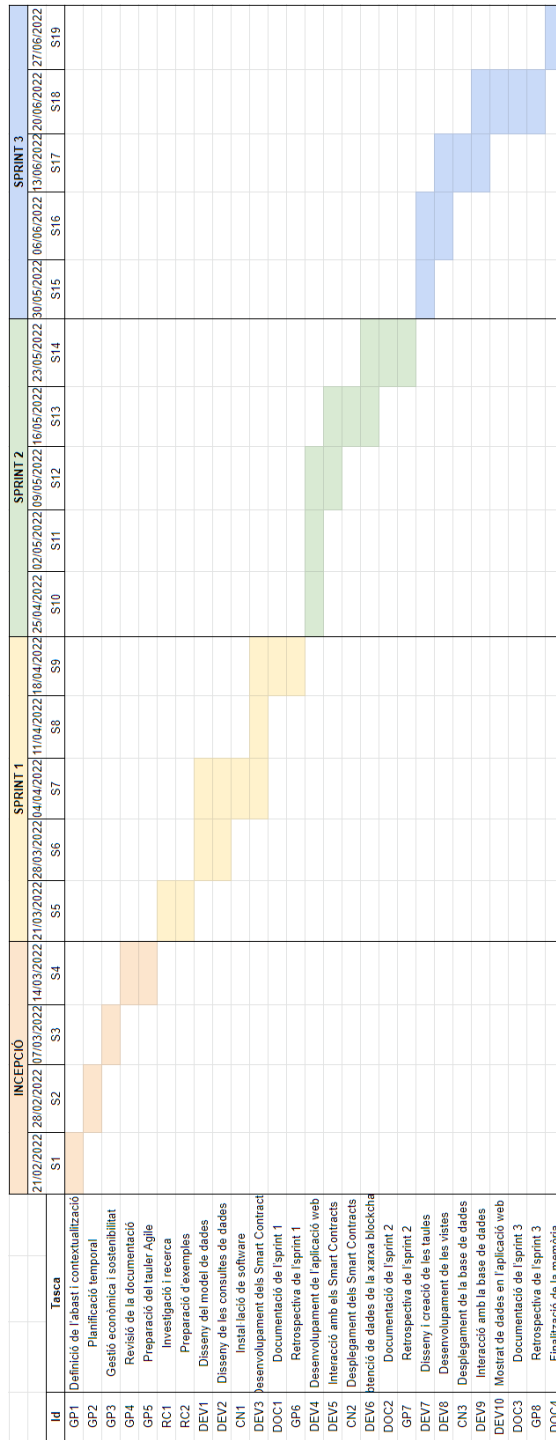


Figura 4: Diagrama de Gantt

## 2.4 Gestió de riscos

S'ha elaborat una taula (Taula 2) on es llisten els possibles riscos que poden aparèixer durant el desenvolupament del projecte, amb l'impacte, la probabilitat i la mitigació corresponents:

Risc	Impacte	Probabilitat	Mitigació
Limitacions de temps	Alt	Alt	Dissenyar una bona planificació i seguir-la tan detalladament com es pugui.
Desconeixença del tema tractat	Mitjà	Baix	Estimació d'hores elevada en aquelles tasques que tractin temes desconeguts.
Documentació escassa	Baix	Baix	Assignar les tasques d'investigació i recerca a l'inici del projecte.

Taula 2: Riscos i mitigacions

Tal i com es mostra en la Taula 2, el primer risc és el més perillós, degut al seu alt impacte i la seva alta probabilitat; el temps del que es disposa per a desenvolupar el projecte, i de forma individual, fa que aquest sigui el major risc. Aquesta limitació de temps afecta a les tasques de desenvolupament, sobretot en aquelles que es concentren al final de la planificació, però que són crucials per obtenir un sistema d'informació funcional. La solució a un possible contratemps en aquestes tasques és l'enfocament a l'estil "MVP" (Minimum Viable Product): es prioritzarà el funcionament correcte del producte encara que l'abast i el detall d'aquest siguin poc profunds. Per exemple, per a desenvolupar i testejar el funcionament del sistema està previst utilitzar un model de dades d'exemple molt senzill, extret de la tesi de Jaume Sistac [6], i més endavant canviar-lo per a un de més complex. Si aparegués un contratemps, aquesta seria una de les tasques que s'eliminarien, degut a la poca importància que té a nivell funcional. Tot i això, la planificació temporal s'ha dissenyat per tal d'assolir totes les tasques desitjades en el temps establert. Si es segueix amb detall, no hauria de suposar un gran problema.

Un fet que podria afectar en el compliment de la planificació és el segon risc de la taula, la desconeixença del tema tractat. El fet d'enfrontar-se a noves tecnologies com la blockchain, i a nous conceptes teòrics com els MCD's, podria resultar en una desviació de la planificació si apareixen problemes en aquests àmbits. Tot i així, s'han estimat les hores d'aquestes tasques de forma elevada, de manera que un contratemps en aquests aspectes no hauria de suposar una penalització rellevant en la duració total del projecte. Pel que fa al tercer risc, és el menys important degut al seu baix impacte i la seva baixa probabilitat.

## 3 Pressupost

L'objectiu d'aquest capítol és calcular el cost aproximat que suposa el desenvolupament del projecte, considerant tant els recursos materials com humans utilitzats en qualsevol moment del treball. Per a fer-ho, es llisten tots els recursos en les següents taules, on es pot veure el cost corresponent a cadascun. Després, s'estimen els costos que poden suposar les possibles desviacions que puguin aparèixer durant el desenvolupament. També es detalla de quina manera es gestionaran aquestes desviacions, amb indicadors numèrics de càlcul que en facilitin el control.

### 3.1 Identificació dels costos

Per a poder calcular el cost total del projecte, es llisten tots els recursos que suposen una despesa. Aquests recursos poden ser de caire material (hardware i software), humà (personal implicat) i general (llum, Internet, etc). A més, es consideren també possibles imprevistos i contingències per a sobreestimar el pressupost i així evitar costos per sobre del previst.

#### 3.1.1 Recursos materials

Els recursos hardware i software que estan prevists utilitzar per al projecte són:

Codi	Recurs	Tipus	Cost (€)	Amortització (€)
RM1	MSI Prestige PS42	Hardware	1149	62,33
RM2	Git	Software	Gratuït	-
RM3	Webstorm IDE	Software	Gratuït	-
RM4	MikTex	Software	Gratuït	-
RM5	Azure DevOps	Software	Gratuït	-
RM6	Google Chrome	Software	Gratuït	-
RM7	DBeaver	Software	Gratuït	-
RM8	Node Package Manager	Software	Gratuït	-
RM9	Angular	Software	Gratuït	-
RM10	Ganache	Software	Gratuït	-
-	<b>TOTAL</b>	-	-	62,33

Taula 3: Costos materials

#### 3.1.2 Recursos humans

Per a poder calcular el cost de les tasques a realitzar, és imprescindible conèixer la duració i el rol assignats a cada tasca a més del sou de cadascun dels rols implicats. Els salaris considerats per a fer l'estimació del cost dels recursos humans són:

<b>Rol</b>	<b>Sou/any</b>	<b>Sou/hora</b>	<b>Sou/hora + SS</b>
Scrum Master	42.600€	20,48€	26,62€
Equip de desenvolupament	33.760€	16,23€	21,10€
Product owner	44.783€	21,53€	27,99€

Taula 4: Salaris dels rols implicats en el projecte

La segona columna conté el salari mitjà anual de cada rol, i amb aquest es poden calcular preu per hora, i el preu per hora amb un cost extra per les despeses de la Seguretat Social. Aquesta despesa extra s'ha calculat multiplicant per 1,3 el preu per hora, i és la que s'utilitza per a calcular el cost de cada tasca.

En la Taula 5 es llisten les tasques del Diagrama de Gantt, cadascuna amb el rol associat i la duració en hores. Amb aquestes dues dades, i considerant els salaris de cada rol detallats en la Taula 4, obtenim els costos corresponents als recursos humans:

Codi	Tasca	Rol	Duració (h)	Cost (€)
GP1	Definició de l'abast i contextualització	SM	20	532,48
GP2	Planificació temporal	SM	20	532,48
GP3	Gestió econòmica i sostenibilitat	SM	20	532,48
GP4	Revisió de la documentació	SM	5	133,12
GP5	Preparació del tauler Agile	SM	10	266,24
RC1	Investigació i recerca	DEV	25	527,475
RC2	Preparació d'exemples	DEV	10	210,99
DEV1	Disseny del model de dades	DEV	15	316,485
DEV2	Disseny de les consultes de dades	DEV	15	316,485
CN1	Instal·lació de software	DEV	10	210,99
DEV3	Desenvolupament dels Smart Contracts	DEV	35	738,465
DOC1	Documentació de l'sprint 1	SM	15	399,36
GP6	Retrospectiva de l'sprint 1	SM, PO	5	273,065
DEV4	Desenvolupament de l'aplicació web	DEV	35	738,465
DEV5	Interacció amb els Smart Contracts	DEV	30	632,97
CN2	Desplegament dels Smart Contracts	DEV	20	421,98
DEV6	Obtenció de dades de la xarxa blockchain	DEV	25	527,475
DOC2	Documentació de l'sprint 2	SM	15	399,36
GP7	Retrospectiva de l'sprint 2	SM, PO	5	273,065
DEV7	Disseny i creació de les taules	DEV	30	632,97
DEV8	Desenvolupament de les vistes	DEV	25	527,475
CN3	Desplegament de la base de dades	DEV	10	210,99
DEV9	Interacció amb la base de dades	DEV	25	527,475
DEV10	Mostrat de dades en l'aplicació web	DEV	20	421,98
DOC3	Documentació de l'sprint 2	SM	15	399,36
GP8	Retrospectiva de l'sprint 2	SM, PO	5	273,065
DOC4	Finalització de la memòria	SM	20	532,48
-	<b>TOTAL</b>	-	<b>485</b>	<b>11509,225</b>

Taula 5: Costos humans

## 3.2 Estimació dels costos

Pel que fa als costos “indirectes” del treball, es consideren les despeses generals (llum, Internet, etc) i les possibles imprevistos i contingències.

### 3.2.1 Recursos genèrics

En la Taula 6 es llisten tots aquells recursos que suposen una despesa en el treball, però que no es poden calcular de forma directa.

El càlcul de la despesa d'electricitat s'ha obtingut considerant que un ordinador portàtil consumeix, de mitjana, 300 watts l'hora, és a dir, 30kWh. El preu actual del kW és de 0.62802€, tot i que fluctua molt degut a la situació de crisi actual. La durada del projecte, tal i com s'especifica en l'estimació de

tasques, és de 485 hores, i el portàtil és necessari per a totes les tasques. Així doncs, amb aquestes dades podem calcular el cost de l'electricitat utilitzada en el desenvolupament d'aquest projecte:

$$0,3 \text{ kWh} * 0,6 \text{ €/kWh} * 485\text{h} = 91,374 \text{ €}$$

Pel que fa a la despesa d'Internet, s'ha calculat d'una manera més senzilla: la tarifa d'Internet és de 30€ mensuals, el que equival a dir 1€/dia. La duració del projecte és de 99 dies, per tant el cost total corresponent a la connexió a Internet és de 99€.

També es considera com a cost genèric el mobiliari de l'espai de treball, com poden ser l'escriptori (150€), la cadira (190€) i la làmpada (25€). Es considera que la vida útil d'aquest tipus de mobiliari són 10 anys, per tant es calcula l'amortització del preu segons els dies que dura el projecte, en aquest cas 99. Per tant, l'amortització de la zona de treball és de 9,89€.

La Taula 6 conté tots els costos corresponents als recursos genèrics, que sumen un total de 200,26€:

Codi	Recurs	Tipus	Cost (€)	Amortització (€)
RG1	Internet	Genèric	99,00	99,00
RG2	Electricitat	Genèric	91,37	91,37
RG3	Escriptori IDE	Mobiliari	150,00	4,07
RG4	Cadira	Mobiliari	190,00	5,15
RG5	Làmpada	Mobiliari	25,00	0,67
-	<b>TOTAL</b>	-	-	200,26

Taula 6: Costos genèrics

### 3.2.2 Cost de contingència

El cost de contingència és una quantitat de diners que s'afegeix al pressupost del projecte per tal de cobrir aquells imprevistos que no s'han anticipat. Aquest cost es calcula com un percentatge del valor total del pressupost, i sol estar determinat pel sector i el nivell de detall del pressupost. En projectes informàtics sol estar entre el 10 i el 20%, essent 15% el marge escollit per a aquest projecte.

El cost dels recursos materials, humans i genèrics és:

$$66,33 + 11509,22 + 200,26 = 11775,81$$

Per tant, el cost de contingència és:

$$11775,81\text{€} * 0,15 = \mathbf{1766,37\text{€}}$$



### 3.2.3 Cost d'imprevistos

En l'apartat de gestió de riscos s'especifiquen els possibles riscos que poden aparèixer durant el desenvolupament del projecte, així com les accions a prendre en cas que apareguin. Aquestes accions tenen un també un cost, i és per això que s'ha d'afegir al pressupost una estimació de la despesa que podria suposar la mitigació d'aquests riscos. El càlcul del cost d'un imprevist s'ha aplicat de la següent manera:

$$\text{probabilitat d'ocurrència (\%)} * \text{sou del rol assignat (€/h)} * \text{temps dedicació (h)} = \text{cost (€)}$$

La Taula 7 conté un llistat dels riscos i el cost que suposaria cadascun, utilitzant la fórmula mencionada:

Risc	Probabilitat	Temps	Rol	Cost (€)
R1	75%	40h	DEV	633,00
R2	50%	30h	DEV	316,50
R3	25%	15h	DEV	79,13
-	<b>TOTAL</b>	-	-	1.028,63

Taula 7: Cost dels riscos identificats

### 3.2.4 Pressupost final

Finalment, un cop detallats tots els costos de cada categoria, es pot obtenir una estimació del pressupost total del projecte:

Tipus	Cost (€)
Recursos materials	62,33
Recursos humans	11.509,23
Recursos genèrics	200,26
Contingència	1.766,37
Imprevistos	1.028,63
<b>TOTAL</b>	<b>14.566,82</b>

Taula 8: Pressupost final

## 3.3 Control de gestió

Tot i que es defineixen de manera precisa l'estimació de tasques i els costos del projecte, s'ha de mantenir un control del desenvolupament per a garantir que totes les tasques segueixen la planificació dissenyada. Aquest cas és gairebé irreal, ja que és molt comú que apareguin desviacions. Aquestes es calculen comparant els valors estimats amb els valors reals, més concretament amb les fórmules següents:

- Desviació d'hores consumides per tasca:

$$(hores estimades - hores reals) * cost estimat$$

- Desviació de costos segons les hores consumides per tasca:

$$(hores estimades - hores reals) * cost real$$

- Desviació total de costos materials:

$$(cost RRMM estimat - cost RRMM real)$$

- Desviació total de costos humans:

$$(cost RRHH estimat - cost RRHH real)$$

- Desviació total de costos genèrics:

$$(cost RRGG estimat - cost RRGG real)$$

- Desviació total de costos d'imprevistos:

$$(cost d'imprevistos estimat - cost d'imprevistos real)$$

- Desviació total d'hores:

$$(hores estimades - hores reals)$$

- Desviació total de costos:

$$(cost total estimat - cost total real)$$

## 4 Informe de sostenibilitat

Després de respondre l'enquesta del EDINSOST, he pogut adonar-me de la profunditat que realment té l'aspecte de la sostenibilitat en un projecte. Generalment, no som conscients de l'impacte que té un projecte TIC, especialment en els àmbits més indirectes, com pot ser la dimensió social. Podem pensar en algun dels impactes que pot tenir en la dimensió econòmica o mediambiental, ja que són àmbits més intuïtius, però moltes vegades sense identificar la majoria dels problemes que poden aparèixer. A més, tot i poder localitzar-los, no som capaços de calcular l'impacte que poden tenir aquests problemes, i és per això que solem menysprear la importància d'aquests càlculs.

### 4.1 Dimensió econòmica

Aquest és, probablement, l'aspecte més intuïtiu a l'hora de valorar el seu impacte. Tots els costos econòmics estan detallats en l'apartat del pressupost, on s'han considerat tots els recursos utilitzats de forma detallada. Aquest és un projecte de caire exploratori, i això implica dos factors importants que cal destaca: el primer és que es poden utilitzar eines i llibreries de codi obert, que permeten reduir el cost econòmic del projecte. Si ens fixem en el llistat de recursos software necessaris, tots els escollits són gratuïts. El segon aspecte és que, al no ser un producte real sinó una investigació, no caldrà posar en producció el sistema desenvolupat, de manera que no s'utilitzaran recursos de tercers, com poden ser bases de dades, servidors, etc. Això també redueix molt el cost del projecte, ja que el desplegament d'aplicacions software en un entorn real requereix moltes eines i serveis de tercers, els quals tenen un preu elevat.

### 4.2 Dimensió mediambiental

Al ser un projecte software, l'únic recurs material que es necessita per al desenvolupament és l'ordinador portàtil, que té un impacte mediambiental degut a que contenen metalls i compostos orgànics tòxics com el mercuri, el plom o el cadmi [12]. És per això que és important reciclar aquests aparells una vegada acaba la seva vida útil.

Un altre aspecte que afecta la sostenibilitat és l'ús d'electricitat, ja sigui pel propi ordinador o per l'electricitat consumida pels recursos de tercers utilitzats (bases de dades, servidors, etc). Una problemàtica força polèmica en el sector del desenvolupament blockchain és el consum d'energia i les emissions que genera aquesta tecnologia. Tal i com s'explica en l'apartat d'introducció, la blockchain destaca per la replicació i la disponibilitat de les dades. Això s'aconsegueix replicant les dades en cadascun dels nodes de la xarxa, i això fa que es consumeixin molts més recursos que en una infraestructura convencional. A més, en moltes xarxes blockchain existeixen els "miners", que són les màquines encarregades de generar els blocs. Per tal de generar aquests blocs, els miners han de resoldre operacions criptogràfiques de molta dificultat, i és per això que necessiten una

gran potència de computació. Això deriva també a un gran consum de recursos [8], fet que té un impacte directe en el medi ambient.

### **4.3 Dimensió social**

Aquest és l'aspecte més difícil de valorar, ja que el tema escollit no té un gran impacte en la dimensió social. Un factor a destacar és la popularitat actual dla blockchain, fet que pot ajudar a convertir aquest projecte d'investigació en un sistema que s'utilitzi, en un futur, en un escenari real. A més, el registre de transaccions i la immutabilitat de les dades fa que la transparència sigui un aspecte clau d'aquesta tecnologia, fet que proporciona credibilitat i confiança als seus usuaris.

## 5 Disseny

L'objectiu d'aquest capítol és definir tant l'arquitectura del sistema i els seus elements, com el domini dissenyat per a demostrar el funcionament i la utilitat del sistema. En aquest cas, el domini definit és una adaptació de l'exemple utilitzat per Jaume Sistac en la seva tesi "Construcció automàtica de prototipus de sistemes d'informació a partir de models conceptuals deductius amb el llenguatge DADES" [6]. Aquest, tracta sobre un inventari, que emmagatzema productes i en facilita la seva gestió. Així doncs, la solució creada permet als usuaris interactuar amb un sistema de gestió d'inventari, el qual ofereix els funcionaments bàsics d'aquest: afegir un producte, fer una comanda, actualitzar el preu d'un producte, afegir estoc, etc.

### 5.1 Arquitectura

El usuaris interactuen amb el sistema a través d'un navegador web, des del qual s'interactua de forma directa amb els Smart Contracts, sense la necessitat d'un *middleware*. Totes les transaccions que realitzen els usuaris queden enregistrades en la blockchain, de manera que són consultables en qualsevol moment. Per una còmoda interpretació d'aquestes transaccions, s'ha desenvolupat un *middleware*, encarregat de: Per una banda, emmagatzemar les transaccions a la base dades. Per l'altra, oferir una API REST que permet una obtenció còmoda de les dades. Així doncs, el mateix frontend que permet la interacció amb els contractes, és també l'encarregat de mostrar les dades "on demand", aplicant filtres pràctics per a l'usuari. El diagrama de l'arquitectura està representat en la Figura 5.

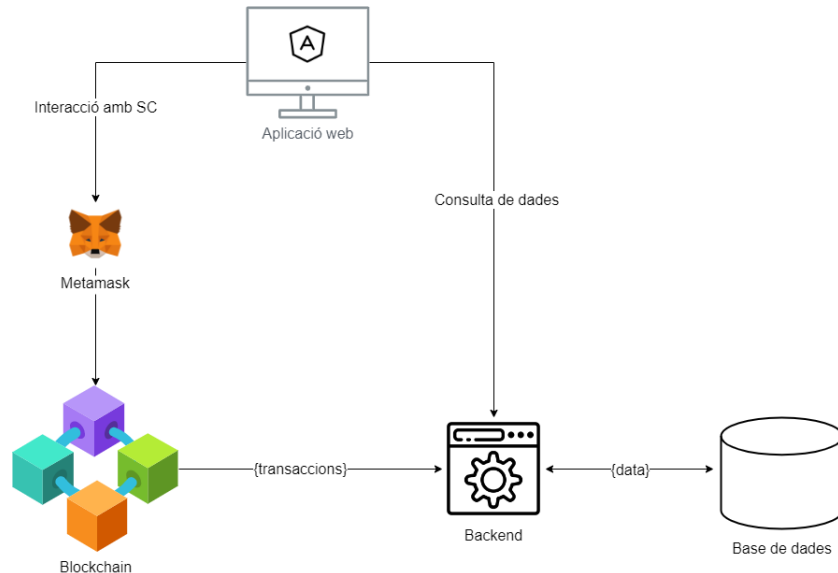


Figura 5: Diagrama de l'arquitectura del sistema

Així doncs, el sistema està format per 4 elements: frontend, backend, blockchain i base de dades. A continuació es profunditza en cadascuna d'aquestes parts.

### 5.1.1 Frontend

S'ha desenvolupat una aplicació web que permet tant interactuar amb els Smart Contracts com consultar les dades del sistema. Aquesta aplicació s'ha desenvolupat amb el framework Angular, que utilitza el llenguatge Typescript. Pel que fa al primer punt, la interacció amb blockchain, s'utilitza la llibreria "ethers.js" [13], la qual és compatible amb Angular i permet una fàcil integració amb Metamask.

**Metamask:** Metamask [16] és una "wallet" que permet interactuar amb les dApps d'Ethereum a través d'una extensió de navegador, i la més popular entre els usuaris d'aquesta xarxa. Una "wallet" és un programari que emmagatzema claus públiques i privades per a poder validar transaccions en la blockchain. A més, la majoria de "wallets" actuals també permeten visualitzar balanços, veure els historials de les transaccions realitzades, enviar tokens a altres adreces, etc. La Figura 6 mostra la pàgina principal d'aquesta eina, la qual mostra els balanços dels tokens que conté aquella adreça (en el nostre cas, tokens ETH de la xarxa Rinkeby, una de les "testnets" d'Ethereum).

### 5.1.2 Backend

El backend és considerat un *middleware*, ja que és l'intermediari de dos processos diferents del sistema. Per una banda, és l'encarregat de traslladar les transaccions de la blockchain a la base de dades, i per altra banda serveix com a REST API per a servir dades al frontend. Aquest *middleware* està desenvolupat amb Node.js, una capa de servidor basada en Javascript. Pel que fa a la API, s'utilitza "Express.js", la llibreria més utilitzada en aquest entorn per al desenvolupament d'APIs.

### 5.1.3 Blockchain

La blockchain escollida per aquest sistema és Ethereum, la qual va ser la primera en introduir el concepte "Smart Contract". Això fa que sigui la més popular per al desenvolupament d'aplicacions descentralitzades (dApps), i això fa que tingui la major comunitat i millor experiència de desenvolupador.

**Smart Contracts:** Els Smart Contractes són els programes que es desenvolupen per a executar instruccions en la blockchain. En el cas d'Ethereum, s'escriuen utilitzant el llenguatge "Solidity". Aquest és un llenguatge d'alt nivell orientat als contractes, té una sintaxi similar a Javascript i està orientat específicament a la Màquina Virtual d'Ethereum (EVM). La invocació d'un mètode en un Smart Contract genera una transacció en la blockchain, la qual ha de ser validada per tal que produeixi la invocació del mètode amb èxit.

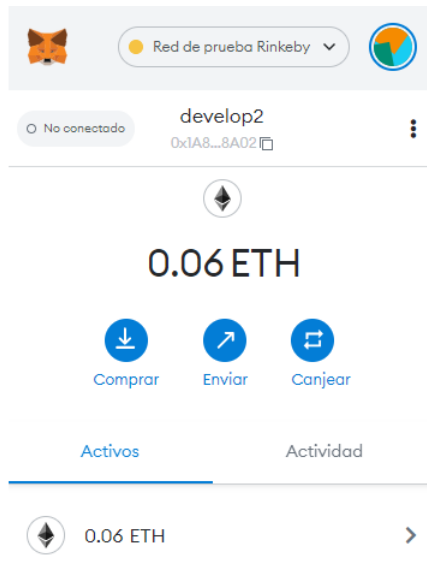


Figura 6: Vista principal de Metamask

Si és aprovada, la transacció és afegida al bloc corresponent, de manera que s'emmagatzema en la blockchain i no és modificable. En cas contrari, el resultat de la invocació del mètode no queda registrat en la blockchain.

Totes les operacions que es realitzen sobre la blockchain (desplegament de contractes, execució de mètodes, etc) tenen un cost, el qual es paga amb el token natiu de la xarxa. Durant la primera etapa del projecte, quan es desenvolupen els Smart Contracts, s'utilitza l'eina Ganache [19] abans de desplegar-los a la xarxa Ethereum. Ganache és una eina que permet aixecar una xarxa Ethereum local i un entorn de proves de forma còmoda i intuïtiva (veure Figura 7). Això agilitza el procés de desplegament de contractes, fet que sol repetir-se amb freqüència durant el desenvolupament d'aquests.

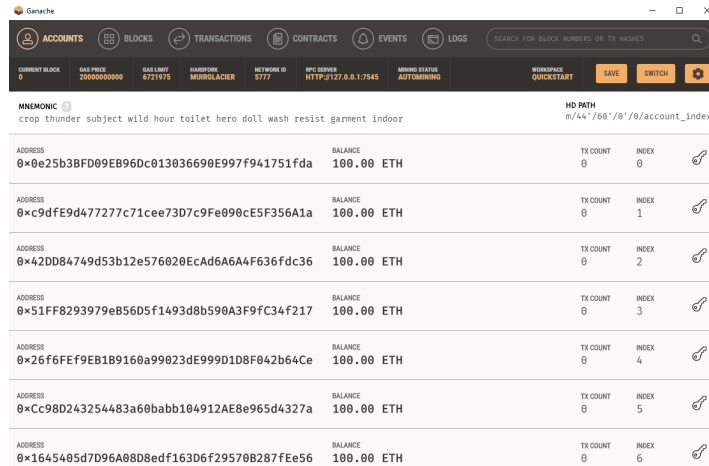


Figura 7: Vista principal de l'eina Ganache

Després d'aquesta primera etapa, quan els contractes ja es troben en un estat més consolidat, es fa el desplegament a una *testnet*. Aquesta xarxa ofereix les mateixes funcionalitats que la *mainnet*, però amb la diferència que es poden obtenir tokens de forma gratuïta. Com que aquest projecte és de caire exploratori, no es fa el desplegament a la *mainnet*, ja que no ofereix cap benefici útil per a aquest sistema respecte la xarxa de proves.

### 5.1.4 Bases de dades

Pel que fa a la base de dades, s'utilitza una base de dades relacional PostgreSQL [17]. Aquest SGBD ofereix una molt bona integració amb Node.js, i a més ja tinc experiència utilitzant-lo, el qual em permet centrar-me en àrees més desconegudes, com pot ser la part blockchain.

## 5.2 Domini

Tal i com es menciona prèviament, el domini del sistema tracta sobre la gestió d'un inventari. Aquest inventari és un conjunt de productes, i permet funcionalitats com afegir un producte, esborrar-lo, comprar-lo, etc.

### 5.2.1 Model de classes

El model de classes aplica, en aquest cas, per la part blockchain, de manera que aquests mètodes i aquestes classes estan definides en els Smart Contracts, desenvolupats en Solidity. Com podem veure a la Figura 8, la classe *Inventory* té els 5 mètodes principals (afegir, esborrar, comprar, reposar i modificar preu). Tot i així, per qüestions de *testing*, s'han afegit els mètodes "GetProductIds" i "GetProductQuantity". Cal destacar que l'objectiu d'aquests mètodes és només per a fer proves, ja que la consulta de dades es fa a través de la API REST.



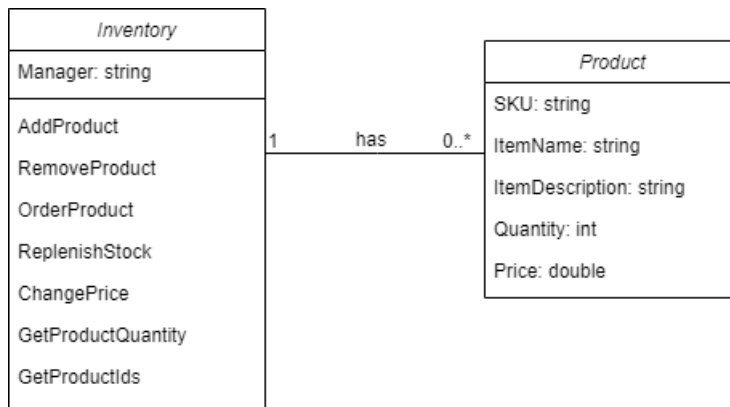


Figura 8: Model de classes

### 5.2.2 Casos d'ús

S'han definit els principals casos d'ús que pot tenir un usuari a l'hora d'utilitzar el sistema. Aquests, queden classificats en dues categories:

- Tipus "Gestió d'inventari": Els casos d'ús relacionats amb els mètodes del contracte.
- Tipus "Consulta de dades": Els casos d'ús relacionats amb la consulta de dades.

En el domini definit, que tracta sobre un inventari, els casos d'ús que s'han considerat són:

<b>Cas d'ús</b>	Afegir producte
<b>Actor</b>	Usuari
<b>Típus</b>	Gestió d'inventari
<b>Flux</b>	<p>L'usuari es troba en la secció "Manage" de l'aplicació web</p> <ol style="list-style-type: none"> <li>1. La pàgina mostra cadascuna de les accions que pot realitzar l'usuari</li> <li>2. L'usuari selecciona l'opció "Add Product"</li> <li>3. La pàgina mostra un formulari amb 5 camps a omplir sobre el producte que es vol afegir: identificador, nom, descripció, quantitat i preu.</li> <li>4. L'usuari omple els camps i envia el formulari</li> <li>5. La pàgina mostra la finestra de Metamask, que demana la validació de l'usuari per a completar la transacció</li> <li>6. L'usuari valida la transacció a Metamask</li> <li>7. La pàgina mostra un missatge amb el resultat de la transacció, segons si s'ha completat amb èxit o amb error</li> </ol>

Taula 9: Cas d'ús "Afegir producte"

<b>Cas d'ús</b>	Comprar producte
<b>Actor</b>	Usuari
<b>Típus</b>	Gestió d'inventari
<b>Flux</b>	<p>L'usuari es troba en la secció "Manage" de l'aplicació web</p> <ol style="list-style-type: none"> <li>1. La pàgina mostra cadascuna de les accions que pot realitzar l'usuari</li> <li>2. L'usuari selecciona l'opció "Order Product"</li> <li>3. La pàgina mostra un formulari amb 2 camps a omplir sobre el producte que es vol comprar: identificador i quantitat.</li> <li>4. L'usuari omple els camps i envia el formulari</li> <li>5. La pàgina mostra la finestra de Metamask, que demana la validació de l'usuari per a completar la transacció</li> <li>6. L'usuari valida la transacció a Metamask</li> <li>7. La pàgina mostra un missatge amb el resultat de la transacció, segons si s'ha completat amb èxit o amb error</li> </ol>

Taula 10: Cas d'ús "Comprar producte"

<b>Cas d'ús</b>	Reposar estoc
<b>Actor</b>	Usuari
<b>Típus</b>	Gestió d'inventari
<b>Flux</b>	<p>L'usuari es troba en la secció "Manage" de l'aplicació web</p> <ol style="list-style-type: none"> <li>1. La pàgina mostra cadascuna de les accions que pot realitzar l'usuari</li> <li>2. L'usuari selecciona l'opció "Replenish Stock"</li> <li>3. La pàgina mostra un formulari amb 2 camps a omplir sobre el producte que es vol reposar: identificador i quantitat.</li> <li>4. L'usuari omple els camps i envia el formulari</li> <li>5. La pàgina mostra la finestra de Metamask, que demana la validació de l'usuari per a completar la transacció</li> <li>6. L'usuari valida la transacció a Metamask</li> <li>7. La pàgina mostra un missatge amb el resultat de la transacció, segons si s'ha completat amb èxit o amb error</li> </ol>

Taula 11: Cas d'ús "Reposar estoc"

<b>Cas d'ús</b>	Eliminar producte
<b>Actor</b>	Usuari
<b>Típus</b>	Gestió d'inventari
<b>Flux</b>	<p>L'usuari es troba en la secció "Manage" de l'aplicació web</p> <ol style="list-style-type: none"> <li>1. La pàgina mostra cadascuna de les accions que pot realitzar l'usuari</li> <li>2. L'usuari selecciona l'opció "Delete Product"</li> <li>3. La pàgina mostra un formulari amb un camp per a especificar l'identificador del producte que es vol eliminar</li> <li>4. L'usuari omple els camps i envia el formulari</li> <li>5. La pàgina mostra la finestra de Metamask, que demana la validació de l'usuari per a completar la transacció</li> <li>6. L'usuari valida la transacció a Metamask</li> <li>7. La pàgina mostra un missatge amb el resultat de la transacció, segons si s'ha completat amb èxit o amb error</li> </ol>

Taula 12: Cas d'ús "Eliminar producte"

<b>Cas d'ús</b>	Modificar preu d'un producte
<b>Actor</b>	Usuari
<b>Típus</b>	Gestió d'inventari
<b>Flux</b>	<p>L'usuari es troba en la secció "Manage" de l'aplicació web</p> <ol style="list-style-type: none"> <li>1. La pàgina mostra cadascuna de les accions que pot realitzar l'usuari</li> <li>2. L'usuari selecciona l'opció "Change price"</li> <li>3. La pàgina mostra un formulari amb 2 camps a omplir sobre el producte que es vol modificar: identificador i preu</li> <li>4. L'usuari omple els camps i envia el formulari</li> <li>5. La pàgina mostra la finestra de Metamask, que demana la validació de l'usuari per a completar la transacció</li> <li>6. L'usuari valida la transacció a Metamask</li> <li>7. La pàgina mostra un missatge amb el resultat de la transacció, segons si s'ha completat amb èxit o amb error</li> </ol>

Taula 13: Cas d'ús "Modificar preu d'un producte"

<b>Cas d'ús</b>	Consultar quantitat d'un producte en una data
<b>Actor</b>	Usuari
<b>Típus</b>	Consulta de dades
<b>Flux</b>	<p>L'usuari es troba en la secció "Explore" de l'aplicació web</p> <ol style="list-style-type: none"> <li>1. La pàgina mostra cadascuna de les consultes que pot realitzar l'usuari</li> <li>2. L'usuari selecciona l'opció "Product quantity"</li> <li>3. La pàgina mostra una barra de cerca amb un camp de text i un selector de data</li> <li>4. L'usuari introdueix l'identificador (SKU) del producte i la data desitjada</li> <li>5. La pàgina mostra la informació del producte en la data seleccionada</li> </ol>

Taula 14: Cas d'ús "Consultar quantitat d'un producte en una data"

<b>Cas d'ús</b>	Consultar preu d'un producte en una data
<b>Actor</b>	Usuari
<b>Tipus</b>	Consulta de dades
<b>Flux</b>	<p>L'usuari es troba en la secció "Explore" de l'aplicació web</p> <ol style="list-style-type: none"> <li>1. La pàgina mostra cadascuna de les consultes que pot realitzar l'usuari</li> <li>2. L'usuari selecciona l'opció "Product price"</li> <li>3. La pàgina mostra una barra de cerca amb un camp de text i un selector de data</li> <li>4. L'usuari introdueix l'identificador (SKU) del producte i la data desitjada</li> <li>5. La pàgina mostra la informació del producte en la data seleccionada</li> </ol>

Taula 15: Cas d'ús "Consultar preu d'un producte en una data"

<b>Cas d'ús</b>	Consultar productes afegits segons quantitat
<b>Actor</b>	Usuari
<b>Tipus</b>	Consulta de dades
<b>Flux</b>	<p>L'usuari es troba en la secció "Explore" de l'aplicació web</p> <ol style="list-style-type: none"> <li>1. La pàgina mostra cadascuna de les consultes que pot realitzar l'usuari</li> <li>2. L'usuari selecciona l'opció "Transactions by quantity"</li> <li>3. La pàgina mostra una barra de cerca amb els filtres: estat, data inici, data fi i ordre</li> <li>4. L'usuari selecciona l'estat "added", juntament amb la resta de filtres desitjats</li> <li>5. La pàgina mostra una taula amb els productes afegits entre les dates seleccionades, i ordenats per <b>quantitat</b> segons el filtre escollit</li> </ol>

Taula 16: Cas d'ús "Consultar productes afegits segons quantitat"

<b>Cas d'ús</b>	Consultar productes venuts segons quantitat
<b>Actor</b>	Usuari
<b>Tipus</b>	Consulta de dades
<b>Flux</b>	<p>L'usuari es troba en la secció "Explore" de l'aplicació web</p> <ol style="list-style-type: none"> <li>1. La pàgina mostra cadascuna de les consultes que pot realitzar l'usuari</li> <li>2. L'usuari selecciona l'opció "Transactions by quantity"</li> <li>3. La pàgina mostra una barra de cerca amb els filtres: estat, data inici, data fi i ordre</li> <li>4. L'usuari selecciona l'estat "sold", juntament amb la resta de filtres desitjats</li> <li>5. La pàgina mostra una taula amb els productes venuts entre les dates seleccionades, i ordenats per <b>quantitat</b> segons el filtre escollit</li> </ol>

Taula 17: Cas d'ús "Consultar productes venuts segons quantitat"

<b>Cas d'ús</b>	Consultar productes reposats segons quantitat
<b>Actor</b>	Usuari
<b>Tipus</b>	Consulta de dades
<b>Flux</b>	<p>L'usuari es troba en la secció "Explore" de l'aplicació web</p> <ol style="list-style-type: none"> <li>1. La pàgina mostra cadascuna de les consultes que pot realitzar l'usuari</li> <li>2. L'usuari selecciona l'opció "Transactions by quantity"</li> <li>3. La pàgina mostra una barra de cerca amb els filtres: estat, data inici, data fi i ordre</li> <li>4. L'usuari selecciona l'estat "replenished", juntament amb la resta de filtres desitjats</li> <li>5. La pàgina mostra una taula amb els productes reposats entre les dates seleccionades, i ordenats per <b>quantitat</b> segons el filtre escollit</li> </ol>

Taula 18: Cas d'ús "Consultar productes reposats segons quantitat"

<b>Cas d'ús</b>	Consultar productes afegits segons preu
<b>Actor</b>	Usuari
<b>Tipus</b>	Consulta de dades
<b>Flux</b>	<p>L'usuari es troba en la secció "Explore" de l'aplicació web</p> <ol style="list-style-type: none"> <li>1. La pàgina mostra cadascuna de les consultes que pot realitzar l'usuari</li> <li>2. L'usuari selecciona l'opció "Transactions by price"</li> <li>3. La pàgina mostra una barra de cerca amb els filtres: estat, data inici, data fi i ordre</li> <li>4. L'usuari selecciona l'estat "added", juntament amb la resta de filtres desitjats</li> <li>5. La pàgina mostra una taula amb els productes afegits entre les dates seleccionades, i ordenats per <b>preu</b> segons el filtre escollit</li> </ol>

Taula 19: Cas d'ús "Consultar productes afegits segons preu"

<b>Cas d'ús</b>	Consultar productes venuts segons preu
<b>Actor</b>	Usuari
<b>Tipus</b>	Consulta de dades
<b>Flux</b>	<p>L'usuari es troba en la secció "Explore" de l'aplicació web</p> <ol style="list-style-type: none"> <li>1. La pàgina mostra cadascuna de les consultes que pot realitzar l'usuari</li> <li>2. L'usuari selecciona l'opció "Transactions by price"</li> <li>3. La pàgina mostra una barra de cerca amb els filtres: estat, data inici, data fi i ordre</li> <li>4. L'usuari selecciona l'estat "sold", juntament amb la resta de filtres desitjats</li> <li>5. La pàgina mostra una taula amb els productes venuts entre les dates seleccionades, i ordenats per <b>preu</b> segons el filtre escollit</li> </ol>

Taula 20: Cas d'ús "Consultar productes venuts segons preu"

<b>Cas d'ús</b>	Consultar productes reposats segons preu
<b>Actor</b>	Usuari
<b>Típus</b>	Consulta de dades
<b>Flux</b>	<p>L'usuari es troba en la secció "Explore" de l'aplicació web</p> <ol style="list-style-type: none"> <li>1. La pàgina mostra cadascuna de les consultes que pot realitzar l'usuari</li> <li>2. L'usuari selecciona l'opció "Transactions by price"</li> <li>3. La pàgina mostra una barra de cerca amb els filtres: estat, data inici, data fi i ordre</li> <li>4. L'usuari selecciona l'estat "replenished", juntament amb la resta de filtres desitjats</li> <li>5. La pàgina mostra una taula amb els productes reposats entre les dates seleccionades, i ordenats per <b>preu</b> segons el filtre escollit</li> </ol>

Taula 21: Cas d'ús "Consultar productes reposats segons preu"



## 6 Desenvolupament

Tal i com s'explica a l'apartat 5, el sistema està format per 4 elements: frontend, backend, base de dades i blockchain. Aquests elements s'han desenvolupat en el mateix ordre que segueix el flux de les dades:

- Frontend → Blockchain
- Blockchain → Base de dades
- Base de dades → Frontend

Cal tenir en compte, a més, que sempre que s'interactua amb la BD, es fa a través de la API (backend).

A continuació es descriuen aquests 3 processos de forma detallada:

### 6.1 Interacció amb Smart Contracts

Tal i com es detalla en l'apartat 5.1, l'usuari interactua amb els Smart Contracts a través de l'aplicació web, la qual utilitza Metamask per a gestionar les transaccions.

En el nostre sistema, aquesta eina ens permet aprovar i validar totes les crides que es fan sobre el nostre Smart Contract. A més, en aquest es pot definir una condició per tal de que només pugui executar certs mètodes del contracte una adreça en concret. Per exemple, podríem establir que la crida del mètode per a afegir un producte fos només permesa per a la nostra adreça. També cal tenir en compte que totes les transaccions de la blockchain suposen una despesa de recursos i "energia", el qual està denominat com "gas". Aquestes comissions de transacció es paguen amb els tokens d'aquella xarxa, i Metamask serà l'encarregat de gestionar també aquests pagaments. La vista que apareix a Metamask quan s'invoca un mètode del Smart Contract per poder validar-la es pot veure en la Figura 9.

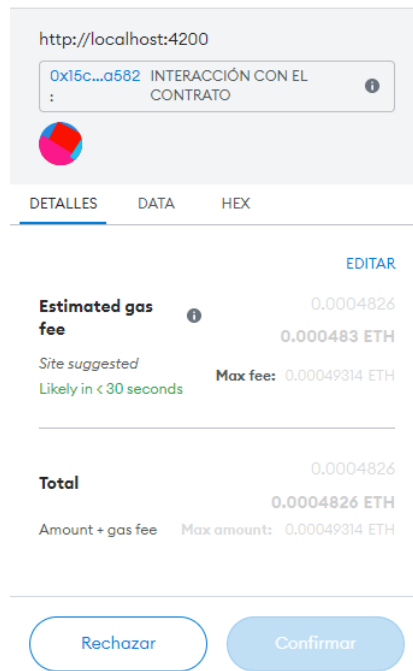


Figura 9: Vista de confirmació de Metamask per a executar el mètode d'un Smart Contract

En el nostre cas, els Smart Contracts desenvolupats tenen com a objectiu oferir la mateixa funcionalitat que un sistema *backend* tradicional, com podria ser una API junt amb una base de dades. Tal i com es detalla en l'apartat 5, el domini d'aquesta part simula el funcionament d'un inventari. Això, en termes de codi, s'ha definit a través d'un "struct" `Product`, que conté els atributs definits. Per a emmagatzemar tots els productes de l'inventari s'utilitza un "mapping", el qual té com a clau l'identificador de producte (encriptat) i com a valor tot l'estruct del producte corresponent. La declaració d'aquestes estructures en el codi es pot veure en la Figura 10.

```
struct Product{
    string sku; //always unique to the product
    string itemName;
    string itemDescription;
    uint qty; //available quantity
    uint price;
    bool isValue; // to keep track if it has been deleted
}

mapping(bytes32 => Product) Inventory;
```

Figura 10: Definició d'estructures de dades del Smart Contract

### 6.1.1 Aplicació web Angular

Pel que fa a l'aplicació web desenvolupada, aquesta disposa d'una secció per a interactuar amb els Smart Contracts. La vista d'aquesta secció, mostra primer totes les accions que l'usuari pot executar sobre el contracte:

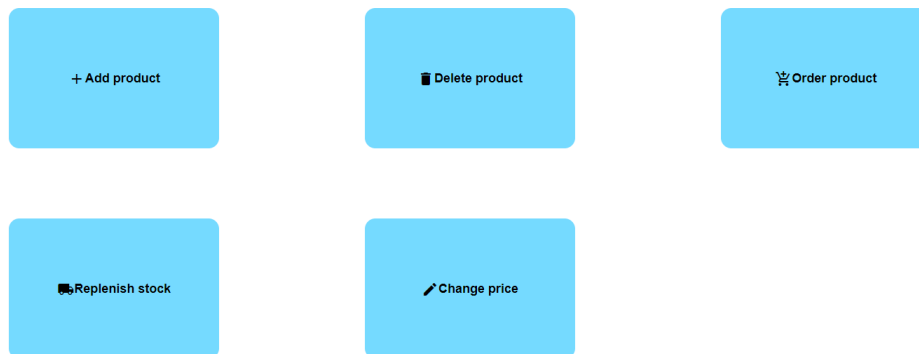


Figura 11: Vista "Manage" de l'aplicació web

S'ha desenvolupat una vista per a cadascun dels mètodes del contracte, les quals són simplement un formulari per a introduir les dades necessàries per a executar-los. Un cop l'usuari omple el formulari i decideix enviar-lo, apareix la vista de confirmació de transacció de Metamask. Una vegada l'usuari prem el botó per a confirmar-la, s'envia la transacció a la blockchain per a intentar emmagatzemar el resultat de l'operació. Després de l'espera del temps de confirmació, obtindrà un missatge comunicant el resultat de la transacció. Un exemple d'aquest procés es pot veure en la Figura 12, on s'ha utilitzat la vista per a afegir un producte, i s'ha obtingut un missatge d'èxit després d'haver-se completat correctament el procés.

## Add product

SKU \*  
usb3

---

Name \*  
USB 3.0 Adapter

---

Description \*  
Ugreen USB 3.0 Adapter for laptops

---

Quantity \*  
18

---

Price \*  
32

---

Add product

Successfully added product 'usb3' OK

Figura 12: Vista "Add product" de l'aplicació web

### 6.2 Obtenció i interpretació de les dades

Un dels objectius principals del projecte és poder accedir a un registre de totes les transaccions que s'han realitzat sobre un contracte, i a partir d'això interpretar els resultats per tal d'aconseguir les dades desitjades. La forma més senzilla d'accedir a aquest registre de transaccions és a través dels exploradors de blocs. Un explorador de blocs és una eina que, tal i com indica el seu nom, permet explorar els blocs de la blockchain, que recordem que manté còpies de totes les transaccions que s'hi executen. En el nostre sistema, l'obtenció d'aquest registre es fa des del *backend*, el qual també és l'encarregat d'interpretar els resultats i emmagatzemar-los a la base de dades. Aquesta peça està desenvolupada amb Javascript, i la part referent a la API utilitza el framework "Node.js".



```

{
  "name": "AddProduct",
  "params": [
    {
      "name": "_sku",
      "value": "monitor27",
      "type": "string"
    },
    {
      "name": "_name",
      "value": "monitor dell 27",
      "type": "string"
    },
    {
      "name": "_description",
      "value": "Dell monitor with 27-inch resolution",
      "type": "string"
    },
    {
      "name": "_qty",
      "value": "6",
      "type": "uint256"
    }
  ],
  "timestamp": "1651851096"
}

```

Figura 14: Exemple de decodificació d'un mètode del contracte

Tot i així, aquesta estratègia només permet obtenir el nom i els paràmetres d'entrada dels mètodes invocats. Això ens limita, per exemple, quan volem també disposar del preu del producte en aquells mètodes on aquest atribut no s'afegeix com a paràmetre. Per a solucionar-ho, s'ha definit un **event** per a cada mètode del contracte, el qual s'emet després de l'execució del mètode corresponent.

**Events:** Els events són membres del contracte que, quan són emesos, emmagatzemen els paràmetres enviats en el registre de transaccions. Tal i com s'explica a [15], aquests registres s'emmagatzemen en la blockchain, de manera que són accessibles a través de l'adreça del contracte. Generalment, aquests events són utilitzats per a informar l'aplicació que utilitza el contracte sobre l'estat d'aquest i, a més, per a mantenir un registre de *logs* del que succeeix en cada invocació.

D'aquesta manera, definint els següents events en el contracte, tenim accés a atributs dels que no disposem si només utilitzem el camp *inputs* de les transaccions:



```

{
  "name": "Add_Product",
  "params": [
    {
      "name": "_sku",
      "type": "string",
      "value": "mouse10"
    },
    {
      "name": "_name",
      "type": "string",
      "value": "Mouse sharkforce"
    },
    {
      "name": "_description",
      "type": "string",
      "value": "Sharkforce mouse size 10"
    },
    {
      "name": "_qty",
      "type": "uint256",
      "value": "45"
    },
    {
      "name": "_price",
      "type": "uint256",
      "value": "20"
    },
    {
      "name": "_requestor",
      "type": "address",
      "value": "0x1a8beb2fffd9ea1731f47d66c7ee0c4692cd08a02"
    }
  ],
  "timestamp": "1653773535"
}

```

Figura 17: Exemple de decodificació dels events emesos en un mètode del contracte

Tal i com es pot veure comparant les Figures 14 i 17, el resultat obtingut és gairebé el mateix, però en el segon cas disposem de més informació sobre la invocació del mètode. Això ens permet oferir consultes més rellevants i útils per a l'usuari.

### 6.2.2 Interpretació de les dades

Una vegada disposem de la informació dels mètodes que s'han executat, amb els paràmetres utilitzats i el seu timestamp, podem emmagatzemar-la en la base de dades per a consultar-la "on demand".

Abans de poder guardar la informació a la base de dades, però, necessitem definir la seva estructura i les taules que contindrà. Tal i com es detalla en la proposta dels "Models Conceptuals Deductius", la base de dades té un canvi de rol important: no guarda l'estat actual de la informació, sinó que guarda els events que succeeixen sobre aquesta. És a dir, en un sistema d'informació convencional, la base de dades té una taula per a cada entitat definida en el diagrama UML, a més d'algunes altres taules que defineixen les relacions entre



aquestes entitats. En canvi, en el cas dels MCD's, la base de dades té una taula per a cadascun dels mètodes que es poden executar. Aquestes taules tenen per columnes, els paràmetres que accepta el mètode, i per files, les invocacions d'aquests mètodes. Addicionalment, les taules també tenen com a columnes alguns atributs afegits en l'event emès després de l'execució del mètode, tal i com s'explica al final de la secció 6.2.1. D'aquesta manera, es poden emmagatzemar totes les accions que es realitzen en el domini sense preocupar-nos de l'estat "actual" de les dades. En el nostre cas, les taules definides són els mètodes del Smart Contract: *AddProduct*, *OrderProduct*, *RemoveProduct*, *ReplenishStock* i *ChangePrice*.

En un domini poc extens, aquesta creació de taules es podria fer de forma manual, però si aquest és gaire extens, crear les taules de forma manual deixa de ser una opció. En el nostre cas, el domini està implementat utilitzant la blockchain d'Ethereum, implementant Smart Contracts en el llenguatge de Solidity. En aquest llenguatge, sempre que es compila un programa (contracte) es genera automàticament un fitxer anomenat "ABI", que ens serveix per a automatitzar aquest procés.

**ABI:** El fitxer "Application Binary Interface" és la interfície que actua com a intermediària entre el contracte (escrit en Solidity) i la "EVM", màquina virtual d'Ethereum (que interpreta en llenguatge màquina). El seu rol és, per una banda, codificar les crides del contracte per a que puguin ser interpretades per la EVM i, per l'altra, poder llegir les dades de les transaccions, les quals s'executen en la EVM (vist a la secció 6.2.1). Així doncs, és l'encarregat de facilitar la comunicació Smart Contract - EVM. Aquesta interfície no deixa de ser un fitxer JSON que defineix tots els mètodes del contracte, especificant per a cadascun, els paràmetres d'entrada i de sortida, i atributs propis de Solidity com "payable" o "constant", entre d'altres. Per exemple, l'objecte que defineix el mètode *AddProduct* del nostre contracte és el següent (Figura 18):

```

{
  "constant": false,
  "inputs": [
    {
      "name": "_sku",
      "type": "string"
    },
    {
      "name": "_name",
      "type": "string"
    },
    {
      "name": "_description",
      "type": "string"
    },
    {
      "name": "_qty",
      "type": "uint256"
    },
    {
      "name": "_price",
      "type": "uint256"
    }
  ],
  "name": "AddProduct",
  "outputs": [],
  "payable": false,
  "stateMutability": "nonpayable",
  "type": "function"
},

```

Figura 18: Exemple de definició del mètode *AddProduct* en el fitxer ABI

Això ens proporciona molta escalabilitat en la creació de taules, ja que podem automatitzar aquest procés iterant sobre el JSON de l'ABI, i creant taules per a cadascun dels mètodes definits. D'aquesta manera es pot evitar la creació de taules de forma manual. És important destacar que aquest procés només s'executa una vegada, durant la preparació i creació del sistema. Una vegada les taules estan creades, ja estan preparades per a rebre la informació.

La inserció de dades en les taules serà, tal i com s'ha mencionat, de cadascuna de les execucions que es fan de cada mètode, en la seva taula corresponent. Seguint el procés definit en la secció 6.2.1, s'obtenen i es decodifiquen tots els events que s'han executat sobre l'Smart Contract, de manera que es disposa de la informació preparada per a ser inserida en la base de dades. Aquest procés està també automatitzat:

- S'itera sobre el JSON obtingut, que conté un *array* amb totes les transaccions del contracte
- Cada objecte de l'*array* disposa del nom del mètode i dels paràmetres amb els que s'ha invocat
- El nom del mètode coincideix amb el nom de la taula, de manera que s'insereixen en les columnes de la taula, els valors passats per paràmetre

Així doncs, un exemple de la taula *AddProduct* després d'aquest procés d'inserció, podria ser el següent:

	_sku text	_name text	_description text	_qty integer	_price integer	_requestor text	timestamp integer
1	mouse10	Mouse sharkf...	Sharkforce mouse size...	45	20	0x1a8beb2ffd9...	1653773535
2	keyboard70	Keyboard new...	Newskill keyboard with...	34	30	0x1a8beb2ffd9...	1653773611
3	monitor27	Monitor dell	Dell monitor with 27-in...	6	160	0x1a8beb2ffd9...	1653773716

Figura 19: Exemple d'estructura i contingut d'una taula de la base de dades

### 6.3 Consulta de dades

Amb tota la informació emmagatzemada a la base de dades, el *middleware* és l'encarregat d'oferir varis *endpoints* en forma d'API REST per a que es puguin mostrar les dades a l'aplicació web. En aquesta, tal i com es menciona anteriorment, està dividida en dues seccions: una per interactuar amb el contracte ("Manage") i una per les consultes de dades ("Explore"). En la segona, es pot escollir entre 4 accions diferents, tal i com es mostra en la Figura 20:



Figura 20: Vista "Explore" de l'aplicació web

Aquestes 4 opcions es poden classificar en 2 grups: les consultes sobre un producte en concret ("Product quantity" i "Product price") i les consultes sobre les transaccions realitzades en el sistema ("Transactions by quantity" i "Transactions by price").

En les vistes que corresponen al primer grup, apareix una barra de búsqueda que conté 2 filtres:

- Product SKU: l'identificador del producte sobre el que es vol realitzar la consulta
- Data: la data sobre la qual es aplicar la cerca

Una vegada l'usuari defineix els dos filtres, es fa una crida a la API per a obtenir la informació desitjada. En el cas de la vista "Product quantity", es mostra el

nom i la quantitat del producte en la data escollida. Pel que fa a la vista "Product price", es mostra el nom i el preu del producte en la data escollida. Un exemple d'aquest tipus de vista es pot veure en la Figura 21, on es mostra la informació del producte "mouse10" en el dia 29/05/2022:

The image shows a search interface with a text input field containing 'mouse10', a date picker set to '29/05/2022', and a 'Search' button. Below the search bar, a box displays the following information:

```
Product info at date 29/5/2022:
Sku: mouse10
Name: Mouse sharkforce
Quantity: 41
```

Figura 21: Vista "Product quantity" de l'aplicació web

Pel que fa a les vistes que corresponen a les consultes sobre transaccions, aquestes contenen una barra de búsqueda amb 4 filtres:

- Estat: que és el tipus de producte sobre el que es vol aplicar la cerca (afegit, venut o reposat)
- Data inici: la data des de la qual es vol aplicar la cerca
- Data fi: la data fins la qual es vol aplicar la cerca
- Ordre: l'ordre en què es volen obtenir els resultats, aplicant-los sobre l'atribut quantitat (de major a menor, de menor a major).

Una vegada l'usuari escull tots els filtres, es fa una crida a la API, que és l'encarregada de traduir-ho a una *query* SQL per a obtenir el resultat desitjat. Aquest resultat conté un llistat dels productes en l'estat escollit, entre les dates definides i ordenades segons el filtre d'ordenació escollit. Un exemple d'aquest tipus de vista es pot veure en la Figura 22, on es mostren els productes venuts durant les dates 01/05/2022 i 02/06/2022, ordenats de major a menor segons el preu:

SKU	Name	Product price	Total price	Date
monitor27	Monitor dell	185	740	May 28, 2022
monitor27	Monitor dell	160	320	May 28, 2022
keyboard70	Keyboard newskill	48	144	May 31, 2022

Figura 22: Vista "Transactions by price" de l'aplicació web

### 6.3.1 Vistes

Les vistes juguen un paper molt important en aquest sistema d'informació, ja que aporten optimització i flexibilitat sobre l'implementació dels models conceptuais deductius. En la nostra base de dades, les dades no estan emmagatzemades com en un sistema d'informació convencional, sinó que les taules contenen informació dels "events" que han succeït, i no de les entitats definides en el domini. Això fa que les consultes SQL que s'han aplicar per a obtenir informació no siguin tant "intuïtives" com són en un SI convencional. Degut a que tenim emmagatzemades les transaccions, és necessari "traduir" aquesta informació als resultats que podríem esperar d'un sistema convencional, on les consultes s'apliquen sobre l'estat actual de les entitats definides.

En el nostre sistema, s'han definit diverses vistes que ens permeten accedir a una versió "adaptada" o "preparada" de les taules definides en la base de dades. D'aquesta manera, es simplifiquen les *queries* que es realitzen per a obtenir la informació de les consultes que ofereix l'aplicació web. Un exemple d'aquestes vistes pot ser la que s'utilitza per a obtenir la quantitat d'un producte en una data determinada. Aquesta, agrupa totes les accions que afecten a la quantitat d'un producte (*AddProduct*, *OrderProduct*, *RemoveProduct* i *ReplenishStock*), per després poder filtrar sobre aquesta i realitzar el càlcul corresponent. A més, és també l'encarregada d'afegir la columna "action", la qual especifica l'acció a la que correspon cada fila, i que ens ajuda a poder realitzar el càlcul de quantitats. Un exemple del contingut d'aquesta vista és el següent:

	sku text	name text	quantity integer	date integer	action text
1	mouse10	Mouse sharkforce	45	1653773535	added
2	keyboard70	Keyboard newskill	34	1653773611	added
3	monitor27	name	4	1653775022	ordered
4	keyboard70	name	0	1653774032	removed
5	keyboard70	name	9	1653774977	ordered
6	monitor27	name	7	1653774077	replenished
7	monitor27	name	2	1653773926	ordered
8	monitor27	Monitor dell	6	1653773716	added
9	mouse10	name	13	1653774932	ordered
10	mouse10	name	7	1653773761	ordered
11	monitor27	name	0	1653774077	removed
12	mouse10	name	0	1653773986	removed
13	mouse10	name	16	1653773986	replenished
14	keyboard70	name	23	1653774032	replenished
15	keyboard70	name	5	1653773821	ordered

Figura 23: Vista "quantities", definida per a facilitar el càlcul de quantitats

A partir d'aquesta vista es pot aplicar una *query* SQL per a filtrar segons l'SKU de producte i la data:

```
SELECT *
FROM quantities
WHERE (date <= ${date} AND sku = ${productSku})
ORDER BY date asc;
```

Figura 24: Query SQL que s'aplica sobre la vista "quantities"

Finalment, ja en el *middleware*, es pot realitzar el càlcul de la quantitat segons les accions que incrementen la quantitat del producte (*AddProduct* i *OrderProduct*) i les que la decremenen (*RemoveProduct* i *ReplenishStock*). El procés que es segueix per a realitzar el càlcul és el següent:

- La quantitat inicial del producte és la corresponent a l'acció *AddProduct*
- Es decremenen la quantitat definida en cada invocació del mètode *OrderProduct*
- S'incrementa la quantitat definida en cada invocació del mètode *ReplenishStock*
- La quantitat passa a ser 0 si el producte és eliminat a través del mètode *RemoveProduct*

## 7 Anàlisi de rendiment

### 7.1 Temps de resposta

A continuació, es fa un anàlisi dels temps de resposta de les funcionalitats del sistema. Com en la resta del projecte, es separa en dues seccions: els càlculs corresponents a les transaccions de la part blockchain, i els de la part de la consulta de dades.

#### 7.1.1 Transaccions blockchain

Per a calcular la duració de les transaccions blockchain, es calcula el temps transcorregut en el *browser* des de la crida del mètode del contracte fins l'obtenció de la resposta de la blockchain. Aquest, és el temps que s'ha d'esperar l'usuari des de la confirmació de la transacció a Metamask fins veure'n la confirmació en l'aplicació web. Pel que fa al temps transcorregut en la blockchain, no es pot consultar el temps exacte que es tarda en confirmar una transacció, ja que aquestes es van afegint al bloc corresponent i, una vegada està ple, s'afegeix a la xarxa. És per això que es considera irrellevant, ja que la transacció acabarà dins del bloc o fora en funció de si s'accepta dins el bloc, però sempre dependrà de la resta de la xarxa i del temps de confirmació del bloc actual. En la taula 22, podem veure els diferents temps de duració de les transaccions blockchain per a cadascun dels mètodes del contracte. Per a obtenir aquests resultats, s'han executat cadascun dels mètodes 3 vegades, i s'ha calculat la mitjana per a obtenir un resultat menys dependent de l'estat de la xarxa.

Mètode	Duració
Add Product	19.267 ms
Order Product	23.657 ms
Replenish Stock	17.864 ms
Delete Product	19.688 ms
Change price	12.874 ms

Taula 22: Temps de resposta de les transaccions blockchain

Com es pot veure en la Taula 22, el temps són força semblants, independentment del mètode executat. Així doncs, el temps mitjà de processament en blockchain obtingut és:

$$T_{blockchain} = \frac{19.267 + 23.657 + 17.864 + 19.688 + 12.874}{5} = 18.670ms \approx 18s$$

Tenint en compte que el temps mitjà esperat per a una transacció en la xarxa Ethereum és de 15 a 20 segons, veiem que els nostres resultats són els esperats. De fet, són resultats força positius, ja que s'ha de tenir en compte, que aquest temps està calculat des del frontend, de manera que es considera també

el temps que tarda l'aplicació web en enviar la petició i en rebre el resultat. De fet, segons l'explorador de xarxa de Rinkeby (<https://www.rinkeby.io/>), el temps de bloc actual es troba en 15,07 segons. Aquest és un temps menor al que havia estat anteriorment, el qual és degut a un menor ús de la xarxa, ja que en les últimes setmanes ha baixat considerablement el nombre total de transaccions Ethereum a causa de l'estat baixista del mercat i el sentiment de por que es troba ara en la majoria d'usuaris. L'índex "Fear and Greed" és un càlcul que mesura el comportament dels mercats borsatils segons el valor de diferents indicadors, com poden ser la volatilitat, l'oferta i demanda, la resistència de preus, etc. Pel que fa al mercat de criptomonedes, també existeix aquest índex. Aquest índex va de 0 (por extrema) a 100 (cobdícia extrema). Com que cadascuna de les xarxes blockchain disposen d'una moneda o "token" natiu, es pot calcular l'índex "Fear and Greed" de la moneda, el qual té un impacte en l'ús de la xarxa. Pel que fa a Ethereum, la xarxa utilitzada en el sistema, el seu índex a data de 4 de juny es situa actualment en 25. També es disposa d'altres indicadors que poden reflectir la baixada en l'ús de la xarxa. En les figures 25 i 26 es mostra, respectivament, el nombre de transaccions mensuals i el nombre d'adreces actives mensuals (dades obtingudes de [18]). Tal i com es pot veure en les dues, els indicadors han estat en descens des del maig de 2021, el qual va ser el mes amb més activitat de la xarxa degut al mercat alcista del moment.

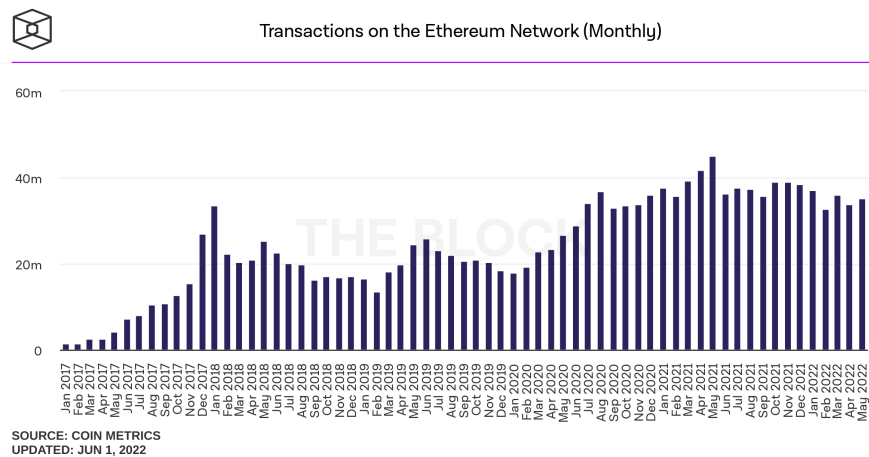


Figura 25: Nombre de transaccions en la xarxa Ethereum (mensual)



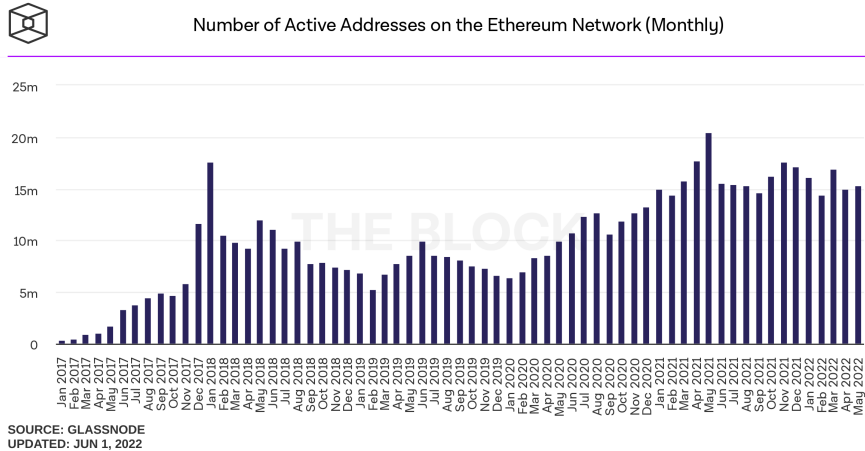


Figura 26: Nombre d'adreces actives en la xarxa Ethereum (mensual)

### 7.1.2 Consulta de dades

En la Taula 23, podem veure els diferents temps de duració de les consultes que ofereix l'aplicació web. És a dir, la taula té una fila per a cadascun dels casos d'ús del tipus "Consulta de dades" definits en la secció 5.2.2. Pel que fa a les columnes, la primera fa referència al temps que tarda en processar-se la consulta en la base de dades. Aquest temps s'obté fent la suma dels valors "Planning time" i "Execution time" que retorna la sentència EXPLAIN. El primer, representa el temps que tarda el motor de la BD en buscar la forma més ràpida d'executar la consulta, i el segon és el temps que suposa l'execució de la consulta com a tal. És a dir, s'inverteix un temps previ per tal d'optimitzar la consulta i poder reduir posteriorment el temps que tarda en executar-se.

Pel que fa la segona columna, aquesta fa referència al temps transcorregut en el *browser* des de la crida del *endpoint* corresponent fins l'obtenció de la resposta de la API. Aquest, és el temps que s'ha d'esperar l'usuari des de la petició de la cerca fins veure'n en resultat en l'aplicació web.

Mètode	T Database	T Server
Consultar quantitat d'un producte en una data	0,249 ms	207,281 ms
Consultar preu d'un producte en una data	0,16 ms	233,175 ms
Consultar productes afegits segons quantitat	0,294 ms	86,885 ms
Consultar productes venuts segons quantitat	0,189 ms	244,749 ms
Consultar productes reposats segons quantitat	0,269 ms	110,785 ms
Consultar productes afegits segons preu	0,373 ms	97,876 ms
Consultar productes venuts segons preu	0,176 ms	92,123 ms
Consultar productes reposats segons preu	0,224 ms	94,97 ms

Taula 23: Temps de resposta de les consultes de dades

Com es pot veure en la taula 23, els temps es mantenen força iguals en els dos casos. El temps mitjà de processament en la base de dades és:

$$T_{database} = \frac{0,24 + 0,16 + 0,29 + 0,18 + 0,26 + 0,37 + 0,17 + 0,22}{8} = 0,242ms$$

I el temps mitjà de processament en el servidor és:

$$T_{server} = \frac{207,2 + 233,1 + 86,8 + 244,7 + 110,7 + 97,8 + 92,1 + 94,9}{8} = 1.167,814ms$$

Aquests són temps molt baixos, però no es pot assumir que sempre serà així. El contingut actual de la base de dades és molt reduït, de manera que les *queries* tarden molt poc en processar-se. A més, la majoria de les consultes dissenyades utilitzen sentències JOIN, les quals incrementen el temps de processament de forma exponencial quan augmenta la quantitat de dades en les taules consultades.

## 7.2 Possibles optimitzacions

En l'apartat anterior (7.1) queda reflectit que el major cost temporal es troba en la part blockchain. Una possible optimització seria reduir aquests temps per tal d'agilitzar el procés de gestió, el qual resulta menys còmode per a l'usuari en comparació amb el procés de consulta de dades. Degut a que el temps de resposta de la blockchain depèn exclusivament de la velocitat i el rendiment de la xarxa, la solució en aquest cas seria desplegar el sistema en una xarxa diferent a la que s'està utilitzant actualment. Una de les millors opcions és la xarxa Avalanche, la qual està basada en una arquitectura EVM i permetria el desplegament dels contractes desenvolupats sense haver de modificar-los ni traduir-los. Pel que fa a la interacció amb Metamask, també es pot integrar sense un cost més elevat del que suposa fer-ho en Ethereum. El temps de bloc mitjà en la xarxa Avalanche és d'aproximadament 1 segon [11], el qual és una millora molt substancial considerant que els temps obtinguts del sistema en la xarxa Ethereum han sigut, de mitjana, de 18 segons.

Una altra millora que es podria aplicar en el sistema és en el procés d'inserció de les transaccions en la base de dades. Aquest procés, ara mateix, es realitza amb un *script* que s'encarrega d'obtenir les transaccions del contracte en la blockchain, decodificar-les i preparar les *queries* INSERT per a les taules de la base de dades amb la informació obtinguda. La part optimitzable és que, actualment, no es té en compte quines transaccions han sigut ja inserides i quines no, de manera que el procés d'obtenció i d'inserció es realitza sempre per a totes les transaccions del contracte. Una possible millora seria mantenir un registre de quina ha sigut la última transacció inserida en la base de dades, de manera que quan es vol actualitzar el contingut de les taules amb les transaccions corresponents, només s'interpretin i s'afegeixin aquelles que encara no han estat inserides prèviament.

## 8 Conclusions

### 8.1 Resultat final

A continuació es fa una valoració general del sistema desenvolupat, resumint primer l'arquitectura final que s'ha acabat utilitzant i després entrant en detall en els punts forts i els punts febles del sistema.

#### 8.1.1 Solució desenvolupada

Aquest treball es tracta d'un projecte d'investigació i exploració, que té com a objectiu implementar els Models Conceptuals Deductius utilitzant blockchain, degut a les semblances que presenta aquest concepte amb el funcionament d'aquesta tecnologia. Així doncs, després d'haver construït el sistema resultant, es pot extreure un resum de l'arquitectura final i de la manera en què es connecten cadascun dels elements que la formen. A continuació es descriu el procés a seguir per a construir aquest sistema, com si es tractés d'un manual d'instruccions per si es volgués replicar:

En primer lloc, es defineix el domini a través de diagrames UML i diagrames de casos d'ús, que faciliten la descripció de tots els escenaris que es poden donar en el sistema i així poder considerar-los en l'etapa de desenvolupament. Una vegada es disposa d'un domini ben definit, es pot procedir amb el desenvolupament de la peça central referent a la part de gestió: els Smart Contracts. Si es vol desplegar el sistema en una xarxa EVM, aquests contractes són desenvolupats amb Solidity. En aquests contractes es defineixen les estructures de dades i les funcionalitats de gestió del sistema. En aquests mètodes es poden afegir, després de la seva funció corresponent, la crida d'un event que té com a paràmetres tota aquella informació addicional que no forma part dels paràmetres del mètode i que pot ser d'interès per a la seva futura consulta. Després del testeig de totes les funcionalitats del contracte en un entorn local, es pot desplegar en una *testnet* per a facilitar-ne la comunicació amb el l'aplicació web. En aquest projecte, el desplegament s'ha realitzat a través d'un *script* escrit en Javascript que utilitza la llibreria "Hardhat", la qual agilitza tot aquest procés. Tant aquest *script* com el resultat que s'obté després de la seva execució es poden veure a l'Annex (figures 28 i 29).

A continuació es pot procedir amb el desenvolupament de la part de gestió de l'aplicació web. En aquest projecte s'ha utilitzat el *framework* Angular, tot i que aquesta decisió està a mans del desenvolupador. En aquesta aplicació web es creen vistes per a cada funcionalitat de gestió, amb un formulari per tal de que l'usuari introdueixi la informació necessària per l'execució de cada mètode, de manera que quan s'envia el formulari es crida el mètode del contracte. La validació i el pagament del *gas* de la transacció es fa a través de Metamask, de manera que s'ha d'integrar el suport d'aquest *plugin* des del frontend. Per a aquesta integració, en aquest projecte s'ha utilitzat la llibreria "Ethers.js", tot i que es poden utilitzar altres, com pot ser la popular "Web3.js".

Una vegada connectats els contractes amb l'aplicació web, es pot procedir amb la construcció del *middleware*, el qual s'ha desenvolupat utilitzant Node.js. Es considera *middleware* perquè, a més d'oferir una API REST que permet consultar informació, té també la responsabilitat d'obtenir les transaccions de la blockchain i inserir-les en la base dades. Aquest procés està descrit en detall en l'apartat 6.2.1, però el seu objectiu és obtenir i decodificar les transaccions d'una API que facilita la consulta de dades de la blockchain escollida. En aquest projecte s'ha utilitzat la API d'Etherscan per a l'obtenció i la llibreria "ABI Decoder" per a la decodificació. Aquest procés permet inserir en la base dades totes les transaccions que succeeixen en el contracte, de manera que després es poden consultar "on demand" des de l'aplicació web. Per a la connexió del *middleware* amb la base de dades, s'ha utilitzat la llibreria "pg-node".

La base de dades té una taula per a cadascun dels mètodes del contracte, de manera que es poden emmagatzemar les execucions dels mètodes amb els paràmetres utilitzats. En aquest projecte s'ha utilitzat PostgreSQL, tot i que aquesta decisió recau també en el desenvolupador. Tant la creació de taules i la inserció de les transaccions es fa a través d'uns *scripts* que estan també en el *middleware* per si es volen executar des de l'aplicació web.

Quan la base de dades ja conté tota la informació, està preparada per a rebre les consultes de dades. Aquestes es defineixen segons els casos d'ús definit, i es poden crear noves consultes al cap del temps si apareixen nous casos d'ús en el domini del sistema. Primer es dissenyen les *queries* SQL i, un cop estan testejades, es poden traslladar a la API REST. En aquesta, s'ofereixen un seguit d'*endpoints* que permeten fer les consultes de dades des de l'aplicació web.

Finalment, en l'aplicació web es crea la segona part, corresponent a la consulta de dades. Les vistes dedicades a aquesta secció contenen una barra de cerca que permet a l'usuari seleccionar els filtres corresponents, els quals solen estar relacionats amb les dates o l'estat de les dades que es volen consultar. Un cop l'usuari fa la cerca, es crida a l'*endpoint* corresponent, el qual afegeix els filtres a la *query* i retorna el resultat. Aquest resultat es mostra en l'aplicació web de forma entenedora per a l'usuari, ja sigui en format de "card", de "llistat", etc.

### 8.1.2 Punts forts del sistema

Els beneficis més rellevants que proporciona l'ús d'aquest sistema són:

- Trasllat de l'etapa de disseny: tal i com s'explica a 1.1.1, un dels punts forts de la implementació dels Models Conceptuals Deductius és la flexibilitat que aquests aporten en l'etapa de disseny del model de dades i de les consultes corresponents. En una arquitectura convencional, aquest procés de disseny s'ha de dur a terme abans de l'etapa de desenvolupament, moment en el qual no es coneixen en detall tots els casos d'ús respecte les consultes de dades que tindrà el sistema. Poder registrar totes les transaccions i aplicar-ne les consultes posteriorment, permet al

desenvolupador traslladar la tasca de disseny a una etapa més avançada del projecte, on es coneixen ja amb més detall les necessitats referents a la consulta de dades. A més, aquesta característica s'alineja amb els principis de les metodologies àgils, les quals prioritzen el desenvolupament de productes funcionals en trams curts per sobre d'etapes llargues de disseny i inepció.

- Traçabilitat: el fet d'enregistrar totes les transaccions que s'executen en el sistema ens permet consultar en qualsevol moment tots els events que han succeït, a més de poder monitoritzar totes les accions que es produeixen. Això proporciona transparència, ja que es pot demostrar la veracitat i la immutabilitat de les dades.
- Business intelligence (BI): com que es poden consultar tots els events que han succeït en el sistema, aquestes dades es podem transformar en informació rellevant sobre el domini aplicat, de manera que es pot fer un major anàlisi en profunditat i un ampli *reporting*. La necessitat d'obtenir mètriques i anàlisis sobre les dades és cada vegada més important per a les empreses, de manera que aquest sistema pot proporcionar un valor afegit molt positiu. Una empresa podrà analitzar, per exemple, l'impacte que té la implementació d'un nou mètode o la modificació d'un procés que s'ha decidit modificar. Això permet, després, saber si aquests canvis han sigut efectius o no, i poder mantenir-los o canviar-los segons si els resultats han estat els esperats. Un altre avantatge important que proporciona aquest anàlisi de dades és la capacitat de realitzar projeccions i pronòstics de futur en base a tota la informació que s'ha enregistrat fins aleshores.
- Millora en l'ambigüitat de les dades: el fet de poder accedir a transaccions passades també dona peu a una possible ambigüitat de dades, ja que si algun dels atributs d'un objecte ha canviat de valor, és important saber quin era el valor d'aquell atribut en el moment de la transacció en qüestió. Per exemple, si volem consultar la venda d'un producte que es va fer 6 mesos enrere, cal saber quin era el preu del producte en aquell moment, i no confondre-ho pel preu que pugui tenir en l'actualitat, ja que el valor d'aquest atribut pot haver canviat. Gràcies a l'enregistrament de totes les transaccions, tenim constància del valor que tenia l'atribut en aquell moment, i així evitar aquest problema d'ambigüitat i irregularitat de dades.

### 8.1.3 Punts febles del sistema

Pel que fa als problemes més destacats de l'ús d'aquest sistema són:

- Temps d'espera elevats: els temps d'espera de la part de "gestió", que és la part en la que intervé la blockchain, són força alts per a un usuari, i més tenint en compte la velocitat de resposta de la majoria d'aplicacions en l'actualitat. Tot i així, podem considerar que això és el "preu a pagar"

a canvi de tots els avantatges que ens proporciona l'ús de la blockchain i l'arquitectura dissenyada.

- Cost elevat del disseny de les consultes de dades: la informació emmagatzemada en la base de dades es troba en format “event” o “mètode” executat, tal i com s'explica a 6.2.2. El cost d'esforços que suposa el disseny de les consultes per tal de convertir aquestes dades en informació rellevant és força elevat, ja que és poc intuïtiu i les consultes són poc reusables. Això és degut a que cada consulta és diferent, ja que no es disposa de taules que continguin la informació actual del sistema, sinó de les transaccions executades. En la majoria de consultes és necessari fer JOINS de les taules per a obtenir la informació desitjada, fet que provoca un augment de complexitat de les consultes.

Tot i així, actualment les empreses dediquen molts recursos en el Business Intelligence, degut al valor que aporta l'anàlisi de dades per millorar els processos i poder fer prediccions de comportaments futurs. És per això que assumir els costos que suposen aquestes consultes de dades és una inversió per a obtenir informació rellevant.

## 8.2 Propostes de futur

A continuació es llisten algunes de les propostes que es podrien implementar en un futur i que aportarien millores en el sistema:

- Nous casos d'ús: es poden definir molts més casos d'ús, especialment pel que fa a les de consultes de dades. Tenint en compte que es disposen d'atributs com l'adreça que ha invocat el mètode o el preu del producte en cadascuna de les accions, aquestes noves consultes podrien ser:
  - Total de vendes en un període: definint una data d'inici i una data fi, es podrien obtenir totes les vendes, mostrant el preu total de la comanda i, fins i tot, amb la possibilitat d'ordenar segons el preu total.
  - Accions realitzades per un usuari: definint una *address* i, opcionalment, una data d'inici i una data fi, es podrien obtenir totes les accions que ha realitzat aquella adreça en el període definit.
- Gestió de rols: es poden establir diversos rols en el sistema, els quals permeten controlar qui pot portar a terme cada acció. El contracte desenvolupat disposa d'un *modifier*, el qual es pot aplicar a cadascun dels mètodes perquè faci la comprovació corresponent. Si es defineixen, per exemple, els rols “treballador” i “administrador”, es podria permetre que el treballador reposi estoc, però no que modifiqui el preu d'un producte, a diferència d l'administrador, que sí que ho tindria permès.
- Ús d'una blockchain privada: utilitzar una blockchain privada com Hyperledger podria proporcionar privacitat, rendiment i escalabilitat al sistema.

Aquesta xarxa destaca per tenir rols, de forma que l'ús dels contractes està protegit, com també ho està la consulta de registres de transaccions. Està enfocada a aplicacions per a empreses, i facilita eines com la gestió de rols o els anomenats "canals". A més, està construïda en un disseny modular, de forma que es poden dividir els processos i d'aquesta forma es millora l'escalabilitat del sistema.

- Canvi de domini: en aquest projecte s'ha utilitzat un exemple senzill per al domini, per tal de poder centrar els esforços en el la construcció del sistema i el seu correcte funcionament. Una proposta de futur és implementar un domini més complex per a veure i analitzar com es comporta el sistema en un escenari diferent, amb més entitats de dades, més casos d'ús, més transaccions, etc. Un exemple d'aquest domini podria ser l'utilitzat per Antoni Olivé en el seu article "El mètode dades" [3], el qual tracta sobre la gestió de ponències d'una conferència. La implementació d'aquest exemple suposaria un desenvolupament per complet dels Smart Contracts, a més del disseny de totes les consultes de dades en la base de dades. L'aplicació web s'hauria d'adaptar a aquests canvis, i pel que fa al *backend*, només s'haurien de crear els *endpoints* corresponents, ja que es podria reutilitzar tota la part actual d'obtenció i interpretació de dades. Tot i així, es pot trobar una explicació detallada del procés de replicació del sistema a la secció 8.1.1.

### 8.3 Integració de coneixements

S'han integrat, principlament, coneixements de l'àmbit de l'enginyeria del software. D'aquests, una part han estat obtinguts a través de les assignatures cursades a la universitat (secció 8.3.1). L'altra part, han estat coneixements que s'han obtingut en el desenvolupament del projecte (secció 8.3.3).

Pel que fa a les competències tècniques assolides en el projecte, aquestes es detallen en la taula 24.

#### 8.3.1 Coneixements aplicats d'altres assignatures

Pel que fa als coneixements obtinguts en les assignatures, podem destacar els següents:

- Disseny UX/UI i desenvolupament de l'aplicació web, juntament amb la seva integració amb el sistema backend. Coneixement obtingut en les assignatures de ASW i PES. També en les assignatures de UID i WEB-DEV, cursades en la meua estada de mobilitat a la universitat de Fontys (Eindhoven).
- Disseny i contrucció de la base de dades, juntament amb l'elaboració de les taules, les vistes i les consultes SQL. Instal·lació d'una base de dades PostgreSQL i d'un SGBD per a gestionar-la. Coneixement obtingut en les assignatures de BD i DBD.



- Disseny i construcció d'una API REST, encarregada de comunicar-se tant amb la base de dades com amb l'aplicació web. Coneixement obtingut en les assignatures de ASW i PES.
- Ús de metodologies àgils. Coneixement obtingut en l'assignatura de GPS.
- Definició de requisits i casos d'ús. Coneixement obtingut en l'assignatura de ER.

### 8.3.2 Grau de satisfacció de les competències tècniques

En la Taula 24, es detalla el grau de satisfacció i de quina manera intervé en el projecte cadascuna de les competències tècniques definides:

Codi	Nom	Grau	Intervenció
CES1.1	Desenvolupar, mantenir i avaluar sistemes i serveis software complexes i/o crítics	Bastant	El software que s'ha desenvolupat no segueix una arquitectura convencional, i el fet d'explorar una nova manera d'implementar un sistema d'informació fa que aquest projecte sigui complex i crític. Pel que fa al manteniment i l'evaluació, són aspectes en els que no es profunditza molt, degut a la curta durada del projecte. Si el sistema s'hagués de convertir en un producte real, s'hauria de profunditzar més en el manteniment per a garantir-ne una llarga vida útil.
CES1.2	Donar solució a problemes d'integració en funció de les estratègies, dels estàndards i de les tecnologies disponibles	En profunditat	L'objectiu principal d'aquest sistema es pot considerar que tracta de solventar un problema d'integració: poder implementar els models conceptuals deductius, els quals ens proporcionen una sèrie d'avantatges, i a través de la blockchain, una tecnologia que no existia quan va aparèixer el concepte dels MCD's.

CES1.3	Identificar, evaluar i gestionar els riscos potencials associats a la construcció de software que puguin presentar-se	Poc	La desconexió respecte la tecnologia blockchain i la poca maduresa d'aquesta, són aspectes que fan que puguin aparèixer riscos potencials per al sistema: problemes de seguretat, de robustesa, etc. Poder identificar-los i actuar al respecte pot evitar possibles problemes en un futur.
CES1.4	Desenvolupar, mantenir i evaluar serveis i aplicacions distribuïdes amb suport de xarxa	Bastant	Tot el sistema desenvolupat recau en l'ús de la xarxa, no hi ha cap procés que pugui fer-se sense connexió a Internet.
CES1.5	Especificar, dissenyar, implementar i evaluar bases de dades	En profunditat	Una de les peces més importants d'aquest projecte és la base de dades. És l'epicentre de tota l'activitat que succeeix en el sistema, ja que és l'encarregada tant d'emmagatzemar totes les transaccions, com de traduir aquesta informació en dades rellevants per a l'usuari. Un bon disseny i implementació, tant de les taules com de les vistes i les consultes, és fonamental per al bon funcionament del sistema.
CES1.6	Administrar bases de dades	Bastant	Tal i com es menciona en la competència CES1.5, la base de dades és fonamental en aquest projecte. D'aquesta manera, és necessari fer-ne una bona gestió i administració.
CES1.7	Controlar la qualitat i dissenyar proves en la producció de software	En profunditat	Una de les últimes tasques del projecte consisteix en l'obtenció i l'anàlisi de resultats per a mesurar la qualitat del software desenvolupat, a més de poder aplicar millores i optimitzacions en aquells processos que no presentin un bon rendiment.

CES1.8	Especificar, dissenyar i implementar sistemes de control i de temps real	Poc	En la part blockchain, s'utilitzen els events per a tenir constància dels mètodes que s'executen i per a poder tenir accés a determinats atributs que no es poden consultar a través del <i>input</i> de la transacció. Tot i així, no s'utilitzen com a <i>trigger</i> per a executar accions de resposta a temps real.
CES1.9	Demostrar comprensió en la gestió i govern dels sistemes software	Poc	La complexitat de l'arquitectura proposada en el sistema demostra comprensió en la gestió de sistemes software.
CES2.1	Definir i gestionar els requisits d'un sistema software	Bastant	Degut a que aquest és un projecte de caire exploratori, és complicat definir exactament quins són els objectius i els requisits que es volen assolir. Definir-los prèviament aporta claredat en la direcció que ha de seguir el sistema durant el seu desenvolupament. No fer-ho pot suposar demores en el desenvolupament, ja que és un error comú avançar cap a una direcció equivocada i haver de tirar enrere, fet que suposa una pèrdua de temps i de recursos.
CES2.2	Dissenyar solucions apropiades en un o més dominis d'aplicació, utilitzant mètodes d'enginyeria del software que integren aspectes ètics, socials, legals i econòmics	Bastant	Un dels beneficis que aporta la tecnologia blockchain és la transparència. Totes les transaccions queden registrades, de forma que es pot assolir un control molt precís de tot el que succeeix en el sistema. Això elimina la necessitat de tercers, ja que és la pròpia tecnologia la que garanteix la veracitat de les dades.

CES3.2	Dissenyar i gestionar un magatzem de dades (data warehouse)	Bastant	Igual que en les competències CES1.5 i CES1.6, l'emmagatzematge de dades és el nucli del sistema. S'ha dedicat un gran nombre d'hores en el disseny i la implementació d'aquest aspecte per a assegurar que compleix el funcionament desitjat.
--------	---	---------	--

Taula 24: Satisfacció de competències tècniques

### 8.3.3 Aprenentatges obtinguts

A nivell teòric, l'aprenentatge principal ha sigut el concepte dels Models Conceptuals Deductius. Aquest era un concepte totalment desconegut per a mi, ja que no ens l'havien mencionat en cap assignatura anterior. Quan el director d'aquest projecte, en Joan Antoni Pastor, em va proposar l'aplicació d'aquest model amb la tecnologia blockchain, em va semblar la oportunitat perfecta per a indagar en aquest concepte.

A nivell tècnic, tal i com s'explica en la secció anterior (8.3.1), conceptes com l'aplicació web, la API REST i la base de dades ja els havia tractat en assignatures anteriors. Així doncs, l'aspecte més desconegut per a mi ha sigut el desenvolupament de la part blockchain. Aquesta és una tecnologia que vaig conèixer en l'assignatura de GCS, però només des d'un punt de vista teòric. Em va cridar molt l'atenció, i vaig decidir investigar sobre la part tècnica: com s'utilitza, com s'hi desenvolupa, com són les aplicacions que s'hi construeixen, etc. Aquest treball ha sigut el motiu perfecte per a poder dedicar una part del meu temps en entendre i provar, de primera mà, com és l'ús d'aquesta tecnologia des del punt de vista d'un programador. Ara, una vegada acabat, puc considerar que he après varis conceptes sobre aquesta nova però tant àmplia tecnologia: l'ús de xarxes locals i de xarxes *testnet*, el desenvolupament dels Smart Contracts en Solidity, l'obtenció de dades de la blockchain, la interpretació i decodificació de la informació sobre les transaccions, el desplegament de contractes en les xarxes EVM o la integració d'aplicacions web a través de Metamask. Tot i així, també he après nous conceptes en les altres parts del sistema, com pot ser el *profiling* de les consultes SQL per a obtenir informació sobre la seva execució, o l'ús de *pools* de connexions en la API REST per a establir múltiples connexions simultànies amb la base de dades.

Pel que fa als aprenentatges menys relacionats amb el sistema, sinó amb el projecte en si, destaquen: l'ús de Latex per al desenvolupament de la memòria, la gestió de referències bibliogràfiques amb Mendeley, el citat de referències en format APA, les estimacions de pressupost d'un projecte informàtic, els càlculs de desviacions de temps o l'anàlisi de l'impacte econòmic, mediambiental i social d'un projecte TIC.

## 9 Desviaments i imprevistos

### 9.1 Variacions respecte la planificació inicial

Durant el procés de desenvolupament del projecte he decidit modificar la planificació inicial, ja que aquesta es centrava molt en la part de desenvolupament i no assignava temps suficient a l'anàlisi dels resultats obtinguts. És per això que he decidit primer centrar-me en un domini poc ampli, que permet assegurar el funcionament de totes les peces del sistema i una bona comunicació entre elles. Aquesta primera part s'assigna doncs als primers dos sprints, de manera que el sistema queda ja construït, en forma d'MVP, al final d'aquesta etapa. Pel que fa al tercer sprint, la planificació establerta queda de la següent manera:

En primer lloc, es documenta tota la part desenvolupada fins al moment, detallant tota l'estructura del sistema i el funcionament d'aquest. Després, s'amplia el domini per tal d'aconseguir un abast més gran i que d'aquesta manera permet l'aparició de nous casos d'ús interessants per al projecte. Això pretén enriquir el contingut del treball i demostrar la seva utilitat en un exemple més elaborat. Aquests canvis s'afegeixen tant en la part blockchain (Smart Contracts) com en la part de les bases de dades relacionals i els middlewares corresponents. Pel que fa a les vistes i les consultes SQL, s'exploraran altres variacions de les actuals, que puguin oferir més flexibilitat i escalabilitat. Per últim, es fa l'anàlisi i documentació dels resultats obtinguts, fet que permet poder obtenir conclusions. L'objectiu d'aquest últim període és concloure si el sistema construït és o no un sistema d'informació funcional en un escenari real, tenint en compte els seus punts forts i els seus punts febles.

La taula següent detalla la diferència entre les hores estimades i les hores reals d'aquelles tasques que s'han vist afectades per algun dels canvis de planificació aplicats:

Codi	Tasca	Temps estimat	Temps real
DEV11	Obtenció de resultats	0	5
RC3	Anàlisi de resultats	0	10
DEV3	Desenvolupament dels Smart Contracts	35	38
DEV4	Desenvolupament de l'aplicació	35	38
DEV10	Mostrat de dades en l'aplicacio web	20	25
DEV7	Disseny i creacio de les taules	30	35
DEV5	Interacció amb els Smart Contracts	30	26
CN2	Desplegament dels Smart Contracts	15	8
CN3	Desplegament de la base de dades	10	5

Taula 25: Diferències de costos temporals causats pels canvis de planificació

Com es pot observar a la Taula 25, s'han hagut d'afegir les tasques corresponents a l'obtenció i l'anàlisi de resultats (DEV11, RC3) les quals han suposat un increment de 15 hores sobre el temps total. Degut a l'ampliació del domini al

tercer sprint, les tasques de DEV3, DEV4 i DEV10 han suposat un cost major de l'estimat, ja que han hagut de ser complementades quan ja es consideraven acabades. Pel que fa a la creació de taules (DEV7), aquesta estava pensada per a ser una tasca manual, ja que en el moment de l'estimació no s'havia fet encara el disseny de les taules ni es coneixia en detall la manera en què es faria aquesta creació. Durant el desenvolupament d'aquesta tasca es va veure la possibilitat d'automatitzar aquest procés, tal i com s'explica a 6.2.2, i això va suposar un increment del cost temporal que s'havia previst per aquesta tasca. Tot i així, degut al trasllat de moltes tasques del tercer sprint al segon, s'ha hagut de reduir el cost temporal d'algunes d'elles (DEV5, CN2 i CN3) per a poder tenir un MVP funcional al final del segon sprint. Això ha equilibrat l'augment del cost temporal causat per les tasques d'anàlisi de resultats i d'ampliació de domini, de manera que el cost total del projecte ha sigut molt semblat al previst inicialment.

## 9.2 Variacions respecte el pressupost inicial

Pel que fa als costos econòmics, aquests només s'han vist afectats pels canvis de planificació, els quals han causat un lleuger augment d'hores. Tal i com s'explica en la secció anterior (9.1), s'han hagut d'afegir les tasques DEV11 i RC3, a més d'augmentar el cost temporal de les tasques DEV3, DEV4 i DEV10:

$$\text{Increment d'hores} = (5 + 10 + (38 - 35) + (38 - 35) + (25 - 20)) = 21 \text{ hores}$$

Tot i així, també hi ha hagut una reducció d'hores en les tasques DEV5, CN2 i CN3:

$$\text{Decrement d'hores} = ((30 - 26) + (15 - 8) + (10 - 5)) = 16 \text{ hores}$$

Així doncs, la desviació total d'hores queda amb un increment de només **5 hores**, el qual es tradueix en un petit increment del pressupost. Tenint en compte que totes les tasques afectades pertanyen al rol DEV, el càlcul de l'increment de pressupost és:

$$\text{Desviació de costos} = (5 * 21,1) = 105,5 \text{ €}$$

Tot i així, aquest augment de costos no té cap implicació respecte el pressupost, ja que en el pressupost inicial s'havien destinat 633€ per als costos d'imprevistos del rol DEV, xifra que cobreix perfectament el petit desviament obtingut.

## 10 Annex

A continuació es llisten aquells continguts addicionals que no s'han volgut incloure en la part principal de la memòria però que no es volien excloure del projecte:

1. Enllaç al contracte desenvolupat en Solidity (*Inventory.sol*):  
`https://gist.github.com/aniol-carbo/a53f372b85c880be48c5448694903f93`
2. Enllaç al repositori del frontend:  
`https://github.com/aniol-carbo/mcd-frontend.git`
3. Enllaç al repositori del backend:  
`https://github.com/aniol-carbo/middleware.git`
4. Vista de Metamask que mostra els paràmetres de configuració de la xarxa Rinkeby:

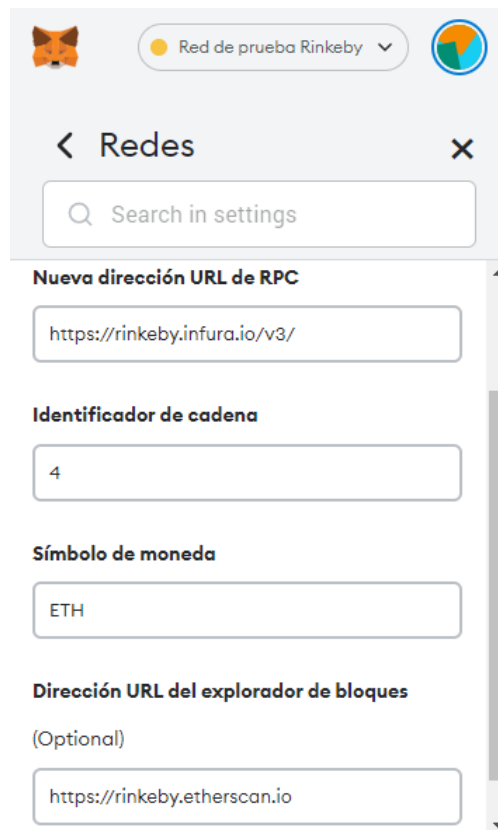


Figura 27: Paràmetres de configuració de la xarxa Rinkeby

5. Script de desplegament de contractes:

```
const main = async () => {
  const [deployer] = await hre.ethers.getSigners();
  const accountBalance = await deployer.getBalance();

  console.log("Deploying contracts with account: ", deployer.address);
  console.log("Account balance: ", accountBalance.toString());

  const smartContractFactory = await hre.ethers.getContractFactory("Inventory");
  const smartContract = await smartContractFactory.deploy();
  await smartContract.deployed();

  console.log("Deployed! Contract address: ", smartContract.address);
};

const runMain = async () => {
  try {
    await main();
    process.exit(0);
  } catch (error) {
    console.log(error);
    process.exit(1);
  }
};

runMain();
```

Figura 28: Script de desplegament de contractes

6. Output que genera l'execució del *script* de desplegament

```
Compiled 1 Solidity file successfully
Deploying contracts with account: 0x1A8bEb2ffD9Ea1731F47D66c7ee0c4692cd08A02
Account balance: 59087647767843151
Deployed! Contract address: 0x6a64a57f7a9496e09e52d580Ee725c3724FC6054
```

Figura 29: *Output* d'èxit del *script* de desplegament d'un contracte

7. Detall de tots els resultats obtinguts en l'execució de les funcionalitats de gestió en l'aplicació web:



```
addProduct: 32721.82080078125 ms
changePrice: 8665.52490234375 ms
orderProduct: 28533.26806640625 ms
orderProduct: 20474.360107421875 ms
deleteProduct: 28417.60791015625 ms
replenishStock: 8425.615234375 ms
addProduct: 12441.77099609375 ms
changePrice: 16402.761962890625 ms
replenishStock: 16428.653076171875 ms
replenishStock: 28739.81591796875 ms
deleteProduct: 18209.45387921095 ms
changePrice: 13557.049072265625 ms
orderProduct: 21964.633056640625 ms
addProduct: 12641.893310546875 ms
deleteProduct: 12438.426025390625 ms
```

Figura 30: Temps de resposta obtinguts en la blockchain

8. Detall de tots els resultats obtinguts en l'execució de les funcionalitats de consulta de dades en la base de dades:

```
Get product quantity:
"Planning Time: 0.024 ms"
"Execution Time: 0.225 ms"

Get product price:
"Planning Time: 0.039 ms"
"Execution Time: 0.121 ms"

Get sold items by quantity:
"Planning Time: 0.090 ms"
"Execution Time: 0.204 ms"

Get added items by quantity:
"Planning Time: 0.051 ms"
"Execution Time: 0.138 ms"

Get replenished items by quantity:
"Planning Time: 0.064 ms"
"Execution Time: 0.205 ms"

Get sold items by price:
"Planning Time: 0.135 ms"
"Execution Time: 0.238 ms"

Get added items by price:
"Planning Time: 0.032 ms"
"Execution Time: 0.144 ms"

Get replenished items by price:
"Planning Time: 0.113 ms"
"Execution Time: 0.111 ms"
```

Figura 31: Temps de resposta obtinguts en la base de dades

9. Detall de tots els resultats obtinguts en l'execució de les funcionalitats de consulta de dades en el *server*:

```
getProductQuantityFromDb: 207.281ms
getProductPriceFromDb: 233.175ms

getSoldItemsByQuantityFromDb: 244.749ms
getAddedItemsByQuantityFromDb: 86.885ms
getReplenishedItemsByQuantityFromDb: 110.785ms

getSoldItemsByPriceFromDb: 92.123ms
getAddedItemsByPriceFromDb: 97.876ms
getReplenishedItemsByPriceFromDb: 94.97ms
```

Figura 32: Temps de resposta obtinguts en el *server*

## 11 Referències

### 11.1 Bibliografia

- [1] COSTAL, D. (1992). *A Method for Reasoning about Deductive Conceptual Models of Information Systems*. Universitat Politècnica de Catalunya.
- [2] Gestió de projectes (GEP). Encaje Gantt con Metodologías Ágiles. In *Mòdul 2.3 - Gestión del tiempo* (pp. 15–19). Facultat d'Informàtica de Barcelona.
- [3] Olivé, A. (1983). *El mètode dades*. Novatica, IX, 35–41.
- [4] Olivé, A. (1989). *On the design and implementation systems from deductive conceptual models*. Universitat Politècnica de Catalunya.
- [5] Quer, C. (1993). *Object Interaction Object-Oriented Deductive Conceptual Models*. Universitat Politècnica de Catalunya.
- [6] Sistac Planas, J. (1987). *Construcció automàtica de prototipus de sistemes d'informació a partir de models conceptuals deductius amb el llenguatge DADES*. Universitat Politècnica de Catalunya.
- [7] Sistac & Olivé (1990). *DADES/GP: A prototype generator from deductive conceptual models of information systems*. Facultat d'Informàtica. Universitat Politècnica de Catalunya.

### 11.2 Webgrafia

- [8] Aguado, M. G. (2020, February 13). *Los bitcoins contaminan: impacto medioambiental de las criptomonedas*. Ethic. <https://ethic.es/2020/02/bitcoins-el-supervillano-virtual-del-medio-ambiente/>
- [9] Atanda, D. (2022, May 10). *Understanding and resolving MetaMask error codes*. LogRocket Blog. <https://blog.logrocket.com/understanding-resolving-metamask-error-codes/>
- [10] BBVA. (2019, March 1). *Metodología “scrum” en “Agile”: ¿Qué es un “sprint” y cuánto dura?* BBVA. <https://www.bbva.com/es/metodologia-scrum-que-es-un-sprint/>
- [11] Covalent HQ. (2022, June 5). *Avalanche indexing & querying data API*. Covalent Docs. <https://www.covalenthq.com/docs/networks/avalanche/>
- [12] Ecoembes. (2016, April 12). *Los ordenadores también emiten CO2*. Ecoembes — The Circular Lab. <https://shorturl.at/inwI4>
- [13] Ethers.io. (2022). *Ethers.js — Documentation* <https://docs.ethers.io/v5/>
- [14] Hayes, A. (2021). *Blockchain Definition: What You Need to Know*. Investopedia. <https://www.investopedia.com/terms/b/blockchain.asp>

- [15] GeeksforGeeks. (2022, May 11). *What are Events in Solidity?* GeeksforGeeks. <https://www.geeksforgeeks.org/what-are-events-in-solidity/>
- [16] Metamask. (2022). *The crypto wallet for Defi, Web3 Dapps and NFTs.* <https://metamask.io/>
- [17] PostgreSQL. (2022). *The World's Most Advanced Open Source Relational Database* <https://www.postgresql.org/>
- [18] The Block. (2022). *Ethereum Charts: Active Addresses, Hash Rate and Miner Revenue* <https://www.theblock.co/data/on-chain-metrics/ethereum>
- [19] Truffle Suite. (2022). *Ganache - Truffle Suite* <https://www.trufflesuite.com/ganache>