



UNIVERSITAT POLITÈCNICA DE CATALUNYA  
BARCELONATECH

Facultat d'Informàtica de Barcelona



# IMPLEMENTACIÓ D'UN SISTEMA DE FIRMES DIGITALS SOBRE UN REGISTRE DESCENTRALITZAT BLOCKCHAIN

GUILLEM GARCÍA GIL

**Director/a:** FERRAN PRAT TIÓ (PEERSYST TECHNOLOGY SL )

**Ponent:** CARLES FARRE TOST (Departament d'Enginyeria de Serveis i Sistemes d'Informació)

**Titulació:** Grau en Enginyeria Informàtica (Enginyeria del Software )

Memòria del projecte

Facultat d'Informàtica de Barcelona (FIB)

Universitat Politècnica de Catalunya (UPC) - BarcelonaTech

29/06/2023

## **Resum**

Aquest document representa la memòria del Treball de Fi de Grau de modalitat B en conveni amb l'empresa Peersyst, consultora de tecnologies *blockchain*.

El projecte desenvolupat té com a objectiu poder millorar l'experiència d'ús i l'adopció de la *blockchain* CKB mitjançant una plataforma que permeti gestionar transaccions d'actius digitals entre aplicacions descentralitzades i els usuaris.

## **Resumen**

Este documento representa la memoria del Trabajo Final de Grado de modalidad B en convenio con la empresa Peersyst, consultora de tecnologías *blockchain*.

El proyecto desarrollado tiene como objetivo poder mejorar la experiencia de uso y la adopción de la *blockchain* CKB mediante una plataforma que permita gestionar transacciones de activos digitales entre aplicaciones descentralizadas y los usuarios.

## **Abstract**

This document constitutes the final degree project report of modality B under a collaboration agreement with the blockchain technology consultant firm Peersyst.

The project's goal is to improve user satisfaction and adoption of the CKB blockchain by providing a platform to manage the transfer of digital assets between users and decentralized applications.

# Índex

<b>1</b>	<b>Context</b>	<b>10</b>
1.1	Introducció . . . . .	10
1.2	Termes i conceptes previs . . . . .	10
<b>2</b>	<b>Definició</b>	<b>13</b>
2.1	Oportunitat . . . . .	13
2.2	Actors implicats . . . . .	14
<b>3</b>	<b>Justificació</b>	<b>16</b>
3.1	Estudi de mercat . . . . .	16
3.2	Conclusions . . . . .	17
<b>4</b>	<b>Abast</b>	<b>19</b>
4.1	Objectius . . . . .	19
4.2	Requisits funcionals . . . . .	20
4.3	Requisits no funcionals . . . . .	20
4.4	Obstacles i riscos . . . . .	21
<b>5</b>	<b>Metodologia i rigor</b>	<b>22</b>
5.1	Metodologia de treball . . . . .	22
5.2	Eines de suport a la metodologia . . . . .	23
<b>6</b>	<b>Planificació</b>	<b>24</b>
6.1	Descripció de les tasques . . . . .	24
6.1.1	Gestió i documentació del projecte . . . . .	25
6.1.2	Desenvolupament de CKBull Developer Console . . . . .	25
6.1.3	Desenvolupament de CKBull Wallet . . . . .	26
6.2	Recursos . . . . .	27
6.3	Resum . . . . .	29
<b>7</b>	<b>Riscos</b>	<b>32</b>
<b>8</b>	<b>Pressupost</b>	<b>33</b>
8.1	Identificació dels costos . . . . .	33
8.1.1	Costos sobre recursos humans interns . . . . .	34
8.1.2	Costos sobre els recursos humans externs . . . . .	35
8.1.3	Costos sobre els recursos físics . . . . .	36
8.1.4	Costos sobre els recursos espacials . . . . .	36

8.1.5	Costos sobre els recursos digitals . . . . .	36
8.1.6	Resum dels costos . . . . .	36
8.2	Estimació dels costos . . . . .	37
8.2.1	Costos de contingència . . . . .	37
8.2.2	Costos associats als riscos . . . . .	38
8.2.3	Pressupost final . . . . .	38
8.3	Desviacions sobre els costos del projecte . . . . .	38
<b>9</b>	<b>Especificació</b>	<b>40</b>
9.1	Descripció tècnica de les transaccions a CKB . . . . .	40
9.2	Casos d'ús del sistema . . . . .	44
9.2.1	Casos d'ús de CKBull Developer Console . . . . .	45
9.2.2	Casos d'ús de CKBull Wallet . . . . .	55
9.3	Esquema conceptual de dades . . . . .	60
9.3.1	Descripció de les classes . . . . .	61
9.4	Cicles de vida d'una Request . . . . .	63
9.4.1	SignInRequest (petició d'inici de sessió) . . . . .	63
9.4.2	TransactionRequest (petició de transacció) . . . . .	64
9.5	Especificació dels processos crítics . . . . .	65
9.5.1	Procés d'acceptació d'una petició d'inici de sessió . . . . .	66
9.5.2	Procés de signatura d'una petició de transacció . . . . .	67
<b>10</b>	<b>Disseny</b>	<b>69</b>
10.1	Arquitectura del sistema . . . . .	69
10.2	Disseny de les interfícies . . . . .	71
10.2.1	Disseny de CKBull Developer Console . . . . .	71
10.2.2	Disseny de CKBull Wallet . . . . .	82
10.3	Disseny de domini . . . . .	88
10.3.1	Disseny de l'API . . . . .	88
10.4	Disseny de la persistència . . . . .	93
10.4.1	Disseny de la base de dades . . . . .	93
10.5	Patrons de disseny . . . . .	97
10.5.1	Patrons creacionals . . . . .	97
<b>11</b>	<b>Implementació</b>	<b>99</b>
11.1	Tecnologies comunes . . . . .	99
11.2	CKBull Developer Console . . . . .	100
11.2.1	Tecnologies . . . . .	100
11.3	CKBull Signer API i base de dades . . . . .	101
11.3.1	Tecnologies de l'API . . . . .	101
11.3.2	Tecnologies de la base de dades . . . . .	102
11.3.3	Seguretat . . . . .	102
11.4	CKBull Wallet . . . . .	104
11.5	Técniques i pràctiques . . . . .	106
11.5.1	Monorepo . . . . .	106
11.5.2	Estructura modular . . . . .	106
11.5.3	Entorns . . . . .	107
11.6	Patrons de disseny . . . . .	108

11.6.1	Patrons creacionals . . . . .	108
11.6.2	Patrons estructurals . . . . .	108
<b>12</b>	<b>Tests i validació de requisits</b>	<b>109</b>
12.1	Tècniques de validació . . . . .	109
12.1.1	Tests unitaris . . . . .	109
12.1.2	<i>Mock objects</i> . . . . .	110
12.1.3	Proves no automatitzades end-2-end . . . . .	111
12.2	Validació de requisits . . . . .	113
12.2.1	Requisits funcionals . . . . .	113
12.2.2	Requisits no funcionals . . . . .	116
12.3	Resum de validació . . . . .	118
<b>13</b>	<b>Desplegament i documentació</b>	<b>119</b>
13.1	Desplegament . . . . .	119
13.1.1	Desplegament de CKBull Developer Console i CKBull Signer API . . . . .	119
13.1.2	Desplegament de CKBull Wallet . . . . .	120
13.2	Documentació . . . . .	120
<b>14</b>	<b>Lleis i regulacions</b>	<b>122</b>
14.1	Identificació de lleis . . . . .	122
14.2	Llicències . . . . .	123
<b>15</b>	<b>Resultats de desenvolupament</b>	<b>124</b>
15.1	Desviacions sobre el pla inicial . . . . .	124
15.2	Afectacions sobre el pressupost . . . . .	128
<b>16</b>	<b>Informe de sostenibilitat</b>	<b>130</b>
16.1	Sostenibilitat ambiental . . . . .	130
16.1.1	Posada en producció del projecte (PPP) . . . . .	130
16.1.2	Vida útil . . . . .	132
16.1.3	Riscos . . . . .	133
16.2	Sostenibilitat econòmica . . . . .	133
16.2.1	Posada en producció del projecte (PPP) . . . . .	133
16.2.2	Vida útil . . . . .	133
16.3	Sostenibilitat social . . . . .	134
16.3.1	Posada en producció del projecte . . . . .	134
16.3.2	Vida útil . . . . .	134
16.3.3	Riscos . . . . .	134
<b>17</b>	<b>Conclusions</b>	<b>135</b>
<b>Referències</b>		<b>136</b>
<b>A</b>	<b>Especificació</b>	<b>140</b>
A.1	Diagrama de casos d'ús . . . . .	140

<b>B Disseny</b>	<b>142</b>
B.1 Rutes de CKBull Signer API . . . . .	142
B.1.1 Rutes per peticions d'inici de sessió (sign-in-requests) . . . . .	142
B.1.2 Rutes per peticions d'autenticació . . . . .	146
B.1.3 Rutes per peticions d'usuari . . . . .	148
B.1.4 Rutes per peticions de fitxers . . . . .	149
B.1.5 Rutes per peticions de dApps . . . . .	150
B.1.6 Rutes per peticions de transacció . . . . .	154
B.1.7 Rutes per peticions de <i>playground</i> . . . . .	161

# Índex de figures

1	Representació de UTXOs dins d'una <i>blockchain</i> . . . . .	12
2	Exemple d'una transacció entre una dApp i un usuari amb la nova funcionalitat. . . . .	14
3	Diagrama del funcionament d'un <i>Sprint</i> amb <i>Scrum</i> . . . . .	23
4	Diagrama de Gantt . . . . .	31
5	Transferència simplificada entre dos entitats. . . . .	40
6	Diagrama simplificat d'una transacció. . . . .	41
7	Diagrama del funcionament d'un OutPoint. . . . .	42
8	Diagrama parcial d'una transacció. . . . .	42
9	Estructura de dades del Script. . . . .	43
10	Diagrama parcial d'una transacció a CKB . . . . .	44
11	Casos d'ús <i>CKBull Developer Console</i> . Àmbit autorització i autenticació. . . . .	46
12	Casos d'ús <i>CKBull Developer Console</i> . Àmbit de gestió de dApps. . . . .	49
13	Casos d'ús <i>CKBull Developer Console</i> . Àmbit de generació de peticions. . . . .	53
14	Casos d'ús <i>CKBull Wallet</i> . . . . .	55
15	Esquema conceptual de dades del sistema . . . . .	60
16	Cicle de vida d'una <i>SignInRequest</i> . . . . .	64
17	Cicle de vida d'una <i>TransactionRequest</i> . . . . .	65
18	Procés d'acceptació d'una petició d'inici de sessió. . . . .	67
19	Procés de signatura d'una petició de transacció. . . . .	68
20	Arquitectura general del sistema. . . . .	70
21	Arquitectura dels components a dissenyar. . . . .	71
22	Arquitectura de <i>CKBull Developer Console</i> . . . . .	72
23	Mapa de navegabilitat de <i>CKBull Developer Console</i> . . . . .	73
24	Disseny de pàgina d'inici de sessió. . . . .	74
25	Disseny de pàgina de verificació de compte. . . . .	75
26	Disseny de pàgina de les meves dApps. . . . .	76
27	Disseny de creació de dApps. . . . .	77
28	Disseny del modal de credencials de la dApp. . . . .	78
29	Disseny de pàgina d'edició de dApp. . . . .	79
30	Connexió entre capa de presentació i domini <i>CKBull Developer Console</i> . . . . .	80
31	Disseny de capa de domini de <i>CKBull Developer Console</i> . . . . .	81
32	Connexió entre capa de domini i accés a dades <i>CKBull Developer Console</i> . . . . .	82
33	Diagrama de navegació de <i>CKBull Wallet</i> . . . . .	83
34	Disseny de la pàgina d'activitat. . . . .	84

35	Dissenys del modal d'inici de sessió . . . . .	85
36	Dissenys del diàleg de declinació d'inici de sessió. . . . .	86
37	Disseny de pàgina d'èxit de signatura de transacció. . . . .	87
38	Arquitectura de CKBull Signer API. . . . .	88
39	Diagrama de la capa de controladors de l'API. . . . .	89
40	Diagrama de serveis de l'API. . . . .	90
41	Diagrama de serveis amb interfícies de repositoris. . . . .	91
42	Diagrama de repositoris de l'API. . . . .	92
43	Disseny de les taules de la base de dades. . . . .	97
44	Procés d'autenticació amb JWT. . . . .	103
45	Procés d'autenticació per dApps. . . . .	104
46	Estructura de carpetes del monorepo. . . . .	106
47	Exemples d'estructura modular a CKBull Developer Console. . . . .	107
48	Relació entre entorns. . . . .	107
49	Test unitari del component DAppList. . . . .	110
50	Mock de la classe CompleteDAppDto. . . . .	110
51	Prova del cas d'ús Signar transacció. . . . .	112
52	Tests unitaris realitzats sobre els components del sistema. . . . .	118
53	Exemple desplegament gestionat desde Expo. . . . .	120
54	Resultat de la tasca <i>Playground</i> . . . . .	126
55	Codi QR de petició d'inici de sessió. . . . .	127
56	Formulari per crear una petició de transacció . . . . .	127
57	Elements de la llista d'events. . . . .	128
58	Diagrama de casos d'ús. . . . .	141

# Índex de taules

1	Comparativa de mercat entre diferents productes i la solució del projecte. . . . .	17
2	Recursos humans interns. . . . .	27
3	Recursos humans externs. . . . .	27
4	Recursos físics. . . . .	28
5	Costs sobre recursos espaials. . . . .	28
6	Recursos digitals. . . . .	28
7	Resum de les tasques. . . . .	30
8	Riscos detectats en el projecte. . . . .	32
9	Costos sobre recursos humans interns. . . . .	34
10	Resum de les tasques. . . . .	35
11	Costos sobre recursos humans externs. . . . .	36
12	Costos sobre recursos físics. . . . .	36
13	Costs sobre recursos espaials. . . . .	36
14	Costos sobre recursos digitals. . . . .	37
15	Resum dels costos. . . . .	37
16	Costos de contingència. . . . .	37
17	Riscos detectats en el projecte. . . . .	38
18	Pressupost final. . . . .	38
19	Actors dins del nostre sistema. . . . .	45
20	Descripció de cas d'ús <i>Enregistrar-se</i> . . . . .	46
21	Descripció de cas d'ús <i>Iniciar sessió</i> . . . . .	47
22	Descripció de cas d'ús <i>Iniciar sessió</i> . . . . .	47
23	Descripció de cas d'ús <i>Reestablir contrasenya</i> . . . . .	48
24	Descripció de cas d'ús <i>Validat compte</i> . . . . .	48
25	Descripció de cas d'ús <i>Tancar sessió</i> . . . . .	49
26	Descripció de cas d'ús <i>Crear dApp</i> . . . . .	50
27	Descripció de cas d'ús <i>Generar claus API</i> . . . . .	50
28	Descripció de cas d'ús <i>Editar dApp</i> . . . . .	51
29	Descripció de cas d'ús <i>Eliminar dApp</i> . . . . .	51
30	Descripció de cas d'ús <i>Regenerar clau secreta API</i> . . . . .	52
31	Descripció de cas d'ús <i>Veure dApps</i> . . . . .	52
32	Descripció de cas d'ús <i>Generar petició d'inici de sessió</i> . . . . .	53
33	Descripció de cas d'ús <i>Consultar petició d'inici de sessió</i> . . . . .	54
34	Descripció de cas d'ús <i>Generar petició de transacció</i> . . . . .	54
35	Descripció de cas d'ús <i>Consultar petició de transacció</i> . . . . .	55

36	Descripció de cas d'ús <i>Veure aplicacions connectades</i> . . . . .	56
37	Descripció de cas d'ús <i>Escanejar codi inici de sessió</i> . . . . .	56
38	Descripció de cas d'ús <i>Rebutjar inici de sessió</i> . . . . .	57
39	Descripció de cas d'ús <i>Acceptar inici de sessió</i> . . . . .	57
40	Descripció de cas d'ús <i>Veure peticions de transaccions</i> . . . . .	58
41	Descripció de cas d'ús <i>Signar transacció</i> . . . . .	58
42	Descripció de cas d'ús <i>Rebutjar transacció</i> . . . . .	59
43	Atributs de la classe Developer. . . . .	61
44	Atributs de la classe DApp. . . . .	61
45	Atributs de la classe Credentials. . . . .	62
46	Atributs de la classe SignInRequest. . . . .	62
47	Atributs de la classe TransactionRequest. . . . .	62
48	Atributs de la classe Transaction. . . . .	63
49	Atributs de la classe Wallet. . . . .	63
50	Restriccions de la taula AccountMetadata. . . . .	93
51	Restriccions de la taula SignInRequest. . . . .	94
52	Restriccions de la taula TransactionRequest. . . . .	94
53	Restriccions de la taula Transaction. . . . .	94
54	Restriccions de la taula DApp. . . . .	95
55	Restriccions de la taula User. . . . .	95
56	Restriccions de la taula VerifyEmailToken. . . . .	96
57	Restriccions de la taula ResetToken. . . . .	96
58	Validació de RF1. . . . .	113
59	Validació de RF2. . . . .	113
60	Validació de RF3. . . . .	114
61	Validació de RF4. . . . .	114
62	Validació de RF5. . . . .	115
63	Validació de RF6. . . . .	115
64	Validació de RF7. . . . .	116
65	Validació de RF8. . . . .	116
66	Validació de RF9. . . . .	116
67	Validació de RNF1. . . . .	117
68	Validació de RNF2. . . . .	117
69	Validació de RNF3. . . . .	117
70	Validació de RNF4. . . . .	117
71	Validació de RNF5. . . . .	118
72	Validació de RNF6. . . . .	118
73	Desviacions per <i>sprints</i> . . . . .	124
74	Petició per obtindre peticions d'inici de sessió. . . . .	142
75	Petició per crear una petició d'inici de sessió. . . . .	143
76	Petició per obtindre les peticions d'inici de sessió creades per una dApp. . . . .	143
77	Petició per obtenir informació d'una petició d'inici de sessió. . . . .	144
78	Petició per fer <i>polling</i> d'una petició d'inici de sessió. . . . .	144
79	Petició d'acceptació d'una petició d'inici de sessió. . . . .	145
80	Petició de declinació d'una petició de d'inici de sessió. . . . .	145

81	Petició de desconexió de dApps. . . . .	146
82	Petició per iniciar sessió com desenvolupador. . . . .	146
83	Petició per iniciar sessió amb Google com desenvolupador. . . . .	146
84	Petició per redirigir després d'iniciar sessió . . . . .	146
85	Petició per validar direcció de correu com desenvolupador. . . . .	147
86	Petició per recuperar la contrasenya com desenvolupador. . . . .	147
87	Petició per a restablir la contrasenya com desenvolupador. . . . .	147
88	Petició per a obtenir informació d'un usuari. . . . .	148
89	Petició per a enregistrar un usuari. . . . .	148
90	Petició per a editar un usuari. . . . .	149
91	Petició per eliminar un usuari. . . . .	149
92	Petició per a pujar una imatge. . . . .	149
93	Petició per a obtindre totes les dApps d'un usuari. . . . .	150
94	Petició per a obtindre una dApp. . . . .	151
95	Petició per a enregistrar una dApp. . . . .	152
96	Petició per a enregistrar una dApp. . . . .	153
97	Petició per eliminar una dApp. . . . .	153
98	Petició per eliminar una dApp. . . . .	154
99	Petició d'obtenció de peticions de transaccions. . . . .	154
100	Petició de creació de peticions de transaccions. . . . .	155
101	Petició d'obtenció de peticions de transacció per dApp. . . . .	156
102	Petició d'obtenció de petició de transacció per transactionToken. . . . .	157
103	Petició d'obtenció de l'estat petició de transacció per transactionToken. . . . .	157
104	Petició de generació d'una transacció de token natiu. . . . .	158
105	Petició de generació d'una transacció d'un nft. . . . .	158
106	Petició de generació d'una transacció d'un nft. . . . .	159
107	Petició de generació d'una transacció d'un nft. . . . .	160
108	Petició de desconexió de dApps. . . . .	161

# Capítol 1

## Context

### 1.1 Introducció

El contingut d'aquest document representa el Treball de Fi de Grau titulat *Implementació d'un sistema de firmes digitals sobre un registre descentralitzat blockchain*, realitzat per l'alumne Guillem García Gil, en l'especialització d'Enginyeria del Software pertanyent al Grau en Enginyeria Informàtica de la Facultat Politècnica de Barcelona a la Universitat Politècnica de Catalunya.

Aquest projecte correspon a la modalitat B (realitzat amb empresa) amb l'empresa Peersyst [1], consultora de tecnologia *blockchain* localitzada a Barcelona. És dirigit per Ferran Prat Tio, CEO de Peersyst, i assessorat per en Carles Farré Tost, personal docent i investigador al Departament d'Enginyeria de Serveis i Sistemes d'Informació.

El projecte és el resultat d'una col·laboració entre Peersyst i l'empresa Nervos [2], que té com a finalitat la implementació d'un sistema de firmes de transaccions a la *blockchain* Nervos Network, mitjançant la cartera digital *CKBull Wallet*, producte dissenyat i implementat per Peersyst.

### 1.2 Termes i conceptes previs

Per tal d'introduir al lector en el sector de la tecnologia *blockchain*, sobre els seus conceptes, implementacions i usos, es definiran a continuació una llista de termes i conceptes que ajudaran a seguir i comprendre el contingut d'aquest projecte.

#### Blockchain

Conegut també com a cadena de blocs, és un sistema de base de dades descentralitzat que permet enregistrar transaccions entre diferents usuaris. Aquestes transaccions poden estar lligades tant a actius tangibles com a actius intangibles, digitals i físics. Les principals propietats de les cadenes de blocs són les següents:

- **Immutabilitat:** Tota transacció realitzada dins de la *blockchain* és immutable, és a dir, no pot canviar d'estat.
- **Transparència:** Tot usuari té accés a tota la resta de transaccions
- **Consens:** Per aprovar una transacció tots els participants de la *blockchain* (nodes) o la majoria han d'estar d'acord.

Una bona analogia d'una *blockchain* és un llibre de comptabilitat, on cada moviment sobre un o diversos actius és enregistrat quan es realitza una transacció d'aquest entre alguns agents [3] [4].

La cadena de blocs utilitza una gran quantitat de conceptes criptogràfics que quedarán fora de l'abast d'aquest projecte, únicament incloent el contingut necessari.

### Nervos

Blockchain sobre la que es realitzarà el sistema de firmes digitals. Nervos és una *blockchain* pública i *open-source* [5] que busca com a objectiu convertir-se en una *blockchain* multiús, aportant seguretat i escalabilitat a totes les solucions.

### Mainnet i Testnet

Corresponen a les diferents xarxes de blockchain de Nervos. Cada xarxa té un propòsit de funcionament. Testnet es una xarxa de proves on es realitzen totes les integracions i desenvolupaments que, un cop finalitzats, passen a la Mainnet, xarxa que podríem catalogar com a xarxa en producció.

### Account o compte

Objecte distribuit utilitzat a la blockchain Nervos per a guardar la informació del usuari i el seu balanç.

### Address o adreça

Cadena de caràcters que representa de forma anònima la identitat d'un usuari dins de la blockchain. A més, són utilitzades com a referència per a la transferència d'actius digitals entre comptes. Un exemple d'adreça dins de la *blockchain* Nervos es el següent:

```
ckb1qzda0cr08m85hc8jlnfp3zer7xulejywt49kt2rr0vthywaa50xwsqdnw7qkdnclfkg  
59uzn8umtd2kwxceqwxquc4
```

on la adreça està composta sempre per el prefix *ckb* per indicar que l'adreça pertany a **Mainnet**, i *ckt* per indicar que l'adreça pertany a **Testnet**. Un compte pot generar múltiples adreces. Per a la seva creació es poden utilitzar dues estratègies: **BIP173**[6] i la **BIP350**[7]. Degut a que no es necessàri conèixer com es generen les adreces, queda exclosa l'explicació.

### Criptomonedra o token natiu

Tipus de divisa digital utilitzada dins d'una xarxa *blockchain* per a fer transaccions. Aquesta divisa té com a propietat principal que no es distribuïda per a cap entitat central. Cada xarxa *blockchain* disposa del seu token natiu que s'utilitza dins de la cadena. Dins de la *blockchain* de Nervos el token natiu es representa amb les sigles CKB [8].

### dApps o aplicacions descentralitzades

Una *dApp* o aplicació descentralitzada correspon a una solució dissenyada i implementada que interactua amb una xarxa *blockchain*. Aquestes aplicacions soelen estar segmentades en una interfície d'usuari (punt de connexió pels usuaris) i un contracte intel·ligent.[9].

### Wallet

Una wallet o cartera de criptomonedes és una aplicació distribuïda que permet a un usuari gestionar de manera senzilla els seus criptoactius dins d'una xarxa blockchain. Les carteres abstraueixen tot el coneixement criptogràfic necessari per a funcionar perquè els usuaris no hagin de ser responsables de gestionar les seves claus. A més, existeixen diferents tipus de carteres per a diferents usos, les quals no seran explicades ja que queden fora de l'abast del projecte [10].

## Problema del doble pagament

El problema del doble pagament correspon al risc de que es puguin realitzar més d'un pagament amb un mateix actiu digital i es una de les principals preocupacions a tindre en compte en l'utilització de sistemes moneratis digitals. El següent exemple ajuda a explicar el problema del doble pagament:

Una persona amb 10 monedes digitals emmagatzemats a una entitat vol realitzar dues compres. Per a les dues transaccions, genera un certificat a mode de xec conforme té 10 monedes digitals i realitza les compres a entitats diferents. Així doncs, la primera entitat que reclami el valor del xec obtindrà les 10 monedes digitals i l'altre no podrà obtenir-les, degut a que ja han sigut reclamades.

Com a simplificació, correspondria al problema on dues entitats han promès ser pagades però només una d'elles ho ha estat.

## UTXO

Les sigles UTXO pertanyen al terme en anglès *Unspent transaction output* i representa una quantitat de monedes digitals que han sigut autoritzades per a transferir-les d'un compte a un altre [11]. Cada UTXO representa una cadena de propietat de criptodivises que s'han realitzat desde l'inici de la creació de la moneda.

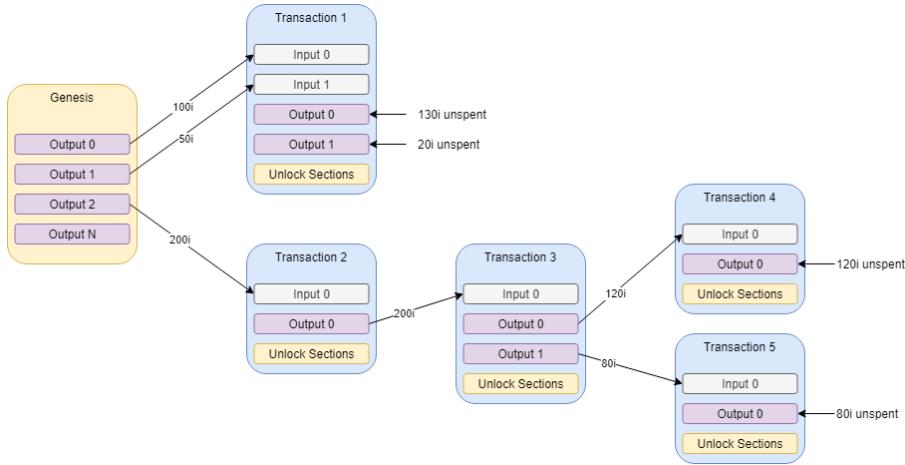


Figura 1: Representació de UTXOs dins d'una *blockchain*.

Font: Wiki IOTA.

Com podem veure a la Figura 1 els UTXOs es van generant a partir de transaccions realitzades entre diferents usuaris. El *Genesis* correspon a l'inici de la cripto moneda. Cada transacció de sortida (*output*) té un valor màxim que no pot excedir-se a la següent transacció. D'aquesta manera, el model UTXO sol·luciona el problema del doble pagament. A més, els UTXOs permeten tindre múltiples transaccions d'entrada (*inputs*) i múltiples transaccions de sortida (*outputs*).

# Capítol 2

## Definició

Aquest capítol comprèn la definició del propòsit del projecte, analitzant l'oportunitat que suposa dins de la *blockchain* CKB i a qui afecta la producció d'aquest.

### 2.1 Oportunitat

Actualment, els usuaris de la *blockchain* CKB disposen de múltiples carteres per a gestionar els seus actius digitals, com ara els *tokens* natius de la *blockchain*, altres *tokens* i NFTs. Dins d'aquest conjunt de carteres es troba **CKBull Wallet**, la primera cartera mòbil per la *blockchain* CKB. Aquesta cartera ha sigut desenvolupada i mantinguda per Peersyst. Un dels objectius que va comportar el desenvolupament d'aquesta cartera mòbil (per iOS i Android) va ser millorar l'experiència d'usuari per afavorir l'adopció de la *blockchain*, pel fet que avui en dia gairebé tothom disposa d'un *smartphone*.

No obstant això, les accions que es poden realitzar mitjançant una cartera dins de CKB són limitades a les accions d'enviar i rebre actius digitals entre comptes, entre altres funcionalitats. Així doncs, amb la idea de poder obrir més oportunitats als usuaris de CKB mitjançant la cartera CKBull Wallet, va sorgir l'oportunitat d'aquest projecte.

Nervos, l'empresa darrere de la creació de CKB, va voler desenvolupar una **plataforma de pagaments i signatures digitals** on qualsevol dApp creada per desenvolupadors pogués **agilitzar i millorar l'experiència d'usuari a l'hora de fer transaccions d'actius entre dApp i usuari**. Per entendre millor aquest concepte, suposem el següent exemple.

Una dApp (anomenada Zara) que ven roba a un preu en CKB (*token* natiu de la *blockchain* CKB) vol poder integrar els pagaments dels seus usuaris amb la cartera CKBull Wallet d'una manera semblant a la que funciona el SCA (*Strong Customer Authentication*) per a compres online amb diners fiduciaris (com euros). La següent figura mostra un exemple de com funcionaria la plataforma durant un pagament (transacció) entre un usuari i una dApp.

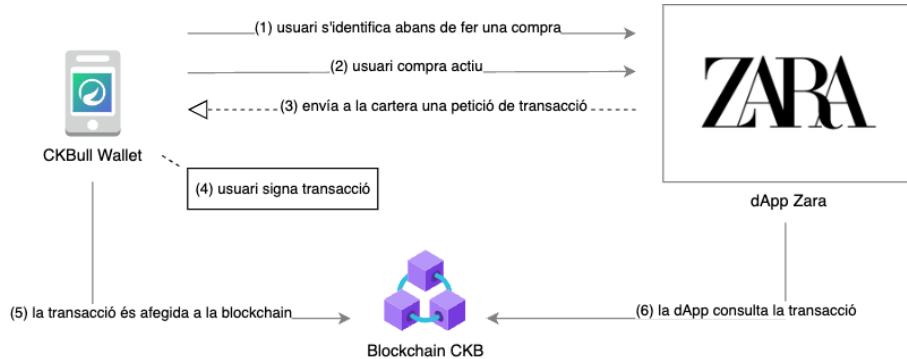


Figura 2: Exemple d'una transacció entre una dApp i un usuari amb la nova funcionalitat.

Font: elaboració pròpria

Així doncs, l'objectiu d'aquest projecte resideix en desenvolupar una solució que permeti integrar a una dApp dins de l'ecosistema de CKB perquè l'usuari pugui gestionar peticions de transaccions mitjançant la cartera CKBull Wallet.

Per a donar-li un nom d'aquí en endavant a la solució del projecte, aquest rebrà el nom de **CKBull Signer App**.

## 2.2 Actors implicats

En ser una col·laboració empresarial entre dues empreses, el nombre de parts interessades en la producció del projecte és més elevada en comparació amb una producció pròpia del producte per part de Peersyst. Per a entrar més en detall, a continuació es detallen els principals actors i parts implicades en aquest projecte:

- **Autor del Treball Final de Grau:** Principal interessat en la producció i finalització del projecte.
- **CKBull Wallet:** Cartera de criptomonedes mòbil on els usuaris de la wallet podran gaudir de la nova funcionalitat.
- **Desenvolupadors d'aplicacions descentralitzades a l'ecosistema CKB:** Actors interessats en el desenvolupament del sistema per aportar major qualitat als seus productes i millorar l'experiència d'usuari.
- **Director del Treball Final de Grau:** Ferran Prat Tio, CEO de Peersyst i director del projecte, és el principal encarregat del seguiment i supervisió dels objectius del projecte.
- **Nervos:** Empresa col·laboradora i beneficiada de la creació del sistema, aportant major facilitat d'ús del seu ecosistema per als seus usuaris.
- **CKB Blockchain:** Actor i tecnologia sobre la que es desenvoluparà el projecte.
- **Peersyst:** Empresa creadora del projecte i principal interessada en la producció d'una sol·lució de qualitat, segura, escalable i usable.
- **Ponent del Treball Final de Grau:** Com a ponent del Treball Final de Grau, en Carles Farré Tost és l'encarregat de guiar el treball a l'autor així ajudar en la possible resolució de conceptes.

- **Usuari de CKB:** Actor beneficiat de la producció del projecte respecte a l'ús d'aquest. Interessat a obtenir major control sobre les seves transaccions i actius digitals.
- **Potencials competidors:** Encara que Peersyst és l'única empresa que duu a terme el projecte sobre la *blockchain* CKB, altres empreses o organitzacions poden fer projectes semblants per a altres *blockchains* competidores amb CKB.
- **Usuaris nous en la tecnologia:** Possibles actors interessats en el nou sistema i atrets pel seu ús dins de l'ecosistema de Nervos.

# Capítol 3

## Justificació

En aquest capítol es detalla en quina posició es troba el projecte respecte al mercat actual a la data de finals de febrer de 2023. A més, s'elaborarà una anàlisi sobre quins són els punts forts i febles de la solució respecte a els seus potencials competidors.

### 3.1 Estudi de mercat

Prèviament al desenvolupament del projecte, és important ser conscient de la situació de la qual es parteix a l'hora d'iniciar el projecte. Cal obtenir certa perspectiva sobre el sector de la *blockchain*, especialment de la *blockchain* CKB, per avaluar com d'únic, prometedor i viable és el nostre projecte en comparació amb els projectes actuals o potencials competidors. Com a objectiu addicional d'aquest estudi de mercat, no busquem únicament trobar potencials competidors del nostre projecte, sinó buscar altres solicions que puguin aportar altres idees de com desenvolupar la solució.

Abans d'iniciar l'estudi de mercat, és rellevant **determinar quins punts principals tindrà la nostra solució** per, a continuació, fer una comparativa entre altres solicions disponibles. Com s'ha descrit al capítol anterior, l'objectiu d'aquest projecte és **integrar una plataforma de pagaments amb l'actual cartera CKBull Wallet**, una de les moltes altres carteres que existeixen dins de la *blockchain* CKB i que poden tractar-se com a potencials competidors.

- **F1 - Cartera sobre CKB:** La cartera permet visualitzar el balanç d'un compte a la *blockchain* CKB.
- **F2 - Plataforma per enregistrar dApps :** Disposa d'una plataforma per als desenvolupadors d'aplicacions descentralitzades per a enregistrar les seves aplicacions al sistema.
- **F3 - Passarel·la de pagament (o semblant) :** Permet a l'usuari poder signar transaccions generades per tercers.
- **F4 - Aplicació mòbil:** La cartera de criptomonedes té aplicació mòbil.
- **F5 - Aplicació multiplataforma:** La cartera de criptomonedes es troba disponible tant per iOS com per Android (en cas de ser una aplicació mòbil).
- **F6 - SDK / Llibreria:** El producte disposa d'un SDK o llibreries que permeti als desenvolupadors implementar funcionalitats de firma als seus projectes.

- **F7 - Funciona sobre testnet i mainnet:** El producte funciona sobre les dues xarxes de la *blockchain* CKB: testnet i mainnet.

Un cop definides les funcionalitats que són rellevants dins la nostra solució, és hora d'investigar quins altres productes ja existents poden declarar-se competidors. Els possibles competidors del sistema són els següents:

- **Neuron:** Neuron és de les primeres carteres de CKB que es van desenvolupar. Els creadors darrere d'aquest *software* és l'empresa Nervos amb la que es col·labora en aquest projecte. Aquesta cartera només pot ser utilitzada com a aplicació d'escriptori, ja que es troba disponible per als sistemes operatius Windows, MacOS i Linux [12].
- **ImToken:** Cartera de criptomonedes multiblockchain que permet gestionar els actius digitals de diferents *blockchains*. No obstant, la seva funcionalitat sobre la *blockchain* CKB es limita a enviar i rebre transaccions. Té format d'aplicació mòbil disponible per a iOS i Android [13].
- **SafePal:** Com ImToken, SafePal es una cartera de criptomonedes multiblockchain que també permet gestionar els actius de la *blockchain* CKB. No obstant això, a diferència d'ImToken aquesta permet intercanviar tipus de criptomonedes dins de l'aplicació [14].
- **JoyID:** A diferència de la resta de productes avaluats, JoyID és una cartera de criptomonedes per a la web. Aquesta té com a finalitat eliminar barreres tecnològiques per a una millor adopció de la tecnologia *blockchain*. Malauradament, aquesta cartera únicament treballa sobre la xarxa **testnet i no té cap utilitat sobre les xarxes mainnet** [15].

Així doncs, a la taula següent es mostra el resultat de l'estudi de mercat realitzat sobre les funcionalitats principals de la nostra solució en comparació amb els potencials competidors:

Productes	Atributs						
	F1	F2	F3	F4	F5	F6	F7
CKBull Signer App	Sí	Sí	Sí	Mòbil	Sí	Sí	Sí
Neuron	Sí	No	No	Escriptori	No	No	Sí
ImToken	Sí	No	No	Mòbil	Sí	Si	Sí
SafePal	Sí	No	No	Mòbil	S	No	Sí
JoyID	Sí	No	No	Web	No	Sí	No

Taula 1: Comparativa de mercat entre diferents productes i la solució del projecte.

### 3.2 Conclusions

Com podem observar a la Taula 1, existeixen dos productes que més funcionalitats comparteixen amb la nostra solució. Aquests dos productes són **ImToken** i **SafePal**. Tot sent les dues carteres de criptomonedes en format d'aplicació mòbil, la principal manca de la qual no disposen és de les funcionalitats F2 i F3. Les dues carteres no disposen d'una plataforma que permeti a dApps gestionar de manera interna pagaments generats per aquests, ni una plataforma on aquests puguin enregistrar-se per a formar part de la cartera. El producte més semblant que pot arribar a ser un potencial competidor per a CKBull Signer API és **ImToken**. Aquesta cartera disposa d'integració

amb dApps, incloent-hi una API per a desenvolupadors perquè lliguin la dApp a la cartera. Malgrat això, no fa cap referència a poder generar peticions de transaccions entre dApp i la cartera. A més, en ser ImToken una cartera ***multiblockchain*** **perd el factor d'exclusivitat que poden rebre els usuaris** utilitzant CKBull Wallet, on disposen de més funcionalitats sobre la *blockchain* CKB que aquest competidor.

Respecte a la resta de productes, Neuron no es considera una cartera amb risc potencial sobre el projecte, perquè funciona sobre un entorn diferent (escriptori) a CKBull Wallet i no disposa de cap mena d'integració per a desenvolupadors.

Per acabar, JoyID no es considera tampoc un producte que pugui afectar a la solució del projecte en el curt termini, pel fet que aquest es troba en una fase inicial de desenvolupament perquè només funciona sobre la xarxa testnet de CKB. Això no obstant, és convenient mantenir un cert seguiment perquè en un futur poden comprometre la solució del projecte si desenvolupen funcionalitats que millorin l'adopció de CKB per part dels usuaris.

# Capítol 4

## Abast

Dins d'aquest capítol del projecte, es definiran els objectius a aconseguir per a la finalització del projecte, així com els corresponents subobjectius, requisits funcionals i no funcionals. A més, s'analitzaran els possibles obstacles i riscos que poden sorgir durant la vida del projecte i com es reaccionarà davant d'aquests per a mitigar l'impacte que comportin.

### 4.1 Objectius

El resultat final del projecte ha de satisfer els següents objectius per a classificar-ho com a satisfactori. La notació utilitzada per a definir els objectius i subobjectius correspon a OBX (on X indica el nombre d'objectiu) i OBX.Y (on Y indica el nombre de subobjectiu i X referència el nombre d'objectius al qual pertany).

- **OB1:** Oferir un espai als desenvolupadors de *dApps* pugui enregistrar les seves dApps a CKBull Signer App.
  - **OB1.1:** L'espai ha de permetre enregistrar una dApp al sistema, poder visualitzar la seva informació, editar-la i eliminar-la de la plataforma.
- **OB2:** Oferir una plataforma que permeti a les dApps enviar peticions sobre els usuaris de CKBull Wallet.
  - **OB2.1:** La plataforma ha de permetre a una dApp enregistrada per un desenvolupador enviar peticions d'inici de sessió on s'identificarà a l'usuari de CKBull Wallet.
  - **OB2.2:** La plataforma ha de permetre a una dApp enregistrada per un desenvolupador enviar peticions de transacció sobre usuaris de CKBull Wallet prèviament identificats.
- **OB3:** Permetre a l'usuari de *CKBull Wallet* prendre accions de signatura i rebutjament de peticions de transaccions realitzades per dApps enregistrades al sistema.
- **OB4:** Mantenir el projecte sota *testing*<sup>1</sup> [16] durant el desenvolupament i pas a producció d'aquest per garantir un funcionament correcte.

---

<sup>1</sup> *Testing:* Procés d'evaluació i verificació del comportament sobre un producte de *software* o aplicació.

## 4.2 Requisits funcionals

Un cop definits els objectius principals del projecte, definim els requisits funcionals del projecte que determinaran si la solució del projecte s'ha dut a terme amb èxit. La seva notació correspon a RFX (on X indica el nombre de requisit).

- **RF1:** El sistema ha de permetre als desenvolupadors poder enregistrar-se al sistema mitjançant l'ús d'un correu electrònic personal o un compte de Google.
- **RF2:** El sistema ha de permetre als desenvolupadors enregistrar, consultar, editar i esborrar una o varíes dApps.
- **RF3:** El sistema ha d'obligar als desenvolupadors de dApps a autenticar-se per a poder utilitzar l'espai d'enregistrament de dApps.
- **RF4:** El sistema ha de permetre generar credencials perquè el sistema identifiqui les dApps enregistrades dels desenvolupadors.
- **RF5:** El sistema ha de permetre a l'usuari de *CKBull Wallet* acceptar, rebutjar i visualitzar les peticions d'inici de sessió generades per les aplicacions enregistrades al sistema.
- **RF6:** El sistema ha de permetre a l'usuari de *CKBull Wallet* acceptar, rebutjar i visualitzar les transaccions generades per les aplicacions enregistrades al sistema.
- **RF7:** El sistema ha de permetre a les dApps enregistrades comprovar l'estat de les transaccions de les seves aplicacions.
- **RF8:** El sistema ha de permetre a les dApps enregistrades comprovar l'estat de les peticions d'inici de sessió de les seves aplicacions.
- **RF9:** S'ha d'ofrir als desenvolupadors la documentació necessària per a poder utilitzar la plataforma de manera correcta.

## 4.3 Requisits no funcionals

A més dels requisits funcionals definits a la secció anterior, s'ha volgut definir una sèrie de requisits no funcionals per a mantenir certs estàndards sobre els diferents dominis: **aspecte**, **rendiment** i **seguretat**. La seva notació correspon a RNFX (on X indica el nombre de requisit). A continuació es llisten els requisits no funcionals per categories segons el sistema Volere [17].

### Requisits d'aspecte

- **RNF1 - Aparença:** El sistema ha de mantenir una aparença semblant amb altres productes de Nervos.
- **RNF2 - Usabilitat:** El sistema ha d'evitar que l'usuari produueixi errors durant l'ús d'aquest.

### Requisits d'usabilitat i d'humanitat

- **RNF3 - Internacionalització:** El sistema, incloent-hi l'espai per a desenvolupadors i *CKBull Wallet*, ha de disposar de suport d'idioma anglès i castellà.

## Requisits de rendiment

- **RNF4 - Fiabilitat i disponibilitat:** El sistema ha de ser usable el 90% del temps que estigui actiu.

## Requisits de seguretat

- **RNF5 - Accés:** El sistema només ha de permetre l'accés a desenvolupadors i dApps mitjançant credencials vàlides.
- **RNF6 - Privacitat:** El sistema ha de complir els requisits necessaris detectats al RGPD<sup>2</sup>.

## 4.4 Obstacles i riscos

L'espai de temps que comporta la realització del Treball de Grau comença a finals de gener de 2023 i acaba finals de juny del mateix any. Sis mesos de desenvolupament és temps suficient perquè puguin sorgir certs obstacles i impediments que afectin de manera negativa sobre la producció del projecte. Per a poder reaccionar de la manera més ràpida i eficient possible, es llisten a continuació els possibles obstacles i riscos contemplats dins del domini del projecte:

- **OR1 - Limitacions de coneixement i limitacions tècniques:** El projecte tracta de manera directa conceptes sobre *blockchain*, comunicacions i transaccions, en específic sobre el domini de la *blockchain* CKB. Aprendre els conceptes necessaris requereix cert temps i romandrà present durant el desenvolupament del projecte. Això no obstant, dins de l'equip de Peersyst hi ha professionals coneixedors d'aquests conceptes. A més, *Nervos* ofereix documentació elaborada sobre tots els aspectes que es treballaran dins del projecte, fet que facilitarà l'aprenentatge.
- **OR2 - Salut:** En recaure el desenvolupament en una única persona, existeix la possibilitat de patir una malaltia o incapacitat que no permeti continuar el treball durant un cert temps. Aquest fet s'hauria de tindre en compte respecte a la planificació de les tasques i entregues.
- **OR3 - Manca o error a la definició:** No definir de manera exhaustiva el problema i la sol·lució a implementar pot comportar un efecte negatiu que arrossegi errors cada cop que avanci el projecte. És realment important trobar els principals punts de fallada del sistema i planificar i gestionar de manera eficaç, i a poder ser eficient, els casos d'ús més crítics.
- **OR4 - Aparició imprevista d'errors o bugs:** A causa de la magnitud del projecte, s'ha de tindre en compte que sorgiran errors i *bugs*<sup>3</sup> de totes dimensions. El problema principal que comporten és l'endarreriment que generi implementar una solució per aquests, ja que la incertesa sobre la tipologia d'aquests es completa i és difícil poder prevenir i classificar com afectaran el projecte. Per a poder estar mínimament preparats és valuós deixar cert marge temporal a la planificació per a possibles aparicions.

---

<sup>2</sup>RGPD: Reglament Europeu de Protecció de Dades.

<sup>3</sup>*Bug:* Error o defecte que causa un resultat incorrecte o inesperat en un programa o sistema, o que es comporti de forma no prevista.

# Capítol 5

## Metodologia i rigor

### 5.1 Metodologia de treball

La metodologia de treball utilitzada al projecte és la metodologia utilitzada a l'equip de desenvolupament de Peersyst. Peersyst fa servir **Scrum** [18]. Aquesta metodologia té com a objectiu completar divisions de tasques en un determinat temps. Aquests períodes de temps s'anomenen **Sprints**, on s'inclouen un nombre de tasques a completar, i en el cas de Peersyst tenen una duració de dues setmanes.

A l'inici del projecte, les funcionalitats són definides i segmentades en tasques al *Product Backlog*, on, posteriorment, es distribuiran als diferents Sprints de desenvolupament.

Respecte a la sincronització entre els membres de l'equip, es realitzen a cada *Sprint* diferents esdeveniments per tal de garantir un seguiment i desenvolupament controlat:

- **Sprint Planning:** Reunió que té lloc a l'inici de cada sprint on es decideix que es produirà durant les pròximes dues setmanes.
- **Daily Stand Up:** Reunió diària on tots els membres de l'equip posen en comú l'estat de desenvolupament, així com errors trobats i idees.
- **Sprint Retrospective:** Reunió en finalitzar el sprint on es posen en comú totes les idees i esdeveniments que hagin succeït durant aquest.

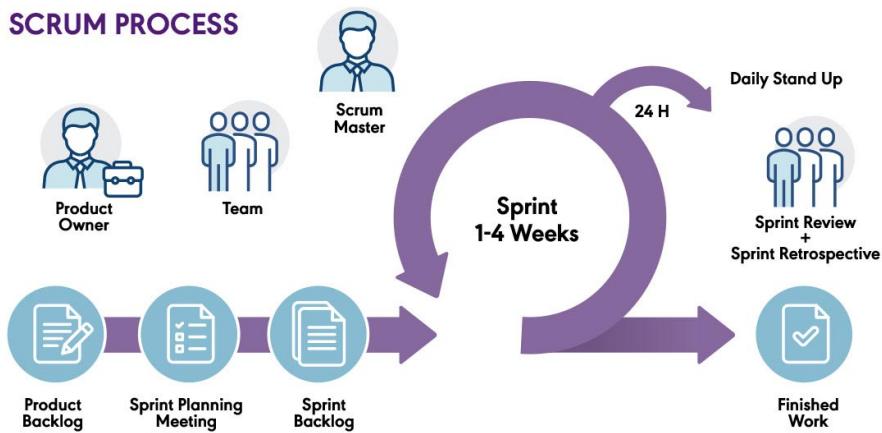


Figura 3: Diagrama del funcionament d'un *Sprint* amb *Scrum*.  
Font: PM-Partners.

## 5.2 Eines de suport a la metodologia

Un cop especificada la metodologia de treball emprada, es descriuran les eines de suport que s'utilitzaran durant el desenvolupament del projecte i ajudaran a seguir i mantenir la metodologia:

- **Slack:** *Software* de missatgeria digital que permet la comunicació entre membres d'un equip mitjançant canals i sistemes de trucades. Serà utilitzat per a realitzar el *Sprint Planning*, *Daily Scrum* i *Sprint Retrospective* [19].
- **Github:** Plataforma en línia per allotjar principalment repositoris de codi. Peersyst gestiona tots els seus projectes a Github degut a les integracions amb eines descrites a continuació [20].
- **Github Projects:** Eina per a planificació de projectes. L'equip de Peersyst fa ús d'aquest *software* per a mantenir un control i seguiment de l'estat dels projectes [21].
- **Github Actions:** Eina de *Continuous Intergration/Deployment* [22] que permet agilitzar processos d'integració del codi i desplegament de projectes [23].

# Capítol 6

## Planificació

Un cop definit l'abast del projecte, la següent passa correspon a definir la planificació per completar l'abast. En aquest capítol es tractarà la planificació total del projecte. La planificació engloba el conjunt de tasques a realitzar i els recursos necessaris per a desenvolupar aquestes tasques. En el següent capítol es parlarà dels riscos associats a la planificació descrita.

Per començar, cal remarcar que aquest projecte es duu a terme sota la modalitat B (empresa) i, per tant, la longitud del projecte com les seves dades d'inici i fi venen condicionades per les dades del contracte acordades entre Peersyst i Nervos.

Aquesta planificació es dividirà en franges de treball de dues setmanes, anomenades *sprints*, on es desenvoluparan, revisaran i milloraran les tasques. Així doncs, l'espai temporal que comprèn el desenvolupament del projecte és el següent:

- **Sprint 1:** del 23 de gener al 3 de febrer.
- **Sprint 2:** del 6 de febrer al 17 de febrer.
- **Sprint 3:** del 20 de febrer al 3 de març.
- **Sprint 4:** del 6 de març al 17 de març.
- **Sprint 5:** del 20 de març al 31 de març.
- **Sprint 6:** del 3 d'abril al 14 d'abril.

Prèviament a la descripció de les tasques, el nom **CKBull Developer Console** farà referència a la plataforma que utilitzaran els desenvolupadors de dApps per enregistrar les seves dApps i realitzar les peticions entre aquestes i els usuaris de la cartera CKBull Wallet.

### 6.1 Descripció de les tasques

En tractar-se d'un projecte que comprèn diferents àrees de desenvolupament s'ha decidit agrupar les tasques segons l'àrea a la qual pertanyi. Cada tasca tindrà un identificador format per l'àrea a què pertany i el número que s'ha assignat.

- **Gestió i documentació del projecte (GDP):** referència a totes les tasques i processos de gestió del projecte i documentació d'aquest.

- **Desenvolupament de CKBull Developer Console (DCDC):** referència a totes les tasques de disseny i implementació de la plataforma per enregistrar dApps i enviament de peticions entre dApps i usuaris (de CKBull Wallet).
- **Desenvolupament de CKBull Wallet (DCW):** referència a totes les tasques d'implementació de l'aplicació mòbil CKBull Wallet.

### 6.1.1 Gestió i documentació del projecte

- **GDP1 - Contextualització i abast:** Redactar els conceptes previs i la introducció del projecte per després definir els objectius principals del projecte, així com els requisits funcionals i no funcionals.
- **GDP2 - Anàlisi temporal:** identificar i analitzar les fases del projecte i definir les tasques necessàries a realitzar per a la finalització d'aquest.
- **GDP3 - Sostenibilitat:** investigació i avaluació dels impactes mediambientals i socials que suposa el desenvolupament i la producció del projecte.
- **GDP4 - Sostenibilitat econòmica:** avaluar i definir els costos econòmics que comporta el projecte i analitzar la viabilitat d'aquest.
- **GDP5 - Documentació del projecte:** redacció de la memòria del Treball Final de Grau.
- **GDP6 - Defensa del projecte:** preparació de la defensa del Treball Final de Grau.

### 6.1.2 Desenvolupament de CKBull Developer Console

#### Configuració del projecte

- **DCDC1.1 - Configuració base UI:** creació del projecte base, instal·lació de llibreries i dependències i configuració de la UI <sup>1</sup> [24].
- **DCDC1.2 - Configuració base API:** creació del projecte base i instal·lació de llibreries necessàries per al desenvolupament de l'API <sup>2</sup> [25].

#### Sistema de creació d'usuaris

- **DCDC2.1 - Creació, obtenció, edició i eliminació d'usuaris:** implementar les crides necessàries per a poder crear un usuari, editar-lo i eliminar-lo de la base de dades, obtenir la seva informació tant per un com per molts usuaris.
- **DCDC2.2 - Procés d'autenticació i d'autorització:** implementar un sistema d'autenticació i autorització respecte als usuaris, limitant les seves accions dins del sistema respecte als recursos i definint rols d'autorització.
- **DCDC2.3 - Formularis d'inici de sessió i enregistrament:** implementar els formularis que permetin als usuaris iniciar sessió i enregistrar-se dins de la plataforma.
- **DCDC2.4 - Vistes d'inici de sessió i enregistrament:** Vistes d'inici de sessió i enregistrament d'usuaris.

---

<sup>1</sup> *UI: User Interface.*

<sup>2</sup> *API: Application Programming Interface.*

- **DCDC2.5 - Securització (1):** investigació i implementació de mecanismes de seguretat per millorar la robustesa del sistema.

### Sistema de creació de dApps

- **DCDC3.1 - Creació, obtenció, edició i eliminació de *dApps*:** implementar les crides necessàries per a poder enregistrar una *dApp*, editar la seva informació i eliminar-la de la base de dades, obtenir la seva informació tant per un com per moltes dApps creades pel mateix desenvolupador [9].
- **DCDC3.2 - Formularis CRUD de les *dApps*:** implementar els formularis que permetin crear, editar, eliminar i visualitzar el contingut de *dApps*.
- **DCDC3.3 - Vistes per interactuar amb les *dApps*:** implementar vistes que utilitzin els formularis anteriors perquè l'usuari pugui navegar entre aquestes.
- **DCDC3.4 - Obtenció i regeneració de credencials:** implementar les crides necessàries per a poder aconseguir les credencials API d'una *dApp* i regenerar-les a petició de l'usuari.
- **DCDC3.5 - Securització (2):** investigació i implementació de mecanismes de seguretat per millorar la robustesa del sistema.

### Sistema de peticions de transaccions

- **DCDC4.1 - Disseny i implementació de sistema de peticions:** implementar les crides per a poder generar peticions d'inici de sessió i peticions de transacció com a *dApps* sota demanda.
- **DCDC4.2 - Polling d'estat de les peticions:** implementar un sistema de *polling* o consultes constants per permetre a les *dApps* obtenir l'estat d'una petició (inici de sessió o transacció).
- **DCDC4.3 - Securització (3):** investigació i implementació de mecanismes de seguretat per millorar la robustesa del sistema.

### Pas a producció

- **DCDC5.1 - Desplegament del projecte:** portar tant la plataforma CKBull Developer Console i l'aplicació CKBull Wallet amb les noves funcionalitats a l'entorn de producció a la fase final del projecte.

### 6.1.3 Desenvolupament de CKBull Wallet

#### Sistema de firma de transaccions

- **DCW6.1 - Redisseny dels components necessaris a la UI:** redissenyar mitjançant estils els components necessaris dins de l'aplicació de *CKBull Wallet*.
- **DCW6.2 - Creació dels nous components de l'aplicació:** definir i implementar els nous components de la UI necessaris per a poder signar o rebutjar peticions procedents d'una *dApp*.

- **DCW6.3 - Obtenir, acceptar o declinar peticions:** implementar les vistes necessàries perquè l'usuari de CKBull Wallet pugui visualitzar la informació de cada petició, el seu estat i pugui acceptar o rebutjar aquestes.

## 6.2 Recursos

A més de definir les tasques que s'han de realitzar, també és necessari estimar quins recursos i de quin tipus s'hauran d'utilitzar. Les següents taules representen tots els recursos previstos que s'ha estimat que s'usaran durant el procés de desenvolupament del projecte. Els recursos es classifiquen segons la categoria a la qual pertanyen, que poden ser: **humans, físics, espacials i digitals**.

### Recursos humans

Dintre d'aquests recursos es troben totes les persones involucrades dins del projecte. No obstant això, dintre d'aquesta categoria és dos tipus diferents de recursos: **recursos humans interns i recursos humans externs**. Els recursos humans interns són tots aquells recursos que interactuen de manera directa amb el TFG, mentre que els externs actuen de manera indirecta sobre el projecte.

Recursos humans interns		
Sigles	Rol	Persona
DP	Desenvolupador principal	Guillem García Gil
CP	<i>Project Manager</i>	Adrià Carrera Mas
T	Tutor del projecte	Ferran Prat Tio
TP	Tutor ponent	Carles Farré Tost
TGEP	Tutor de GEP	Natalia de Fátima Sánchez Arrieta

Taula 2: Recursos humans interns.

Recursos humans externs	
Sigles	Rol
URA	Urano Studio

Taula 3: Recursos humans externs.

El recurs humà descrit a la Taula 3 correspon a una empresa de disseny de producte anomenada Urano Studio [26]. La seva relació del projecte consta de la necessitat d'encarregar dissenys per a les noves funcionalitats per a les interfícies tant de CKBull Developer Console com de CKBull Wallet.

### Recursos físics

Els recursos físics fan referència a tot el material *hardware* utilitzat durant el projecte.

Recursos físics	
Sigles	Recurs
FPMP	Macbook Pro M1 PRO 14" (2022) [27]
FT	Apple Magic Keyboard [28]
FR	Ratolí Logitech MX Master [29]
FS1	Servidor d'AWS per l'entorn de <i>staging</i>
FS2	Servidor d'AWS magatzem d'imatges
FS3	Servidor d'AWS per l'entorn de producció

Taula 4: Recursos físics.

### Recursos espaials

La metodologia de treball de Peersyst consta de la realització de la jornada de manera remota. Per tant, gran part de la fase de desenvolupament es farà a casa. Això no obstant, per a poder simular un entorn semblant al que seria una oficina, s'ha decidit afegir un espai de *coworking*<sup>3</sup>, als recursos per a després poder fer una estimació pressupostària més realista. La següent taula mostra els recursos espacials utilitzats.

Recursos espaials	
Sigles	Recurs
IDT	Instal·lacions de treball <i>coworking</i>

Taula 5: Costs sobre recursos espaials.

### Recursos digitals

Per finalitzar, els recursos digitals faran referència a tot el *software* necessari per a poder desenvolupar el projecte. Dins d'aquesta categoria es troben tant recursos per poder desenvolupar *software* com editors de text per a redactar la memòria.

Recursos digitals	
Sigles	Recurs
DIJ	IntelliJ IDEA: IDE utilitzat per al desenvolupament del projecte [30].
DG	Github: Eina online de control de versions i allotjament de repositoris [31].
DGP	Github Projects: software utilitzat per a mantenir control sobre les tasques [21].
DGA	Github Actions: software de CI utilitzat al projecte [23].
DSL	Slack: Eina de comunicació utilitzada a Peersyst [19].
DNV	Navegadors web: Google Chrome, Safari i Mozilla Firefox.
DO	Overleaf: editor online de latex.

Taula 6: Recursos digitals.

---

<sup>3</sup>Coworking: Espai físic compartit entre diferents empreses, autònoms, i altres professionals on es realitza una jornada laboral.

### **6.3 Resum**

Un cop detallades les tasques i els recursos, a continuació es mostra a la següent taula el resum general de la planificació, incloent-hi les hores, dependències entre tasques i els recursos utilitzats per cada una. Per a una visualització més detallada de la planificació temporal, la Figura 4 mostra la planificació temporal.

ID	Nom	Hores	Dependències	Recursos
<b>GP</b>	<b>Gestió de projecte</b>	<b>162</b>		
GDP1	Contextualització i abast	20		DP, T, TP, FPMP, FT, FR, O, DNV, DSL
GDP2	Anàlisi temporal	12	GDP1	DP, T, TP, SM, FPMP, FT, FR, O, DNV, DSL
GDP3	Sostenibilitat	10	GDP2	DP, T, TP, FPMP, FT, FR, O, DNV, DSL
GDP4	Sostenibilitat econòmica	10	GDP2	DP, T, TP, FPMP, FT, FR, O, DNV, DSL
GDP5	Documentació del projecte	90		DP, T, TP, FPMP, FT, FR, O, DNV, DSL
GDP6	Defensa del projecte	20		DP, T, TP, FPMP, FT, FR, O, DNV, DSL
<b>DCDC1</b>	<b>Configuració base</b>	<b>12</b>		
DCDC1.1	Configuració base UI	6		DP, SM, FPMP, FT, FS1, FR, DIJ, DG, DGP, DGA, DSL, DNV
DCDC1.2	Configuració base API	6		DP, SM, FPMP, FT, FS1, FR, DIJ, DG, DGP, DGA, DSL, DNV
<b>DCDC2</b>	<b>Sistema de creació d'usuaris</b>	<b>90</b>		
DCDC2.1	Creació, obtenció, edició i eliminació d'usuaris	24		DP, FPMP, FT, FS1, FR, DIJ, DG, DGP, DGA, DSL, DNV
DCDC2.2	Procés d'autenticació i d'autorització	24	DCDC2.1	DP, FPMP, FT, FS1, FR, DIJ, DG, DGP, DGA, DSL, DNV
DCDC2.3	Formularis d'inici de sessió i enregistrament	16	DCDC2.2	DP, FPMP, FT, FS1, FR, DIJ, DG, DGP, DGA, DSL, DNV
DCDC2.4	Vistes d'inici de sessió i enregistrament	16	DCDC2.3	DP, FPMP, FT, FS1, FR, DIJ, DG, DGP, DGA, DSL, DNV
DCDC2.5	Securització (1)	10	DCDC2.2	DP, SM, FPMP, FT, FS1, FR, DIJ, DG, DGP, DGA, DSL, DNV
<b>DCDC3</b>	<b>Sistema de creació de dApps</b>	<b>90</b>		
DCDC3.1	Creació, obtenció, edició i eliminació de dApps	16	DCDC1.2	DP, FPMP, FT, FS1, FR, DIJ, DG, DGP, DGA, DSL, DNV
DCDC3.2	Formularis CRUD de les dApps	20	DCDC3.4	DP, FPMP, FT, FS1, FR, DIJ, DG, DGP, DGA, DSL, DNV
DCDC3.3	Vistes per interactuar amb les dApps	20	DCDC2.4, DCDC3.2	DP, FPMP, FT, FS1, FR, DIJ, DG, DGP, DGA, DSL, DNV
DCDC3.4	Obtenció i regeneració de credencials	24	DCDC3.1	DP, FPMP, FT, FS1, FR, DIJ, DG, DGP, DGA, DSL, DNV
DCDC3.5	Securització (2)	10	DCDC3.3,	DP, SM, FPMP, FT, FS1, FR, DIJ, DG, DGP, DGA, DSL, DNV
<b>DCDC4</b>	<b>Sistema de peticions de transaccions</b>	<b>54</b>		
DCDC4.1	Disseny i implementació de sistema de peticions	30		DP, FPMP, FT, FS1, FR, DIJ, DG, DGP, DGA, DSL, DNV
DCDC4.2	Polling d'estat de les peticions	12	DCDC4.1	DP, FPMP, FT, FS1, FR, DIJ, DG, DGP, DGA, DSL, DNV
DCDC4.3	Securització (3)	12	DCDC4.2	DP, SM, FPMP, FT, FS1, FR, DIJ, DG, DGP, DGA, DSL, DNV
<b>DCDC5</b>	<b>Pas a producció</b>	<b>8</b>		
DCDC5.1	Desplegament del projecte	8	DCDC3.4, DCDC4.3, DCW1.3	DP, FPMP, FT, FR, FS1, FS2, FS3, DIJ, DG, DGP, DGA, DSL, DNV
<b>DCDW1</b>	<b>Sistema de firma de transaccions</b>	<b>82</b>		
DCDW1.1	Redisseny dels components necessaris a la UI	30		DP, FPMP, FT, FS1, FR, DIJ, DG, DGP, DGA, DSL, DNV
DCDW1.2	Creació dels nous components de l'aplicació	28		DP, FPMP, FT, FS1, FR, DIJ, DG, DGP, DGA, DSL, DNV
DCDW1.3	Obtenir, acceptar o declinar peticions	24	DCDW1.1, DCDW1.2	DP, FPMP, FT, FS1, FR, DIJ, DG, DGP, DGA, DSL, DNV

Taula 7: Resum de les tasques.

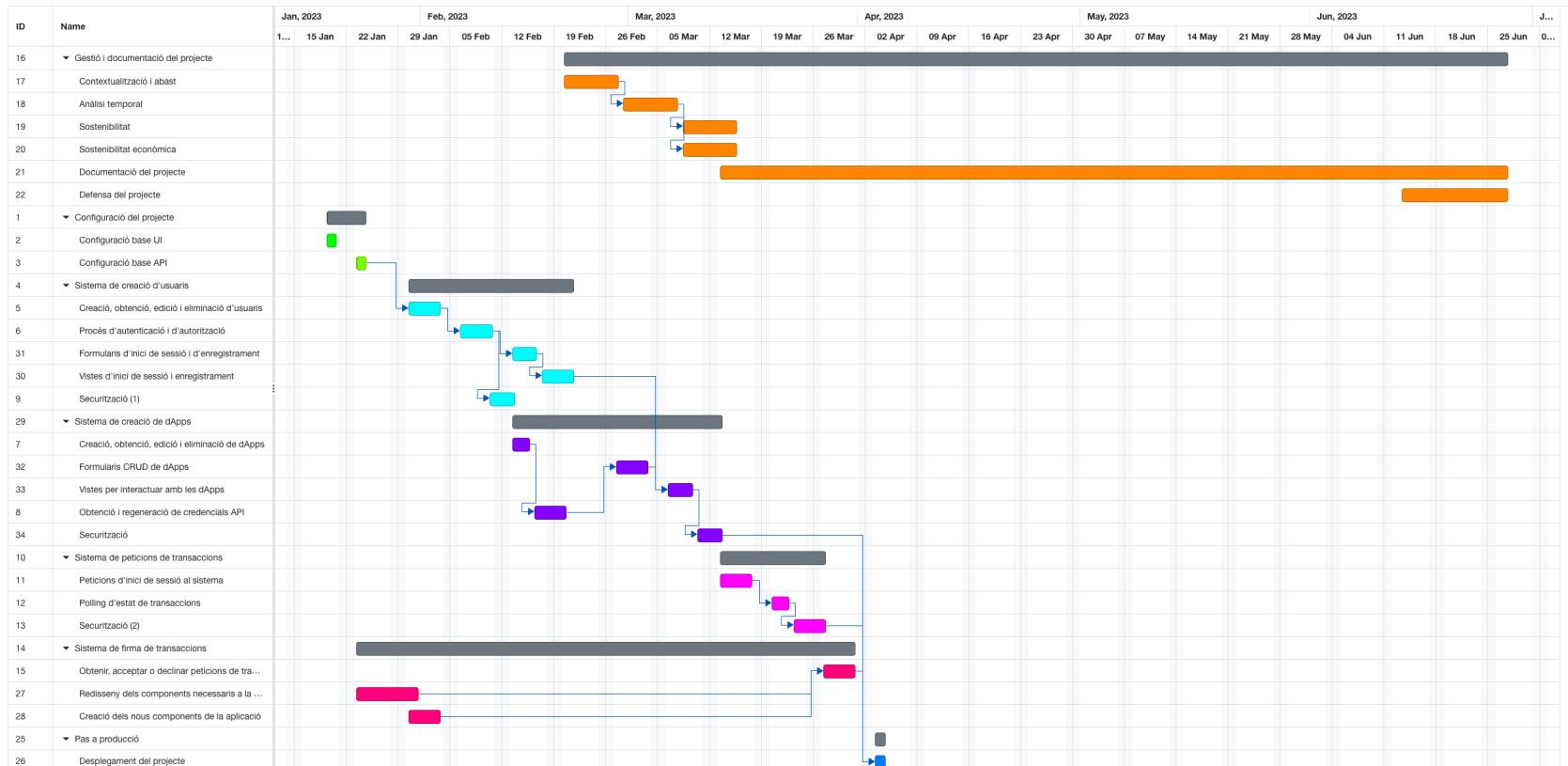


Figura 4: Diagrama de Gantt

Font: elaboració pròpria.

# Capítol 7

## Riscos

Per a poder dur a terme el projecte segons la planificació feta és important ser conscients dels possibles riscos que assumim o ens podem trobar. Aquests poden ser generats per la mateixa planificació o ser factors externs al projecte. A continuació, es presenta quins riscos s'han detectat, quin impacte poden tindre sobre el projecte i com es respondrà davant d'aquests.

Riscos detectats			
Risc	Impacte sobre el projecte	Percentatge	Resposta
Manca o error a la definició de tasques	Alt	70%	Dedicar temps i esforç en una bona planificació.
Limitacions de coneixement i limitacions tècniques	Mitjà	40%	Estudiar quines àrees de coneixement envolten el projecte i profunditzar a les necessàries.
Aparició d'errors i bugs	Mitjà	40%	Realitzar testing unitari i d'integració per a verificar el funcionament correcte.
Salut	Mitja / Alt	60%	Redistribuir la prioritat de les tasques segons el temps restant.

Taula 8: Riscos detectats en el projecte.

Tots els riscos especificats a la Taula 8 fan referència als riscos definits a la Secció 4.4. Com es pot observar, gran part dels riscos tenen un impacte mitjà/alt sobre el projecte, raó per la qual s'han de tindre clars els mecanismes de resposta en cas que succeeixin. Els tres primers compten com a riscos generats amb la planificació o interns del projecte, mentre que la malaltia és un factor extern difícil de prevenir i, per tant, la resposta contra aquesta serà la menys efectiva respecte als altres riscos. Així doncs, s'haurà de fer més èmfasi en els tres primers.

# Capítol 8

## Pressupost

Per a poder estimar quin cost econòmic comporta la solució del projecte, en aquest capítol, s'ha realitzat un pressupost especificant els costos de cada recurs definits a la Secció 6.2.

### 8.1 Identificació dels costos

En aquesta secció s'identifiquen tots els costos del projecte. Tots els costos i preus dins d'aquest document seran mostrats en la divisa Euro (€). Els costos estan agrupats segons la categoria o grup al qual pertanyen.

- **Costos sobre recursos humans interns**
- **Costos sobre recursos humans externs**
- **Costos sobre recursos físics**
- **Costos sobre recursos espacials**
- **Costos sobre recursos digitals**

Els costos resultants de les següents taules han sigut calculats de la següent manera. Respecte als **recursos humans interns i recursos humans externs** s'ha calculat:

$$\text{Cost total} = \text{Sou hora brut} \times \text{Cost SS} \times \text{hores de desenvolupament}$$

Tenint en compte com a *Cost SS* un factor multiplicatiu de 1.33 per a simplificar el càlcul. Respecte als **recursos digitals i espacials**, s'ha calculat el cost total segons la tarifa del recurs pel temps d'ús d'aquest. Es pot veure representat com:

$$\text{Cost total} = \text{Tarifa del recurs} \times \text{temps d'ús}$$

Finalment, l'amortització sobre els **recursos físics hardware** s'ha calculat a partir de la següent fórmula (on la vida útil seran quatre anys, dies laborals 220 i 4 hores diàries):

$$\text{Amortització} = \frac{\text{Cost del recurs}}{\text{Vida útil (anys)} \times \text{dies laborals} \times \text{hores diàries}} \times \text{hores de projecte}$$

### 8.1.1 Costos sobre recursos humans interns

Els únics actors que participaran en el desenvolupament de les tasques del projecte seran l'autor principal d'aquest TFG, amb un rol de desenvolupador *full-stack* junior, i el cap de projecte. Els sous mostrats a la Taula 9 han sigut extrets de la web [www.talent.com](http://www.talent.com) [32] per a obtenir una referència del sou hora brut dels actors involucrats. A més, cal tindre en compte el cost de la seguretat social que, per simplificació, serà el resultat de multiplicar el sou brut per 1.3.

Sigles	Rol	Sou hora brut	Afegit SS
DP	Desenvolupador <i>full-stack</i> junior	13,33 €	17,73 €
CP	<i>Project Manager</i>	18,46 €	24,55 €

Taula 9: Costos sobre recursos humans interns.

Així doncs, el cost total dels recursos humans interns es mostra a la següent taula amb la quantitat d'hores de les tasques.

ID	Nom	Hores	DP	PM	Total
<b>GP</b>	<b>Gestió de projecte</b>	<b>162</b>	<b>162</b>	<b>0</b>	<b>2.872,26€</b>
GDP1	Contextualització i abast	20	20	0	354,60€
GDP2	Anàlisi temporal	12	12	0	212,76€
GDP3	Sostenibilitat	10	10	0	177,30€
GDP4	Sostenibilitat econòmica	10	10	0	177,30€
GDP5	Documentació del projecte	90	70	0	478,71€
GDP6	Defensa del projecte	20	20	0	354,60€
<b>DCDC1</b>	<b>Configuració base</b>	<b>16</b>	<b>12</b>	<b>4</b>	<b>310,96€</b>
DCDC1.1	Configuració base UI	8	6	2	155,48€
DCDC1.2	Configuració base API	8	6	2	155,48€
<b>DCDC2</b>	<b>Sistema de creació d'usuaris</b>	<b>96</b>	<b>90</b>	<b>6</b>	<b>1.743,02€</b>
DCDC2.1	Creació, obtenció, edició i eliminació d'usuaris	24	24	0	425,52€
DCDC2.2	Procés d'autenticació i d'autorització	28	24	4	523,72€
DCDC2.3	Formularis d'inici de sessió i enregistrament	16	16	0	283,68€
DCDC2.4	Vistes d'inici de sessió i enregistrament	16	16	0	283,68€
DCDC2.5	Securització (1)	12	10	2	226,40€
<b>DCDC3</b>	<b>Sistema de creació de dApps</b>	<b>94</b>	<b>90</b>	<b>4</b>	<b>1.493,90€</b>
DCDC3.1	Creació, obtenció, edició i eliminació de dApps	16	16	0	283,68€
DCDC3.2	Formularis CRUD de les dApps	20	20	0	354,60€
DCDC3.3	Vistes per interactuar amb les dApps	20	20	0	354,60€
DCDC3.4	Obtenció i regeneració de credencials API	26	24	2	474,62€
DCDC3.5	Securització (2)	12	10	2	226,40€
<b>DCDC4</b>	<b>Sistema de peticions de transaccions</b>	<b>64</b>	<b>54</b>	<b>10</b>	<b>1.202,92€</b>
DCDC4.1	Peticions d'inici de sessió al sistema	34	30	4	630,10€
DCDC4.2	Polling d'estat de transaccions	14	12	2	261,86€
DCDC4.3	Securització	16	12	4	310,96€
<b>DCDC5</b>	<b>Pas a producció</b>	<b>12</b>	<b>8</b>	<b>4</b>	<b>240,04€</b>
DCDC5.1	Desplegament del projecte	12	8	4	240,04€
<b>DCDW1</b>	<b>Sistema de firma de transaccions</b>	<b>86</b>	<b>82</b>	<b>4</b>	<b>1.552,06€</b>
DCDW1.1	Redisseny dels components necessaris a la UI	32	30	2	581,00€
DCDW1.2	Creació dels nous components de l'aplicació	28	28	0	496,44€
DCDW1.3	Obtenir, acceptar o declinar peticions de transaccions	26	24	2	474,62€
<b>Total:</b>					<b>9.415,16€</b>

Taula 10: Resum de les tasques.

### 8.1.2 Costos sobre els recursos humans externs

Per al desenvolupament del projecte, l'empresa on es realitza el projecte ha contractat els serveis de l'empresa de disseny Urano Studio per obtenir dissenys UI<sup>1</sup>/UX<sup>2</sup> per CKBull Developer Console i CKBull Wallet. El sou hora mostrat a la Taula 11 també ha sigut obtingut de [www.talent.com](http://www.talent.com) [32].

<sup>1</sup>UI: *User interface*

<sup>2</sup>UX: *User experience*

Sigles	Rol	Sou hora brut	Hores	Cost total
DUI	Dissenyadors UI/UX	16,41 €	30	492,30 €

Taula 11: Costos sobre recursos humans externs.

### 8.1.3 Costos sobre els recursos físics

A continuació, es llisten tots els recursos físics utilitzats en el projecte. Dins d'aquesta categoria únicament es fa referència als recursos *hardware*. A més, també es mostra l'amortització de cada recurs durant el temps de desenvolupament.

Sigles	Recurs	Cost total	Amortització
FPMP	Macbook Pro M1 PRO 14" (2022) [27]	2.449,00 €	303,85€
FT	Apple Magic Keyboard [28]	109,00 €	13,25€
FR	Ratolí Logitech MX Master [29]	84,99 €	10,33€
FS1	Servidor d'AWS entorn <i>staging</i>	-	144,9 €
FS2	Servidor d'AWS magatzem d'imatges	0€ (cost negligible)	-
FS3	Servidor d'AWS entorn producció	-	119,93 €
<b>Total:</b>			<b>592,26 €</b>

Taula 12: Costos sobre recursos físics.

Per a calcular els costos dels recursos d'AWS, s'ha usat l'eina Estimator per a fer una estimació dels costos que comportaria utilitzar servidor per a *staging* durant 5 mesos i un servidor per a producció per a un mes de desenvolupament [33].

### 8.1.4 Costos sobre els recursos espacials

Com s'ha descrit prèviament, s'ha tingut com a referència de cost l'espai de *coworking* **Crec Gràcia** [34] situat a Barcelona per a poder simular un cost espacial.

Sigles	Recurs	Cost mensual	Temps (mes)	Cost total
IDT	Instal·lacions de treball	195,00 €	5	975,00 €

Taula 13: Costs sobre recursos espacials.

### 8.1.5 Costos sobre els recursos digitals

Per acabar, a la Taula 14 s'esmenten tots els recursos digitals o de *software* utilitzats durant el desenvolupament del projecte. Degut a que alguns dels recursos només ofereixen una tarifa fixa, s'ha calculat el cost segons els temps en mesos que es necessitarà el recurs.

### 8.1.6 Resum dels costos

A mode de resum, s'adjunta la Taula 15 amb el cost total del projecte desglossat en els 5 tipus diferents de costos.

Sigles	Recurs	Tarifa	Temps (mes)	Cost total
DIJ	IntelliJ IDEA [30]	59,90 €/mes	4	239,60 €
DG, DGP, DGA	Ecosistema Github [31]	4,00 €/mes	6	24,00 €
DSL	Slack	0€	4	0€
DNV	Google Chrome	0€	4	0€
DO	Overleaf	0€	4	0€
<b>Total:</b>				<b>263,60€</b>

Taula 14: Costos sobre recursos digitals.

Tipus	Subtotal
Costos de recursos humans interns	9.415,16 €
Costos de recursos humans externs	492,30 €
Costos de recursos físics <i>hardware</i>	592,26 €
Costos de recursos espaials	975,00 €
Costos dels recursos digitals	263,60 €
<b>Total:</b>	<b>11.738,32 €</b>

Taula 15: Resum dels costos.

## 8.2 Estimació dels costos

A partir de la identificació feta a l'apartat anterior, es farà l'estimació de costos del projecte tenint en compte els riscos associats al projecte.

### 8.2.1 Costos de contingència

Per a calcular el cost de contingència del projecte, partirem del resultat final dels costos a la Taula 15. A més, es definirà com a factor de contingència un 15% del cost total del projecte. Aquest percentatge ve donat a causa del marge de temps disponible per a desenvolupar el projecte (entre 6-7 *sprints*) i la possible incertesa que es pot trobar durant la fase de desenvolupament, a causa que el nivell d'especificació ha sigut lleu.

Tipus de cost	Cost general	General amb contingència
Costos de recursos humans interns	9.415,16 €	10.827,43 €
Costos de recursos humans externs	492,30 €	566,15 €
Costos de recursos físics <i>hardware</i>	592,26 €	681,10 €
Costos de recursos espaials	975,00 €	1.121,25 €
Costos dels recursos digitals	263,60 €	303,14 €
<b>Total:</b>		<b>13.499,07 €</b>

Taula 16: Costos de contingència.

### 8.2.2 Costos associats als riscos

A la Taula 17 es pot trobar els costos associats als riscos explicats al capítol anterior. Per a donar un valor numèric al tipus d'impacte, s'adjunta una estimació en hores de temps s'estima que comporti corregir l'impacte del risc.

Riscos detectats			
Risc	Impacte sobre el projecte	Hores	Cost total
Manca o error a la definició	Alt	20	212,76€
Limitacions tècniques	Mitjà	15	106,38€
Aparició imprevista d'errors i bugs	Mitjà	20	141,84€
Salut	Mitja / Alt	0	0,00€
<b>Total:</b>			<b>460,98€</b>

Taula 17: Riscos detectats en el projecte.

### 8.2.3 Pressupost final

Finalment, la següent taula mostra l'estimació del pressupost final del projecte.

Tipus	Cost
Cost general	11.738,32 €
Cost de contingència	1.720,91 €
Cost de riscos	460,98 €
<b>Total:</b>	<b>13.920,21 €</b>

Taula 18: Pressupost final.

## 8.3 Desviacions sobre els costos del projecte

Per finalitzar aquest capítol, definirem el sistema de control que es realitzarà sobre els costos del projecte. Cada indicador serà posteriorment utilitzat per mantenir un seguiment de l'estimació pressupostaria, amb els valors reals, fins al final del projecte.

### Desviació temporal d'una tasca

$$\text{horas reales} - \text{horas estimadas}$$

### Desviació econòmica d'una tasca

$$(horas reales - horas estimadas) \cdot \text{costo de la tarea}$$

### Desviació total de costos humans

$$\text{costo de recursos humanos real} - \text{costo de recursos humanos estimado}$$

### Desviació total de costos físics hardware

*cost de recursos físics hardware real – cost de recursos físics hardware estimats*

#### **Desviació total de costos digitals**

*cost de recursos digitals real – cost de recursos digitals estimats*

#### **Desviació total de costos espaiials**

*cost de recursos espaiials real – cost de recursos espaiials estimats*

#### **Desviació de costos generals**

*cost general real – cost general estimats*

#### **Desviació de costos de contingència (%)**

*cost de contingència real – cost de contingència estimat*

#### **Desviació de costos de riscos**

*cost de riscos real – cost de riscos estimat*

#### **Desviació total del costos**

*cost total real – cost total estimat*

#### **Desviació total temporal**

*temps total real – temps total estimat*

El motiu de la creació de les desviacions llistades anteriorment és poder avaluar quins paràmetres temporals i monetaris es veuen afectats durant el desenvolupament del projecte i com es desvien de la previsió inicial. Com a referència, es faran 2 mesures: una a l'inici del projecte i una final un cop es finalitzi el projecte.

# Capítol 9

## Especificació

Un cop fet l'anàlisi del sistema a desenvolupar, iniciarem la fase d'especificació. Dins d'aquesta fase es detallarà en major profunditat el problema a desenvolupar, especificant quins actors s'involucren dins del sistema, quines accions poden fer, amb quines entitats interactuaran i com es produirà la interacció. Per aconseguir-ho, s'han utilitzat diferents eines com el diagrama de casos d'ús o l'esquema conceptual de dades.

No obstant això, per a poder dur a terme una especificació adient, cal entendre el domini del problema. Tot i haver descrit diferents conceptes com la *blockchain*, els comptes i els tokens, és necessari saber quin concepte clau és sobre el que funciona la solució. Aquest concepte clau són les transaccions de la *blockchain* CKB. Així doncs, la següent secció explica de manera acotada, però detallada com funcionen les transaccions d'actius digitals a CKB.

### 9.1 Descripció tècnica de les transaccions a CKB

La transacció, com a concepte quotidià, correspon a l'intercanvi pactat d'un actiu entre dues parts interessades. Per exemple, una transacció bancària és l'intercanvi d'una quantitat de diners entre dos comptes.

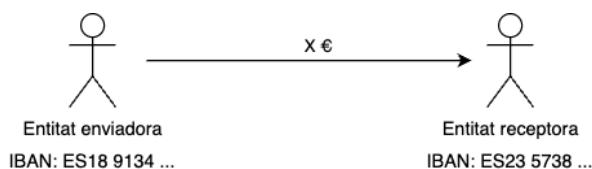


Figura 5: Transferència simplificada entre dos entitats.  
Font: elaboració pròpia.

Cada entitat que conforma una transacció disposa d'un actiu a transferir i un identificador (direcció). No obstant això, la manera en la qual es gestionen les dades pot variar. Per exemple, un banc disposa de tota la seva informació de forma centralitzada amb una estructura A mentre que un altre banc té la seva informació en una estructura B.

La *blockchain* CKB té la seva pròpia manera de gestionar els actius i la seva transferència. CKB adopta el model UTXO per a la realització de les transaccions. Com s'ha descrit al Capítol 1, els

UTXO permeten poder determinar quins actius poden transferir-se i quin compte és autoritzat per a transferir-los, evitant problemes com el doble pagament.

A continuació, s'explicarà pas a pas quins camps (estructura de dades) necessita una transacció per a poder afegir-se a la *blockchain*. Tota la informació ha sigut extreta del RFC 0022 CKB Transaction Structure [35] generat per Nervos.

### Estructura d'una transacció

Inicialment, una transacció ha de contenir tant la informació d'una o més transaccions prèvies com a entrada, que anomenarem *inputs*, i els resultats de la transacció pròpia, que anomenarem *outputs*. Aquests dos conceptes són els mateixos utilitzats pel model UTXO.

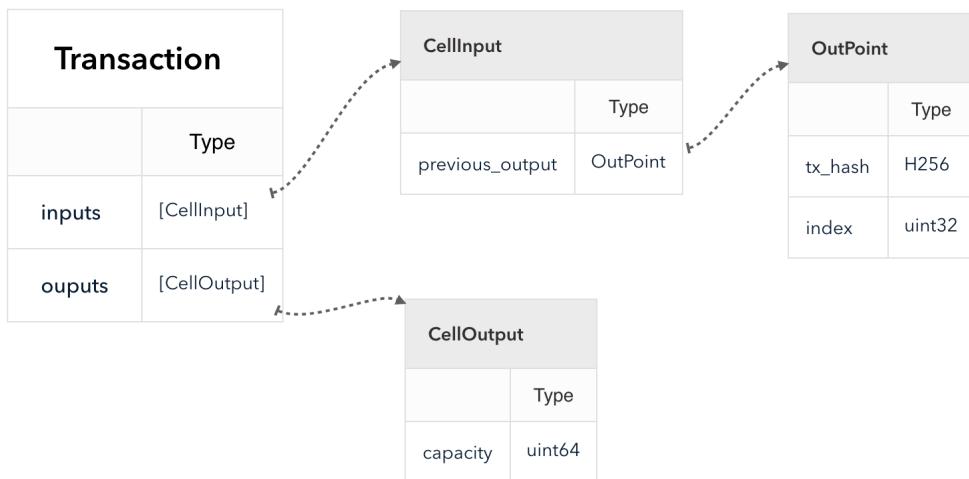


Figura 6: Diagrama simplificat d'una transacció.  
Font: RFC 0022 CKB Transaction Structure.

Com es pot veure a la Figura 6 una transacció té com a camps *inputs* i *outputs* que segueixen les següents estructures de dades:

- **OutPoint:** Estructura de dades que referencia una transacció prèvia dins d'un bloc de la xarxa. Conté dos camps, *tx\_hash* i *index*. El camp *tx\_hash* fa referència al *hash* de la transacció (identificador d'una transacció ja existent) i el camp *index* fa referència a la posició on es troba el *CellOutput* de la transacció dins del seu vector *outputs*. La figura 7 mostra com l'outpoint de la *cell1* fa referència a un *output* d'una transacció prèvia.

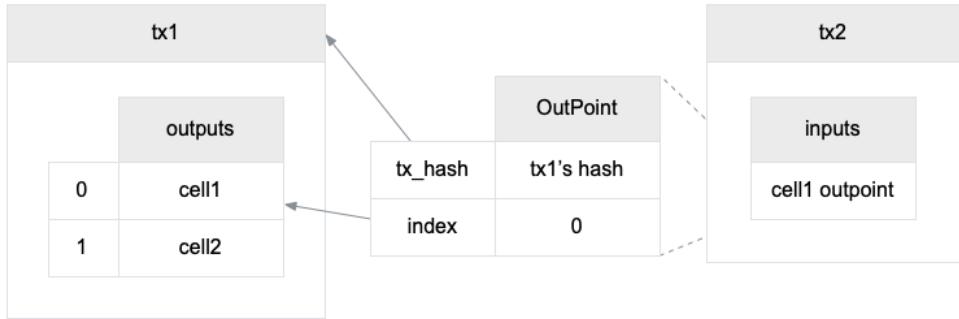


Figura 7: Diagrama del funcionament d'un OutPoint.

Font: RFC 0022 CKB Transaction Structure.

- **CellInput:** Estructura de dades que conté un *OutPoint* referenciant la localització del *output* d'una transacció dins de la *blockchain* que s'utilitzarà com a *input* a la nova transacció.
- **CellOutput:** Estructura de dades que representarà l'*output* de la transacció. El camp *capacity* fa referència a la capacitat màxima d'informació que pot emmagatzemar un *output*.

Fins ara hem pogut veure com funcionen les referències a les transaccions anteriors. Però encara queda la part més important, que és com s'adjunten els actius o la informació dins de la transacció i com es representa. Així doncs, per **emmagatzemar la informació que contindrà la transacció** s'afegeix un nou camp, anomenat *outputs\_data*.

Transaction	
	Type
inputs	[CellInput]
outputs	[CellOutput]
outputs_data (*)	[Bytes]

(\*) New Fields

Figura 8: Diagrama parcial d'una transacció.

Font: RFC 0022 CKB Transaction Structure.

El camp *outputs\_data* correspon a un vector de *bytes* on la longitud d'aquest ha de ser la mateixa que la del camp *outputs*. D'aquesta manera es realitza una relació per posició entre la *CellOutput* i la informació d'aquest *output* al vector *outputs\_data*. És a dir, les dades associades a la *CellOutput* a la posició 0 del vector d'*outputs* es trobaran a la posició 0 del vector *outputs\_data* i així per a la resta del contingut del vector.

A més, s'ha de tindre en compte que la **quantitat de dades** d'una *CellOutput* no pot excedir la **capacity** definida dins d'aquesta. Altrament, la transacció no serà vàlida.

Per a finalitzar l'explicació de les transaccions, introduirem l'últim concepte rellevant. Aquest concepte correspon al ***lock***, una estructura de dades que utilitza la CKB VM (màquina d'estats que computa les transaccions) quan els *CellOutputs* són utilitzats com a *inputs* dins d'una nova transacció. La seva funcionalitat principal és **determinar qui pot usar** (*lock* o *unlock*) **els actius digitals** dels *CellOutputs*.

El bloqueig dels actius es realitza mitjançant la **signatura digital amb criptografia asimètrica**. L'estructura principal del *lock* correspon a l'anomenada ***Script*** que conté els camps ***code\_hash*** i ***hash\_type***, per a referenciar l'algoritme de criptografia asimètrica que ha d'usar la CKB VM i un camp anomenat ***args*** que conté la **clau pública** que referencia a la clau privada que fa la signatura. Per defecte l'algoritme fet servir és el **secp256k1**. A la Figura 9 es pot visualitzar l'estructura.

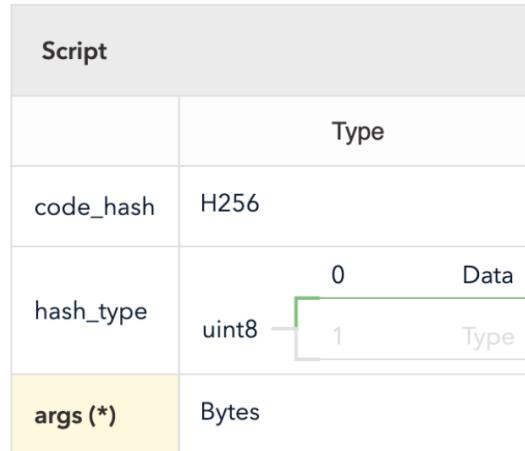


Figura 9: Estructura de dades del Script.  
Font: RFC 0022 CKB Transaction Structure.

La signatura de l'actiu té com a missatge a encriptar tota la transacció, i el resultat final es guarda dins d'un nou camp anomenat **witnesses** (testimonis) dins de la transacció. Aquest camp segueix la mateixa estructura de vector paral·lel, utilitzada pels *outputs* i els *outputs\_data*, amb el vector d'*inputs* ja que corresponen a les cel·les que executen els **locks** dels *inputs* per comprovar si l'actiu es pot desbloquejar o no.

Així doncs, la visió global de la transacció segons l'explicació prèvia es pot visualitzar a la Figura 10.

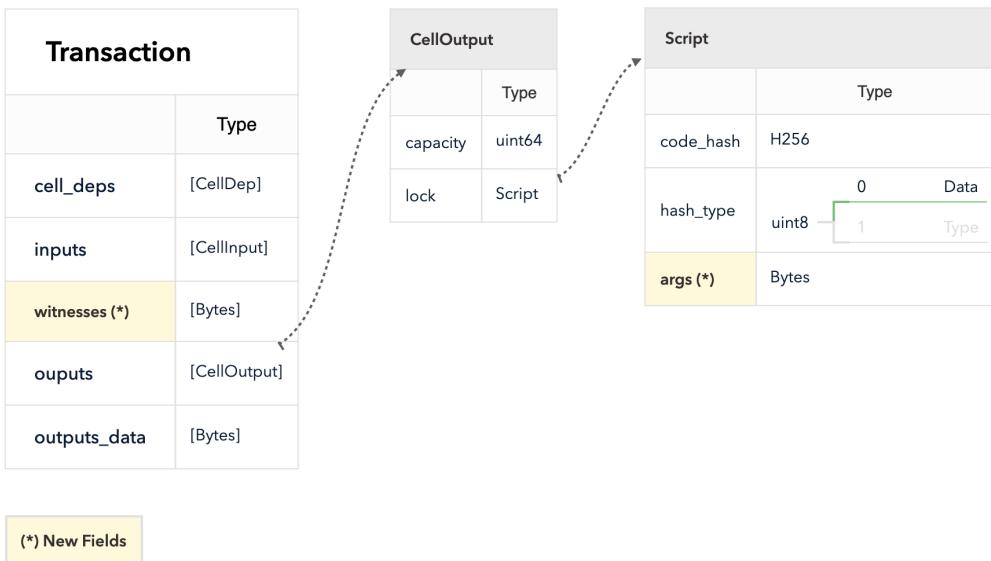


Figura 10: Diagrama parcial d'una transacció a CKB

Font: RFC 0022 CKB Transaction Structure.

Cal destacar que l'objecte transacció té més camps que no s'han inclòs dins de l'explicació pel fet que no són necessaris per entendre la funcionalitat del projecte. En cas de voler obtenir més informació, a les referències es troba el RFC corresponent a l'explicació.

Un cop entès com funcionen les transaccions a la *blockchain* CKB, ja podem començar a especificar la solució del projecte.

## 9.2 Casos d'ús del sistema

Comencem l'especificació del projecte analitzant quins casos d'ús hauria de tindre la solució. A mode d'introducció, un cas d'ús representa una **acció realitzada per un actor dins d'un sistema**.

En aquest capítol es definiran tots els casos d'ús principals que ha de satisfet el projecte per a garantir el resultat esperat definit als **requisits funcionals**.

Prèviament a la descripció dels casos d'ús, cal identificar quins actors, i de quin tipus, formen part del sistema.

### Actors

Dins del sistema trobarem dos tipus d'actors, els **actors primaris** (actors que interaccionen de manera directa amb el sistema) i els **actors secundaris** (actors que són utilitzats pel sistema i que no interaccionen de manera directa amb aquest). Dins del sistema es troben els següents actors:

<b>Actor</b>	<b>Tipus</b>	<b>Descripció</b>
Usuari CKBull Wallet	Primari	Usuari recurrent de la cartera CKBull Wallet
Desenvolupador dApp	Primari	Persona o entitat que desenvolupa dApp a la <i>blockchain</i> CKB
dApp	Secundari	Aplicació descentralitzada creada per <i>Desenvolupador dApp</i>

Taula 19: Actors dins del nostre sistema.

## Escenaris

Un cop definits quins actors té el sistema, cal determinar on realitzen les accions. L'escenari principal del sistema és **CKBull Signer App** i consta dels següents dos subescenaris:

- **CKBull Wallet:** Aplicació mòbil multiplataforma que actual com una cartera de criptomonedes sobre la *blockchain* CKB. Permet als usuaris interaccionar amb els seus comptes, tokens, actius, etc...
- **CKBull Developer Console:** *Software* que permet als desenvolupadors enregistrar les seves dApps al sistema perquè aquestes puguin usar la funcionalitat de signatura de peticions amb *CKBull Wallet*.

Un cop descrits els actors i els escenaris, passem a explicar els casos d'ús del sistema. Per a poder entendre de manera senzilla tots els casos d'ús s'ha decidit agrupar-los per escenari i àmbit. El diagrama complet de casos d'ús es pot trobar a l'annex Secció A.1. Alguns casos d'ús poden semblar més aviat estranys pel fet que, en el moment de la creació d'aquesta secció, ja es disposaven dissenys de les interfícies que condicionaven certes accions dels actors. Un exemple és l'extensió del cas d'ús *Eliminar dApp* estenent de *Editar dApp*, ja que als dissenys cal accedir a editar la dApp per a poder esborrar-la.

### 9.2.1 Casos d'ús de CKBull Developer Console

A continuació, es mostrerà un diagrama de l'àmbit als que pertanyen els casos d'ús i dins de l'escenari *CKBull Developer Console*.

## Autorització i autenticació

A la Figura 11 es mostren tots els casos d'ús relacionats amb l'autorització i autenticació del desenvolupador amb CKBull Developer Console.

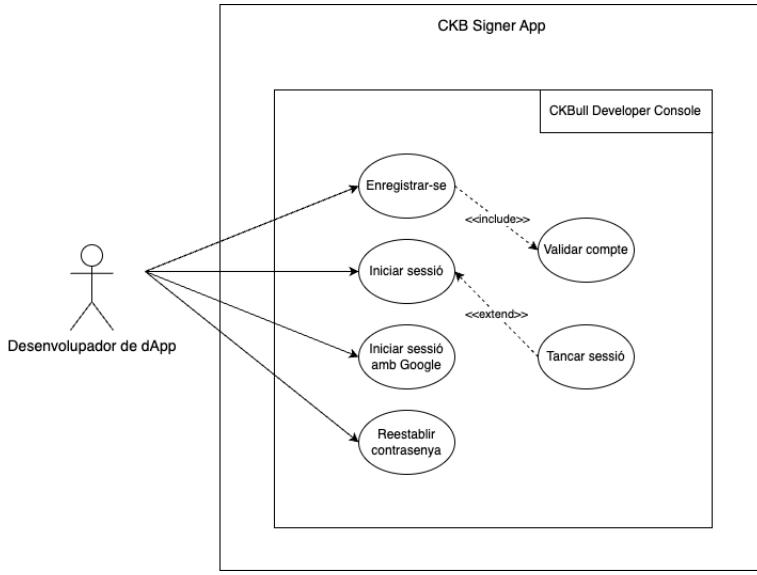


Figura 11: Casos d'ús *CKBull Developer Console*. Àmbit autorització i autenticació.  
Font: elaboració pròpria.

<b>Identificador</b>	Enregar-se
<b>Actors principals</b>	Desenvolupador de dApp
<b>Àrea</b>	CKBull Developer Console
<b>Disparador</b>	El desenvolupador de dApp vol utilitzar per primer cop la consola.
<b>Precondicions</b>	El desenvolupador de dApp no s'ha enregistrat en el sistema prèviament.
<b>Inclou</b>	Validar compte
<b>Descripció</b>	<ol style="list-style-type: none"> <li>1. El desenvolupador, un cop dins de la pàgina, navega fins a la pàgina d'enregistrament.</li> <li>2. El desenvolupador introduceix les dades necessàries per a enregar-se (incloent-hi una adreça de correu electrònic).</li> <li>3. Un cop proporcionades les dades, el desenvolupador rebrà un correu per a validar el compte i serà redirigit a la pàgina de validació.</li> <li>4. El desenvolupador proporciona la informació rebuda per a validar el seu compte. Si és correcte, el desenvolupador queda enregistrat, altrament no.</li> </ol>

Taula 20: Descripció de cas d'ús *Enregar-se*.

<b>Identificador</b>	Iniciar sessió
<b>Actors principals</b>	Desenvolupador de dApp
<b>Àrea</b>	CKBull Developer Console
<b>Disparador</b>	El desenvolupador de dApp vol identificar-se dins de la plataforma.
<b>Precondicions</b>	El desenvolupador de dApp s'ha enregistrat en el sistema prèviament.
<b>Descripció</b>	<ol style="list-style-type: none"> <li>1. El desenvolupador, un cop dins de la pàgina, navega fins a la pàgina d'inici de sessió.</li> <li>2. El desenvolupador proporciona les dades necessàries per a identificar-se (prèviament proporcionades a l'enregistrament). Si les dades coincideixen, el desenvolupador inicia sessió satisfactoriament, altrament no.</li> </ol>

Taula 21: Descripció de cas d'ús *Iniciar sessió*.

<b>Identificador</b>	Iniciar sessió amb Google
<b>Actors principals</b>	Desenvolupador de dApp
<b>Àrea</b>	CKBull Developer Console
<b>Disparador</b>	El desenvolupador de dApp vol identificar-se dins de la plataforma mitjançant un compte de Google.
<b>Precondicions</b>	-
<b>Descripció</b>	<ol style="list-style-type: none"> <li>1. El desenvolupador, un cop dins de la pàgina, clica a l'enllaç d'inici de sessió amb Google.</li> <li>2. El desenvolupador és redirigit a una pàgina segura on pot identificar-se amb el seu compte de Google. En cas d'una identificació satisfactoria l'usuari torna a ser redirigit a CKBull Developer Console com a usuari identificat, altrament no.</li> </ol>

Taula 22: Descripció de cas d'ús *Iniciar sessió*.

<b>Identificador</b>	Restablir contrasenya
<b>Actors principals</b>	Desenvolupador de dApp
<b>Àrea</b>	CKBull Developer Console
<b>Disparador</b>	El desenvolupador de dApp vol restablir la contrasenya del seu compte dins la plataforma.
<b>Precondicions</b>	El desenvolupador ha de estar enregistrat i validat.
<b>Descripció</b>	<ol style="list-style-type: none"> <li>1. El desenvolupador sense identificar-se, clica a l'opció de restablir contrasenya.</li> <li>2. El desenvolupador proporciona una adreça de correu electrònic per a recuperar la contrasenya. Si la direcció és enregistrada dins del sistema i és vàlida, el sistema envia un correu a la direcció amb les dades necessàries per a restablir la contrasenya. Altrament el desenvolupador rep un missatge d'error.</li> <li>3. El desenvolupador amb les dades enviades per correu valida la informació rebuda a la pàgina i proporciona una nova contrasenya. Si les dades són correctes, es restableix la seva contrasenya a la nova introduïda, altrament no es restableix i es notifica al desenvolupador de l'error.</li> </ol>

Taula 23: Descripció de cas d'ús *Reestablir contrasenya*.

<b>Identificador</b>	Validar compte
<b>Actors principals</b>	Desenvolupador de dApp
<b>Àrea</b>	CKBull Developer Console
<b>Disparador</b>	El desenvolupador ha omplert el formulari d'enregistrament al sistema.
<b>Precondicions</b>	El desenvolupador ha proporcionat les dades necessàries per a enregistrar-se al sistema.
<b>Descripció</b>	<ol style="list-style-type: none"> <li>1. El desenvolupador valida la informació rebuda per correu electrònic dins la pàgina de validació.</li> <li>2. Si la informació proporcionada pel desenvolupador és correcta, el desenvolupador és validat dins del sistema i és redirigit a la pàgina principal. Altrament, no i es notifica al desenvolupador de l'error.</li> </ol>

Taula 24: Descripció de cas d'ús *Validat compte*.

<b>Identificador</b>	Tancar sessió
<b>Actors principals</b>	Desenvolupador de dApp
<b>Àrea</b>	CKBull Developer Console
<b>Disparador</b>	El desenvolupador vol sortir del sistema.
<b>Precondicions</b>	El desenvolupador ha iniciat sessió prèviament dins del sistema.
<b>Descripció</b>	<p>1. El desenvolupador clica l'opció de tancar la sessió i deixa de estar autenticat dins del sistema.</p>

Taula 25: Descripció de cas d'ús *Tancar sessió*.

### Gestió de dApps

A la següent figura es mostren tots els casos d'ús relacionats amb la gestió de les dApps. En aquest àmbit ens trobem amb dos actors: desenvolupador de dApp (primàri) i la dApp (secondari).

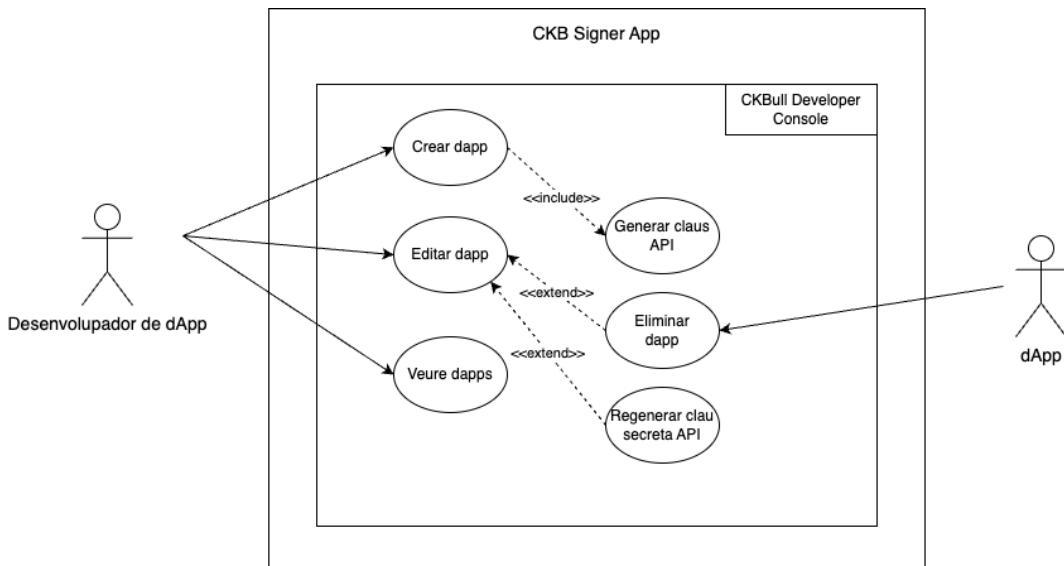


Figura 12: Casos d'ús *CKBull Developer Console*. Àmbit de gestió de dApps.  
Font: elaboració pròpia.

<b>Identificador</b>	Crear dApp
<b>Actors principals</b>	Desenvolupador de dApp
<b>Area</b>	CKBull Developer Console
<b>Disparador</b>	El desenvolupador vol enregistrar una dApp al sistema.
<b>Precondicions</b>	El desenvolupador ha iniciat sessió prèviament dins del sistema.
<b>Inclou</b>	Generar claus API
<b>Descripció</b>	<ol style="list-style-type: none"> <li>1. El desenvolupador clica l'opció d'enregistrar una nova dApp al sistema.</li> <li>2. El desenvolupador es redirigit a un formulari on ha d'introduïr la informació necessària per a enregistrar la dApp. Si la informació proporcionada es vàlida i no existeix cap dApp amb el mateix nom creada per el mateix usuari , la dApp es creada satisfactòriament, altrament el desenvolupador es notificat amb el problema.</li> <li>3. Un cop enregistrada la dApp, es generen les claus API associades a aquesta.</li> <li>4. El desenvolupador guarda les claus API.</li> </ol>

Taula 26: Descripció de cas d'ús *Crear dApp*.

<b>Identificador</b>	Generar claus API
<b>Actors principals</b>	Desenvolupador de dApp
<b>Area</b>	CKBull Developer Console
<b>Disparador</b>	El desenvolupador ha enregistrat una dApp al sistema.
<b>Precondicions</b>	El desenvolupador ha iniciat sessió prèviament dins del sistema i ha creat una dApp satisfactòriament.
<b>Descripció</b>	<ol style="list-style-type: none"> <li>1. Un cop enregistrada la dApp, es generen les claus API associades a aquesta.</li> <li>2. El desenvolupador guarda les claus API.</li> </ol>

Taula 27: Descripció de cas d'ús *Generar claus API*.

<b>Identificador</b>	Editar dApp
<b>Actors principals</b>	Desenvolupador de dApp
<b>Area</b>	CKBull Developer Console
<b>Disparador</b>	El desenvolupador vol editar les dades d'una de les seves dApps al sistema.
<b>Precondicions</b>	El desenvolupador ha iniciat sessió prèviament dins del sistema i ha creat, com a mínim, una dApp satisfactòriament.
<b>Extensió 1</b>	Eliminar dApp [29]
<b>Extensió 2</b>	Regenerar clau secreta API [31]
<b>Descripció</b>	<ol style="list-style-type: none"> <li>1. El desenvolupador clica a l'opció d'editar una dApp ja creada anteriorment.</li> <li>2. El desenvolupador modifica els camps que vol editar.</li> <li>3. El desenvolupador clica l'opció de desar els canvis. Si les dades proporcionades són correctes, el sistema guarda les noves dades. Altrament el desenvolupador es notificat amb l'error corresponent.</li> </ol>

Taula 28: Descripció de cas d'ús *Editar dApp*.

<b>Identificador</b>	Eliminar dApp
<b>Actors principals</b>	Desenvolupador de dApp
<b>Actors secundaris</b>	dApp
<b>Area</b>	CKBull Developer Console
<b>Disparador</b>	El desenvolupador vol eliminar una de les seves dApps al sistema.
<b>Precondicions</b>	El desenvolupador ha iniciat sessió prèviament dins del sistema i ha creat, com a mínim, una dApp satisfactòriament.
<b>Descripció</b>	<ol style="list-style-type: none"> <li>1. El desenvolupador clica a l'opció d'editar una dApp ja creada anteriorment.</li> <li>2. El desenvolupador selecciona l'opció d'eliminar la dApp.</li> <li>3. El desenvolupador es notificat de les conseqüències d'eliminar la dApp del sistema i es demana que confirmi l'acció.</li> <li>4. En cas de que el desenvolupador confirmi, la dApp esborrada del sistema i l'actor secundari dApp deixa d'existir. Altrament es cancela l'operació d'eliminació.</li> </ol>

Taula 29: Descripció de cas d'ús *Eliminar dApp*.

<b>Identificador</b>	Regenerar clau secreta API
<b>Actors principals</b>	Desenvolupador de dApp
<b>Area</b>	CKBull Developer Console
<b>Disparador</b>	El desenvolupador vol regenerar la clau secreta PI d'una de les seves dApps.
<b>Precondicions</b>	El desenvolupador ha iniciat sessió prèviament dins del sistema i ha creat, com a mínim, una dApp satisfactòriament.
<b>Descripció</b>	<ol style="list-style-type: none"> <li>1. El desenvolupador clica a l'opció d'editar una dApp ja creada anteriorment.</li> <li>2. El desenvolupador selecciona l'opció de regenerar una nova clau secreta API.</li> <li>3. El desenvolupador es guarda la nova clau per a poder utilitzar-la a la seva dApp.</li> </ol>

Taula 30: Descripció de cas d'ús *Regenerar clau secreta API*.

<b>Identificador</b>	Veure dApps
<b>Actors principals</b>	Desenvolupador de dApp
<b>Area</b>	CKBull Developer Console
<b>Disparador</b>	El desenvolupador vol visualitzar les dApps que ha enregistrat al sistema.
<b>Precondicions</b>	El desenvolupador ha iniciat sessió prèviament dins del sistema.
<b>Descripció</b>	<ol style="list-style-type: none"> <li>1. El desenvolupador, un cop ja identificat, es redirigit a la pantalla de visualització de dApps, on pot veure en forma de llista les dApps que ha enregistrat.</li> </ol>

Taula 31: Descripció de cas d'ús *Veure dApps*.

## Generació de peticions

A continuació es mostren tots els casos d'ús relacionats amb la generació de les peticions per part d'una dApp a CKBull Developer Console.

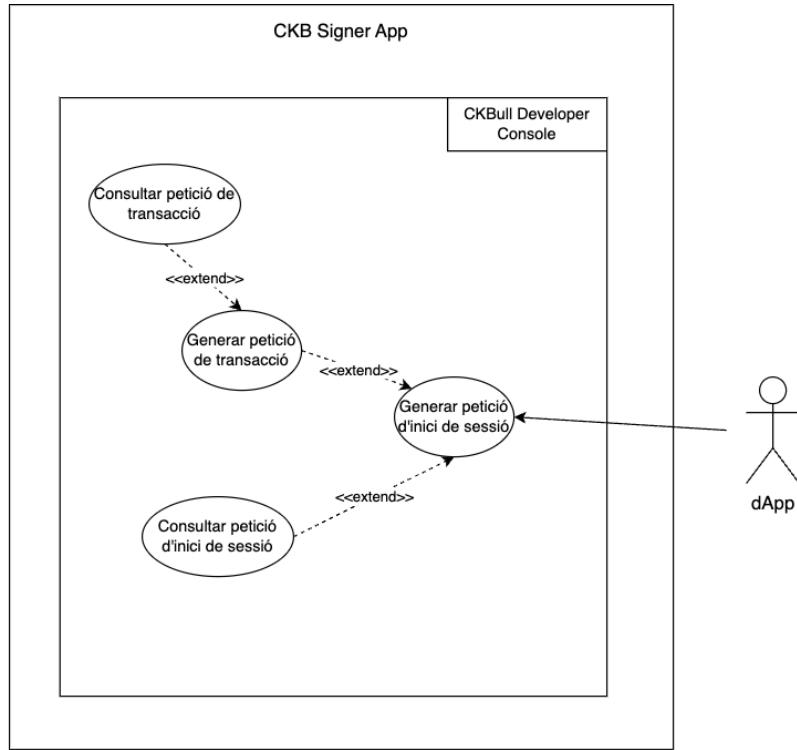


Figura 13: Casos d'ús *CKBull Developer Console*. Àmbit de generació de peticions.  
Font: elaboració pròpia.

<b>Identificador</b>	Generar petició d'inici de sessió
<b>Actors secondaris</b>	dApp
<b>Àrea</b>	CKBull Developer Console
<b>Disparador</b>	La dApp vol generar una petició d'inici de sessió per a un dels seus usuaris.
<b>Precondicions</b>	La dApp ha de comptar amb les claus API generades un cop el desenvolupador enregistra la dApp al sistema.
<b>Extensió no.1</b>	Consultar petició d'inici de sessió [33]
<b>Extensió no.2</b>	Generar petició de transacció [34]
<b>Descripció</b>	<ol style="list-style-type: none"> <li>La dApp demana generar una petició d'inici de sessió amb unes claus API. En cas de les claus API siguin correctes, es genera una nova petició d'inici de sessió. Altrament es notifica a la dApp de l'error produït al sistema.</li> </ol>

Taula 32: Descripció de cas d'ús *Generar petició d'inici de sessió*.

<b>Identificador</b>	Consultar petició d'inici de sessió
<b>Actors secondaris</b>	dApp
<b>Àrea</b>	CKBull Developer Console
<b>Disparador</b>	La dApp vol conèixer l'estat en el que es troba la petició d'inici de sessió.
<b>Precondicions</b>	La dApp ha generat una petició d'inici de sessió per a que un usuari la signi.
<b>Descripció</b>	<p>1. La dApp demana al sistema obtindre l'estat d'una petició d'inici de sessió mitjançant l'identificador d'aquesta. Si existeix, es retorna la informació de la petició, altrament es notifica a la dApp de l'error.</p>

Taula 33: Descripció de cas d'ús *Consultar petició d'inici de sessió*.

<b>Identificador</b>	Generar petició de transacció
<b>Actors secondaris</b>	dApp
<b>Àrea</b>	CKBull Developer Console
<b>Disparador</b>	La dApp vol generar una petició de transacció per a un dels seus usuaris.
<b>Precondicions</b>	La dApp ha de comptar amb les claus API generades un cop el desenvolupador enregistra la dApp al sistema i una petició d'inici de sessió signada.
<b>Extensió no.1</b>	Consultar petició de transacció [34]
<b>Descripció</b>	<p>1. La dApp demana generar una petició de transacció amb unes claus API i l'identificador d'una petició d'inici de sessió signada per un usuari de la dApp.</p> <p>2. En cas de les claus API siguin correctes i l'identificador de la petició estigui signat, es genera una nova petició de transacció associada a l'usuari que ha signat la petició d'inici de sessió. Altrament es notifica a la dApp de l'error produït al sistema.</p>

Taula 34: Descripció de cas d'ús *Generar petició de transacció*.

<b>Identificador</b>	Consultar petició de transacció
<b>Actors secondaris</b>	dApp
<b>Àrea</b>	CKBull Developer Console
<b>Disparador</b>	La dApp vol conèixer l'estat en el que es troba una petició de transacció.
<b>Precondicions</b>	La dApp ha generat una petició de transacció amb informació per a que l'usuari addressat la signi.
<b>Descripció</b>	<p>1. La dApp demana al sistema obtindre l'estat d'una petició de transacció mitjançant l'identificador d'aquesta. Si existeix, es retorna la informació de la petició, altrament es notifica a la dApp de l'error.</p>

Taula 35: Descripció de cas d'ús *Consultar petició de transacció*.

### 9.2.2 Casos d'ús de CKBull Wallet

Per a facilitar la visualització dels casos relacionats amb CKBull Wallet a continuació es mostren els casos d'ús, on es troben dos actors involucrats: **Usuari CKBull Wallet** (primari) i **dApp** (secondari).

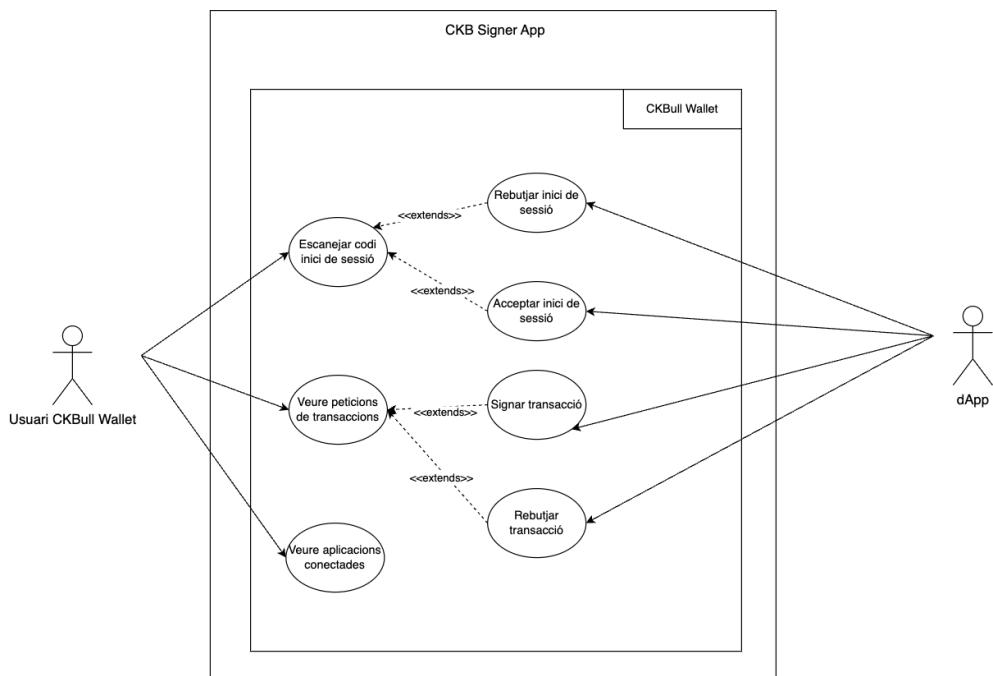


Figura 14: Casos d'ús CKBull Wallet.  
Font: elaboració pròpia.

<b>Identificador</b>	Veure aplicacions connectades
<b>Actors principals</b>	Usuari CKBull Wallet
<b>Àrea</b>	CKBull Wallet
<b>Disparador</b>	L'usuari de CKBull Wallet vol visualitzar les dApps a les que s'ha connectat.
<b>Precondicions</b>	-
<b>Descripció</b>	<ol style="list-style-type: none"> <li>1. L'usuari un cop dins de l'aplicació navega a la pantalla on es troben les dApps a les que està conectat.</li> </ol>

Taula 36: Descripció de cas d'ús *Veure aplicacions connectades*

<b>Identificador</b>	Escanejar codi inici de sessió
<b>Actors principals</b>	Usuari CKBull Wallet
<b>Àrea</b>	CKBull Wallet
<b>Disparador</b>	L'usuari de CKBull Wallet vol iniciar sessió en una dApp enregistrada a CKBull Developer Console.
<b>Precondicions</b>	La dApp ha de mostrar un codi QR a l'usuari amb una petició d'inici de sessió no signada o rebutjada.
<b>Extensió 1</b>	Rebutjar inici de sessió [38]
<b>Extensió 2</b>	Signar inici de sessió [39]
<b>Descripció</b>	<ol style="list-style-type: none"> <li>1. L'usuari compta amb un codi QR generat per la dApp.</li> <li>2. L'usuari escaneja amb l'aplicació el codi QR. En cas que el codi QR sigui vàlid es mostra la informació de la petició d'inici de sessió, altrament es notifica a l'usuari de l'error.</li> </ol>

Taula 37: Descripció de cas d'ús *Escanejar codi inici de sessió*.

<b>Identificador</b>	Rebutjar inici de sessió
<b>Actors principals</b>	Usuari CKBull Wallet
<b>Actors secundaris</b>	dApp
<b>Àrea</b>	CKBull Wallet
<b>Disparador</b>	L'usuari de CKBull Wallet vol rebutjar una petició d'inici de sessió prèviament escanejada.
<b>Precondicions</b>	L'usuari ha escanejat una petició d'inici de sessió vàlida.
<b>Descripció</b>	<ol style="list-style-type: none"> <li>1. L'usuari, dins de l'opció de signar o rebutjar una petició d'inici de sessió, rebutja l'operació dins l'aplicació.</li> <li>2. La dApp rep la petició rebutjada i cancel·la les operacions pertinents.</li> </ol>

Taula 38: Descripció de cas d'ús *Rebutjar inici de sessió*.

<b>Identificador</b>	Acceptar inici de sessió
<b>Actors principals</b>	Usuari CKBull Wallet
<b>Actors secundaris</b>	dApp
<b>Àrea</b>	CKBull Wallet
<b>Disparador</b>	L'usuari de CKBull Wallet vol acceptar una petició d'inici de sessió prèviament escanejada.
<b>Precondicions</b>	L'usuari ha escanejat una petició d'inici de sessió vàlida.
<b>Descripció</b>	<ol style="list-style-type: none"> <li>1. L'usuari, dins de l'opció d'acceptar o rebutjar una petició d'inici de sessió, accepta l'operació dins l'aplicació.</li> <li>2. La dApp rep la petició signada.</li> <li>3. La dApp verifica a l'usuari, que pot començar a realitzar accions dins de la dApp.</li> </ol>

Taula 39: Descripció de cas d'ús *Acceptar inici de sessió*.

<b>Identificador</b>	Veure peticions de transaccions
<b>Actors principals</b>	Usuari CKBull Wallet
<b>Àrea</b>	CKBull Wallet
<b>Disparador</b>	L'usuari de CKBull Wallet vol signar una petició de transacció pendent de signar generada per la dApp.
<b>Precondicions</b>	-
<b>Extensió 1</b>	Signar transacció [41]
<b>Extensió 2</b>	Rebutjar transacció [42]
<b>Descripció</b>	<ol style="list-style-type: none"> <li>1. L'usuari, un cop dins de l'aplicació navegarà fins a la pestanya on es llistaran les peticions de transacció pendents de signar.</li> <li>2. En cas que no hi hagi cap transacció, l'usuari rebrà un missatge assegurant que no existeixen transaccions pendents de signar.</li> </ol>

Taula 40: Descripció de cas d'ús *Veure peticions de transaccions*.

<b>Identificador</b>	Signar transacció
<b>Actors principals</b>	Usuari CKBull Wallet
<b>Actors secundaris</b>	dApp
<b>Àrea</b>	CKBull Wallet
<b>Disparador</b>	L'usuari de CKBull Wallet vol signar una petició de transacció prèviament generada per la dApp.
<b>Precondicions</b>	L'usuari té com a mínim una petició de transacció pendent de signar generada per la dApp.
<b>Descripció</b>	<ol style="list-style-type: none"> <li>1. L'usuari navega a la pantalla de signatura de la petició de transacció.</li> <li>2. L'usuari visualitza la informació de la transacció.</li> <li>3. L'usuari signa la transacció, emetent la signatura a la <i>blockchain</i>.</li> <li>4. La dApp confirma la transacció dins la <i>blockchain</i>.</li> </ol>

Taula 41: Descripció de cas d'ús *Signar transacció*.

<b>Identificador</b>	Rebutjar transacció
<b>Actors principals</b>	Usuari CKBull Wallet
<b>Actors secondaris</b>	dApp
<b>Àrea</b>	CKBull Wallet
<b>Disparador</b>	L'usuari de CKBull Wallet vol rebutjar una petició de transacció prèviament generada per la dApp.
<b>Precondicions</b>	L'usuari té com a mínim una petició de transacció pendent de signar generada per la dApp. Aquesta petició no pot estar signada, rebutjada o expirada.
<b>Descripció</b>	<ol style="list-style-type: none"> <li>1. L'usuari navega a la pantalla de signatura de la petició de transacció.</li> <li>2. L'usuari visualitza la informació de la transacció.</li> <li>3. L'usuari rebutja la transacció.</li> <li>4. La dApp és notificada de l'acció de l'usuari.</li> </ol>

Taula 42: Descripció de cas d'ús *Rebutjar transacció*.

### 9.3 Esquema conceptual de dades

Després d'haver identificat els actors principals del projecte i quines accions han de desenvolupar per a un funcionament correcte del sistema passem a especificar quines entitats trobarem al sistema i com es relacionen entre elles.

Per a poder visualitzar-ho, s'ha realitzat un esquema conceptual de dades on es mostren totes les classes que interaccionen dins del projecte, conjuntament amb els seus atributs i relacions. La Figura 15 mostra el diagrama complet.

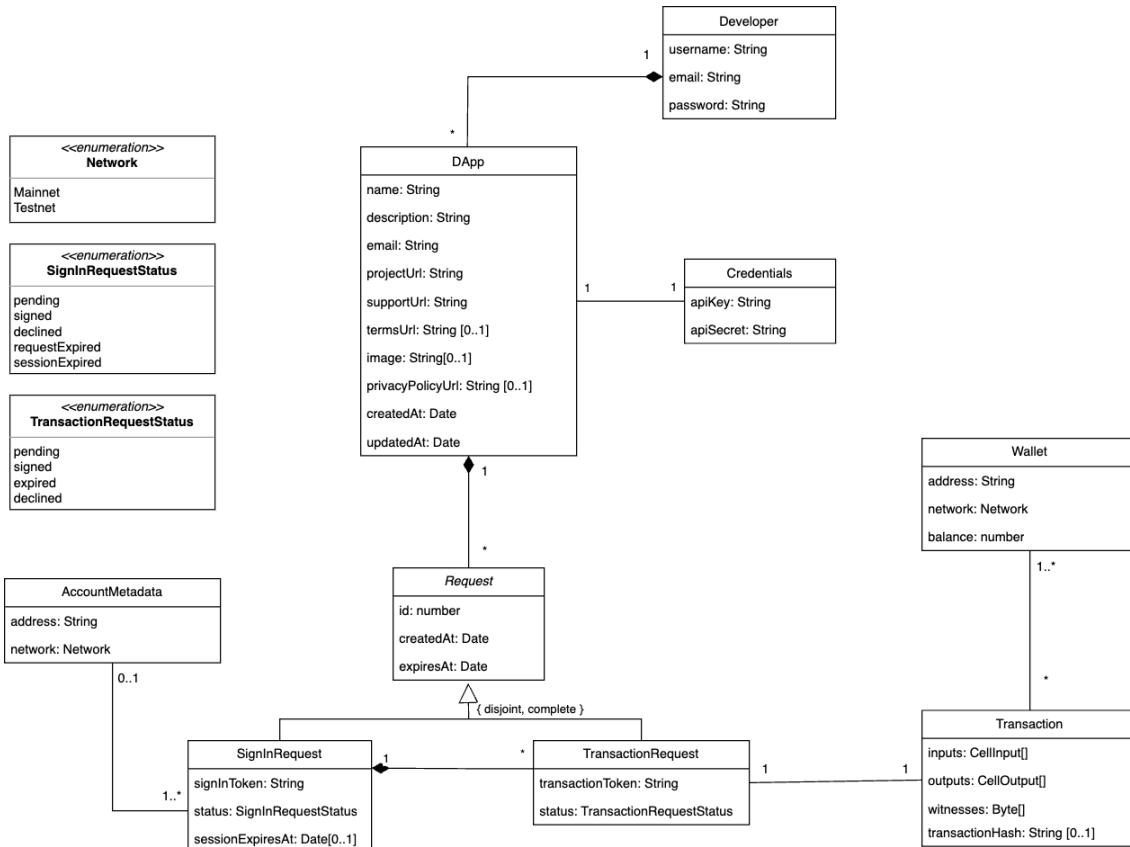


Figura 15: Esquema conceptual de dades del sistema

Font: elaboració pròpia.

A continuació es descriuen les restriccions textuales de l'esquema conceptual de dades:

- **Claus externes:** (Developer, email), (DApp, name), (Credentials, apiKey), (SignInRequest, id), (TransactionRequest, id), (AccountMetadata, address + network), (Wallet, address).
- L'atribut username de la classe Developer no es pot repetir.
- L'atribut signInToken de la classe SignInRequest no es pot repetir.
- L'atribut transactionToken de la classe TransactionRequest no es pot repetir.
- L'atribut updatedAt de la classe DApp no pot ser anterior a l'atribut createdAt.

- L'atribut expiresAt de la classe Request no pot ser anterior a l'atribut createdAt.
- L'atribut sessionExpiresAt de la classe SignInToken, en cas de tindre valor, no pot ser anterior a l'atribut createdAt.
- La classe SignInRequest només pot tindre un objecte de la classe AccountMetadata si el seu status es signed.
- L'atribut sessionExpiresAt només pot tindre valor si el status de la SignInRequest es signed.
- Una TransactionRequest només pot tindre una SignInRequest associada si aquesta té com a status signed.

### 9.3.1 Descripció de les classes

Per aprofundir més a l'esquema conceptual, a continuació s'adjunta una breu descripció de cada classe del sistema amb els seus atributs.

#### Developer

Representa a la persona o entitat que desenvolupa les dApps que s'integraran al sistema. Els seus atributs son els següents:

Atribut	Descripció	Tipus
username	Nom del compte del desenvolupador	String
email	Adreça de correu electrònic	String
password	Contrasenya utilitzada per identificar-se	String

Taula 43: Atributs de la classe Developer.

#### DApp

Representa la informació de la dApp enregistrada pel desenvolupador que utilitzarà la plataforma per a enviar peticions als usuaris. Aquesta és desenvolupada per un Developer.

Atribut	Descripció	Tipus
name	Nom de la dApp	String
description	Descripció general de la dApp	String
email	Adreça de correu electrònic	String
projectUrl	Adreça URL de la dApp	String
supportUrl	Adreça URL de per a suport	String
termsUrl (opcional)	Adreça URL on trobar els termes d'ús de la dApp	String
privacyPolicyUrl (opcional)	Adreça URL on trobar les polítiques de privacitat de la dApp	String
createdAt	Data d'enregistrament de la dApp dins la plataforma	Data
updatedAt	Data d'actualització de la informació de la dApp	Data

Taula 44: Atributs de la classe DApp.

## Credentials

Credencials (o claus) que usrà la dApp dins del sistema per a autenticar-se i poder generar peticions (Requests). Un objecte de la classe Credentials només pot tindre associada una dApp. L'ús de les credencials es pot trobar al Capítol 11.

Atribut	Descripció	Tipus
apiKey	String de 32 caràcters en base64 que identifica la dApp	String
apiSecret	String de 32 caràcters en base64 que s'utilitza per fer signatures	String

Taula 45: Atributs de la classe Credentials.

## SignInRequest

Representa la petició d'inici de sessió que realitzen les dApps per identificar un compte d'un usuari de CKBull Wallet. Una dApp pot generar múltiples SignInRequest. Si una SignInRequest té un status de signed, aleshores ha de tindre un objecte de la classe AccountMetadata associat i podrà tindre TransactionRequest associades.

Atribut	Descripció	Tipus
id	Identificador de la petició	nombre enter
createdAt	Data de creació de la SignInRequest	Data
expiresAt	Data d'expiració de la SignInRequest	Data
signInToken	String de 32 caràcters aleatòris identificador de la petició	String
status	Estat en el qual es troba la petició	SignInRequestStatus
sessionExpiresAt (opcional)	Indica la data d'expiració de la sessió si el status es signed	Data

Taula 46: Atributs de la classe SignInRequest.

## TransactionRequest

Petició de transacció generada per la dApp perquè sigui signada o rebutjada per un compte prèviament identificat amb una SignInRequest. En el moment de la creació, s'associa una Transaction a la TransactionRequest. A més, cal que la SignInRequest associada a la TransactionRequest tingui l'atribut status *signed*.

Atribut	Descripció	Tipus
id	Identificador de la petició	nombre enter
createdAt	Data de creació de la TransactionRequest	Data
expiresAt	Data d'expiració de la TransactionRequest	Data
transactionToken	String de 32 caràcters aleatòris identificador de la petició	String
status	Estat en el qual es troba la petició	TransactionRequestStatus

Taula 47: Atributs de la classe TransactionRequest.

## Transaction

Objecte que conté tots els atributs necessaris perquè es realitzi una transacció. Aquest objecte es crea amb una TransactionRequest i s'associa a un o més comptes (participants de la transacció). A la secció Secció 9.1 es descriuen els camps de la transacció en més detall.

Atribut	Descripció	Tipus
inputs	CellInputs necessaris per fer la transacció	Vector de CellInput
outputs	CellOutputs necessaris per fer la transacció	Vector de CellOutput
witnesses	Signatures codificades en format byte	Vector de bytes
transactionHash (opcional)	hash resultant de la signatura de la transacció	String

Taula 48: Atributs de la classe Transaction.

## Wallet

Representa el compte d'un usuari de la *blockchain* CKB. Aquest pot tindre entre 0 i moltes transaccions. Els seus atributs són els següents:

Atribut	Descripció	Tipus
address	Adreça única dins de la blockchain	String
network	Cadena on es troba la cartera (mainnet o testnet)	Network
balance	Quantitat d'actius digitals que disposa	número

Taula 49: Atributs de la classe Wallet.

## 9.4 Cicles de vida d'una Request

Com s'ha descrit a la Figura 15 existeixen dos tipus de Request: les SignInRequest i les TransactionRequest. Aquestes peticions corresponen a les peticions d'inici de sessió (SignInRequest) i a les peticions de transacció (TransactionRequest). Tant les SignInRequest com les TransactionRequest tenen un atribut anomenat status, que correspon a l'estat en el qual es troba la petició. Per entendre en profunditat perquè necessiten estat, com canvia i quina relació existeix entre els dos tipus de peticions, en aquesta secció es defineixen els cicles de vida dels diferents tipus de peticions.

### 9.4.1 SignInRequest (petició d'inici de sessió)

Les SignInRequest tenen com a objectiu poder identificar el compte d'un usuari de CKBull Wallet que utilitzi la dApp per a fer transaccions. A continuació, a la Figura 16 es mostra el cicle de vida d'una SignInRequest des de la seva creació.

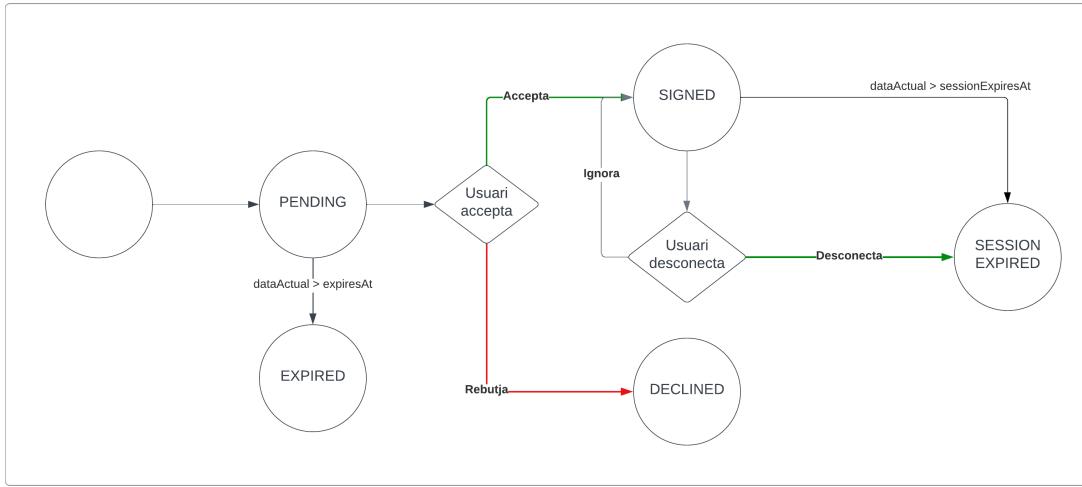


Figura 16: Cicle de vida d'una SignInRequest.

Font: elaboració pròpia.

L'estat inicial d'una SignInReuest quan és creada correspondrà a *pending* (pendent). A partir d'aquest punt poden passar dos casos. Si la data actual és posterior a la data de l'atribut *expiresAt* de la petició, aleshores la petició ja no serà vàlida i canviàrà d'estat a *expired* (expirada). Altrament, si un usuari decideix actuar sobre la petició podrà realitzar dues accions: acceptar-la, proporcionant la informació necessària per a identificar-lo, o rebutjar-la. En cas d'acceptar la petició, l'usuari proporcionarà una tupla que conté l'adreça del compte que utilitzi i la xarxa (network) a la qual pertany (mainnet o testnet), representada a la Figura 15 com a AccountMetadata. D'aquesta manera la dApp podrà identificar l'usuari per a posteriorment generar peticions de transacció. Si l'usuari accepta l'estat de la petició muta a *signed* (signada) i si la rebutja l'estat romandrà com a *declined* (rebutjada).

Quan una petició es acceptada l'atribut *sessionExpiresAt* obté una data que, en cas que la data actual sigui posterior a aquesta, canvi l'estat a *session expired* (sessió expirada), inutilitzant la sessió i evitant que la dApp pugui crear peticions de transacció. No obstant, un cop la petició ha sigut acceptada, l'usuari pot desconectar la seva sessió si aquesta no es troba expirada. En cas que vulgui desconectar la sessió, l'estat passaria de *signed* a *session expired* i anulant l'atribut *sessionExpiresAt* de la petició.

#### 9.4.2 TransactionRequest (petició de transacció)

Una petició de transacció té un cicle de vida més simple respecte a una petició d'inici de sessió. No obstant això, perquè una **TransactionRequest** pugui ser creada ha de **tindre associada una SignInRequest que es trobi en estat signed**. El motiu d'aquesta **precondició** és pel fet que no s'han de generar peticions de transacció si es desconeix quins actors estan involucrats en aquesta. Així doncs, si la SignInRequest es troba en estat de *signed* significa que té associada un objecte de la classe AccountMetadata que identifica a la part interessada de la transacció.

Explicada la precondició, a la figura següent trobem el cicle de vida d'una TransactionRequest.

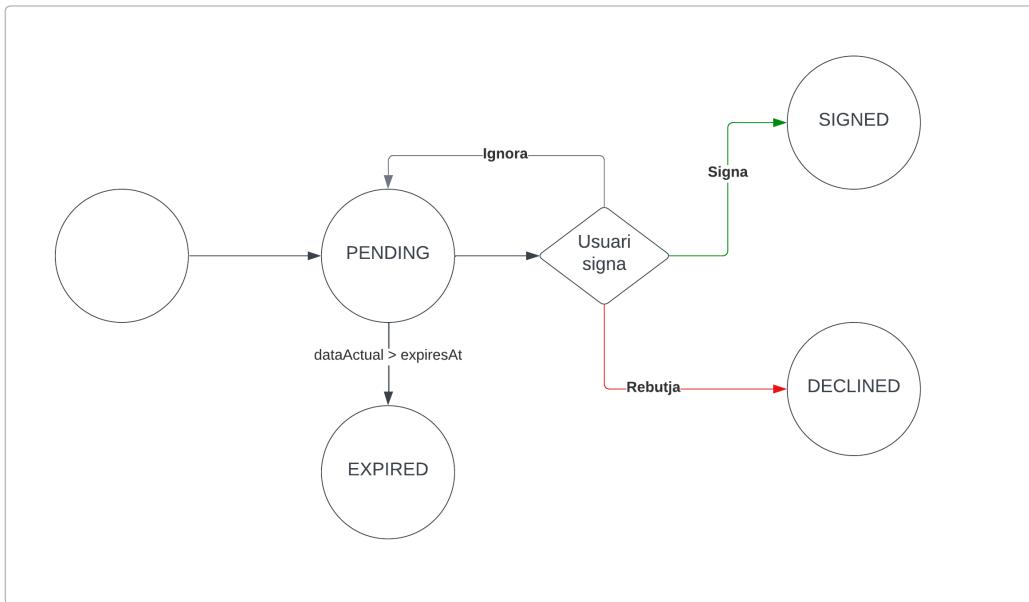


Figura 17: Cicle de vida d'una TransactionRequest.  
Font: elaboració pròpia.

L'estat inicial sempre serà *pending* (pendent). A més, tal com funcionen les SignInRequest, si la data actual és posterior a la data de l'atribut *expiresAt*, l'estat mutarà a *expired* (expirat). La resta de canvis d'estat vindran donats de l'acció que decideixi realitzar l'usuari. Com es pot veure a la Figura 17 si l'usuari signa la transacció aquesta mutarà a l'estat *signed* (signada) i finalitzarà el seu canvi d'estat, ja que la transacció associada a la petició serà signada a CKBull Wallet i emitida cap a la *blockchain*. En cas contrari, si rebutja la petició, aquesta mutarà a *declined* (rebutjada) on també finalitzarà el seu canvi d'estat. Altrament, si l'usuari decideix ignorar la transacció i no dur a terme cap acció, aquesta romandrà en estat *pending* (pendent) fins que l'usuari realitzi alguna acció o la petició caduqui.

## 9.5 Especificació dels processos crítics

Per a finalitzar la fase d'especificació del projecte, dins d'aquesta secció es definiran els processos crítics del projecte, sent aquests els processos més importants dins del sistema. Sense aquests processos, el sistema seria incapaç de complir la funcionalitat d'enviament de peticions entre una dApp i un compte de CKBull Wallet. Aquests processos actuen com a nucli del sistema i s'han considerat rellevants per a poder evitar els riscos definits al Capítol 7.

Els processos crítics detectats són els següents:

- **Procés d'acceptació d'inici de sessió:** Procés que engloba la creació d'una petició d'inici de sessió per part d'una dApp i l'acceptació o rebutja per part d'un usuari, proporcionant en cas d'acceptació informació que l'identificaria (com acompte).
- **Process de signatura d'una petició de transacció:** Procés que comporta el procés anterior, la creació de la petició de transacció per part d'una dApp, l'acció de signatura o rebutja per part de l'usuari i l'emissió de la transacció a la *blockchain*.

A continuació, es descriuran els processos amb més detall.

### 9.5.1 Procés d'acceptació d'una petició d'inici de sessió

Els actors que s'involucren dins d'aquest procés són: Usuari CKBull Wallet i la dApp. Per acotar la dimensió del procés, es té com a precondició que l'actor dApp ja ha sigut creat pel desenvolupador i que disposa d'unes credencials vàlides que l'identifiquen. Així doncs, el **disparador** de l'acció correspon a **un usuari de CKBull Wallet que vol iniciar sessió dins d'una dApp amb un compte**.

Per a fer-ho demana iniciar sessió, fent que la dApp cridi a CKBull Developer Console perquè generi una nova SignInRequest per a un usuari. **Aquesta crida s'haurà de realitzar amb les credencials de la dApp per verificar l'autenticació de la dApp**. Un cop creada, la plataforma retorna una SignInRequest sense signar que és mostrada a l'usuari en forma de codi QR. A continuació, l'usuari mitjançant l'aplicació CKBull Wallet, escaneja el codi QR per poder obtenir la SignInRequest i poder acceptar-la o rebutjar-la. En cas d'acceptarla, s'envia a CKBull Developer Console on s'actualitza l'estat de la petició a *signed* i es crea un objecte de la classe AccountMetadata amb l'adreça del compte amb el que ha signat l'usuari i la seva xarxa (*network*). Altrament, si l'usuari rebutja la petició, l'API actualitza el *status* a *declined*.

Durant tot aquest procés, la dApp ha d'estar actualitzada respecte a l'estat de la SignInRequest. Per a fer-ho, enviarà peticions al sistema demandant l'estat en el qual es troba la petició fins que aquesta deixa de trobar-se en estat *pending*. Per comprendre com es faran les peticions, el Capítol 10 defineix l'arquitectura que tindrà la solució. La figura següent permet visualitzar l'explicació del procés d'acceptació d'inici de sessió:

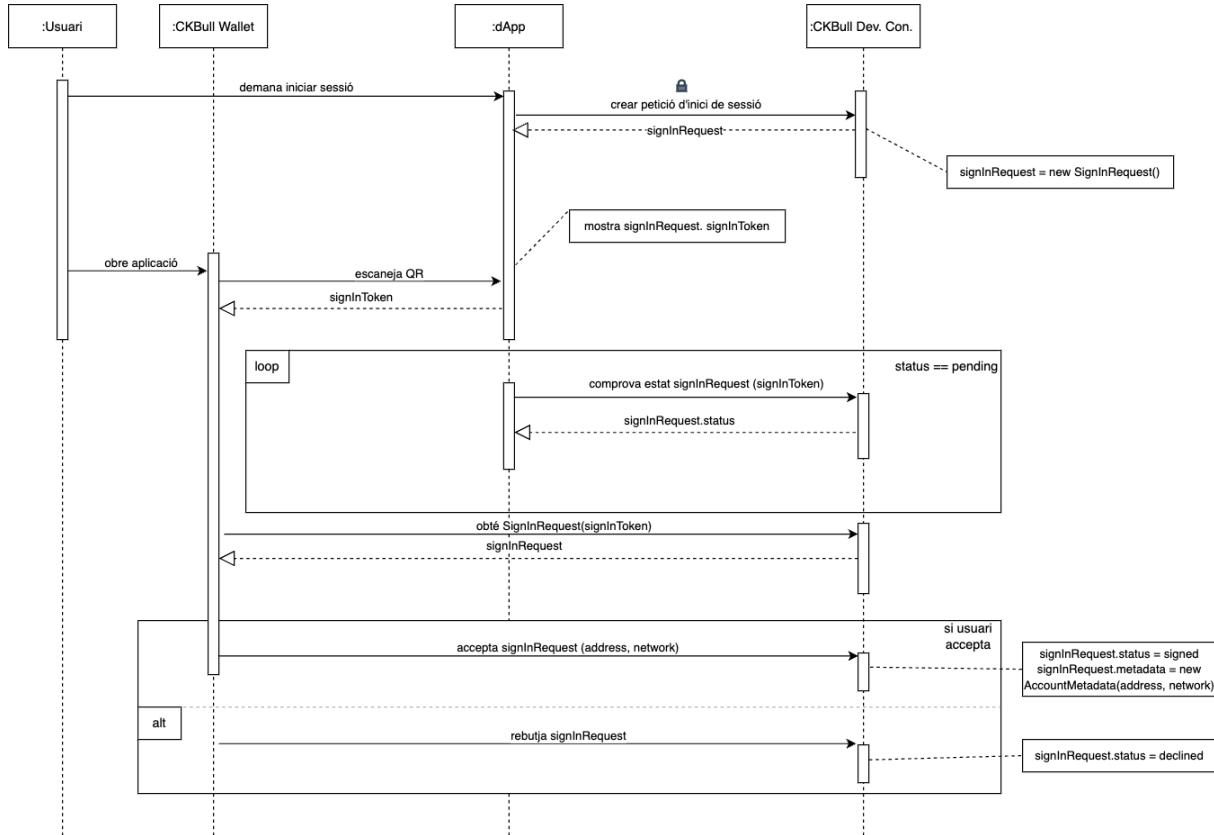


Figura 18: Procés d'acceptació d'una petició d'inici de sessió.

Font: elaboració pròpria.

Com es pot veure a la Figura 18 la crida de creació d'una petició d'inici de sessió entre la dApp i CKBull Developer Console conté un cadenat, que representa l'autenticació que ha de proporcionar la dApp a la CKBull Developer Console amb les credencials generades. Més endavant, al Capítol 11, s'especificarà quin tipus d'autenticació s'aplicarà.

### 9.5.2 Procés de signatura d'una petició de transacció

El segon procés crític del sistema consisteix en la signatura d'una petició de transacció per part d'un usuari de CKBull Wallet. Com a precondició d'aquest procés s'han de complir les següents precondicions:

1. L'usuari de CKBull Wallet ha iniciat sessió amb la dApp amb el compte que vol realitzar la transacció.
2. La petició d'inici de sessió signada per l'usuari ha de trobar-se en estat (*status*) *signed*.

En cas que aquestes dues precondicions siguin certes, una dApp podrà generar una petició de transacció per a l'usuari que hagi iniciat sessió i vulgui fer la transacció. La Figura 19 resumeix el procés.

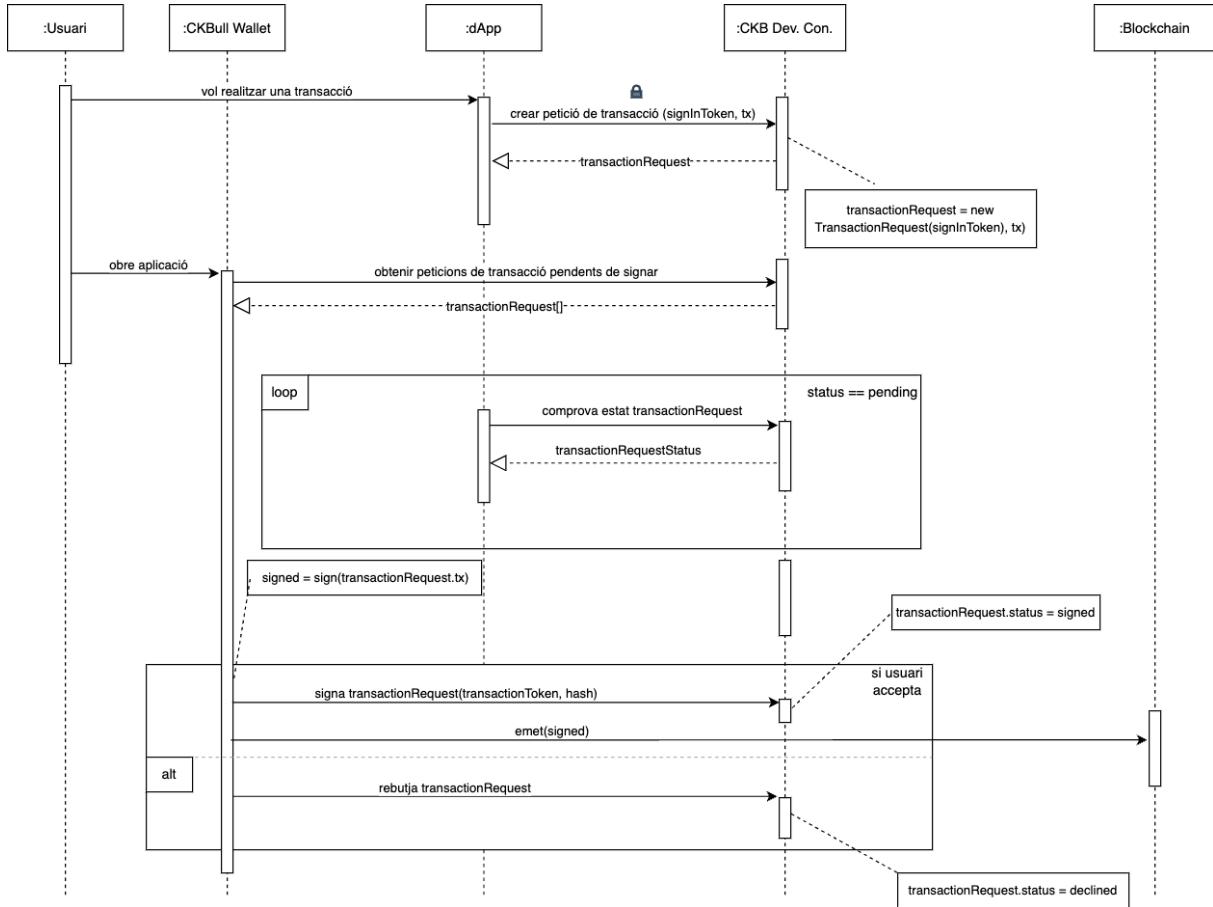


Figura 19: Procés de signatura d'una petició de transacció.  
Font: elaboració pròpia.

Com es pot observar a la figura anterior, la crida de creació de peticions (en aquest cas la creació d'una petició de transacció) també disposa d'un mecanisme d'autenticació per part de la dApp. Amb aquest filtre d'autenticació aconseguim garantir que qui vol crear les peticions de transacció és realment la dApp amb les credencials proporcionades. A més, podem veure que es repeteix el mateix patró en els dos processos crítics. L'usuari desencadena l'acció, fent que la dApp generi les peticions a CKBull Developer Console. A més, els dos processos disposen d'un recurs de *polling*<sup>1</sup> per a consultar periòdicament l'estat de la petició.

No obstant, la part de signatura i rebutjament d'una petició varia en aquest segon procés pel fet que intervé la *blockchain*. En cas que l'usuari realitzi la signatura de la petició, CKBull Wallet es el responsable d'emetre la transacció signada cap a la *blockchain*. **Aquesta acció no es pot desfer**, i comporta l'actualització de l'estat (status) de la petició de transacció a *signed*. En cas contrari, únicament s'actualitzarà l'estat a *declined* i la *blockchain* no es veurà afectada.

<sup>1</sup>*Polling*: operació de consulta constant sobre un recurs.

# Capítol 10

## Disseny

Un cop especificat el domini del problema, els seus actors, escenaris, entitats i casos d'ús es pot iniciar la fase de disseny de la solució. En aquest capítol es descriurà quina arquitectura tindrà el sistema, quins són els seus components i com es relacionen.

### 10.1 Arquitectura del sistema

A causa del temps disponible per a fer el projecte i els recursos humans, s'ha decidit realitzar una solució que funcioni per sobre de la *blockchain*, de manera isolada i dividida per capes. Com es pot veure a la Figura 20 l'arquitectura general del sistema consta de 4 capes:

- **Presentació:** Capa on es troben totes les interfícies amb les quals interactuaran els actors del sistema.
- **Domini:** Capa on es troben els components que gestionen la lògica del sistema i comuniquen amb les dades.
- **Persistència:** Capa on s'emmagatzemarà totes les dades necessàries del sistema.
- **Blockchain:** Capa horitzontal sobre la qual es desenvoluparà el projecte. L'únic recurs que existeix dins d'aquesta capa és la pròpia *blockchain* CKB.

La motivació a implementar un sistema de capes a l'arquitectura de sistema és poder agrupar els components de cada capa segons la funció que duguin a terme. Segmentar els comportaments facilita una major escalabilitat de la solució i un major control sobre aquests.

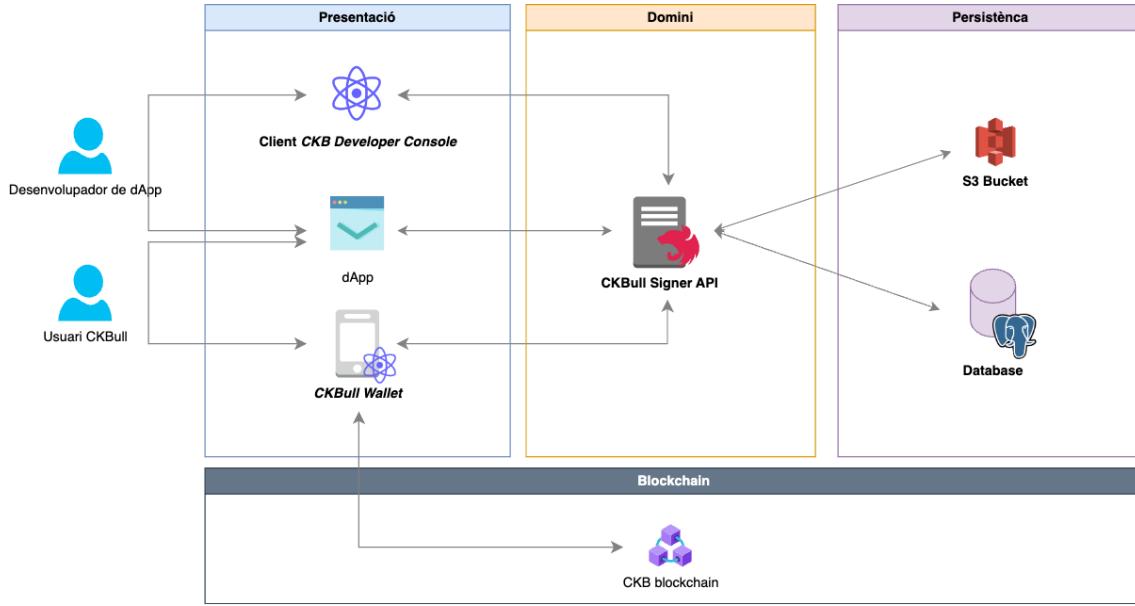


Figura 20: Arquitectura general del sistema.

Font: elaboració pròpria.

Dins d'aquestes capes **no tots els components s'hauran de dissenyar ni implementar** perquè són components ja existents que interactuen amb el sistema. Així doncs, la llista de components que s'hauran de dissenyar i implementar són:

- **Client CKBull Developer Console:** Client que permetrà als desenvolupadors de dApps enregistrar la seva dApp i obtenir les credencials per generar peticions.
- **CKBull Wallet:** Cartera de criptomonedes sobre la que s'afegirà la funcionalitat acceptar o rebutjar peticions.
- **CKBull Signer API:** API que gestionarà tant les dApps enregistrades al sistema com les peticions d'inici de sessió i de transaccions que generin.
- **Database:** Base de dades que emmagatzemarà la informació necessària per a l'execució del sistema com, per exemple, les entitats.

Cal destacar que **el concepte CKBull Developer Console**, esmentat al capítol anterior, **se segmentarà en els dos components Client CKBull Developer Console i CKBull Signer API**. Aquesta segmentació és donada per afavorir el desacoplament entre funcionalitats diferents i la separació de responsabilitats d'interfície (fetes per Client CKBull Developer Console) i de domini (CKBull Signer API). Per aprofundir una mica més en l'arquitectura del component la Figura 21 mostra l'arquitectura interna de cada component a dissenyar. A les següents seccions es definirà més en detall com està composta l'arquitectura dels components.

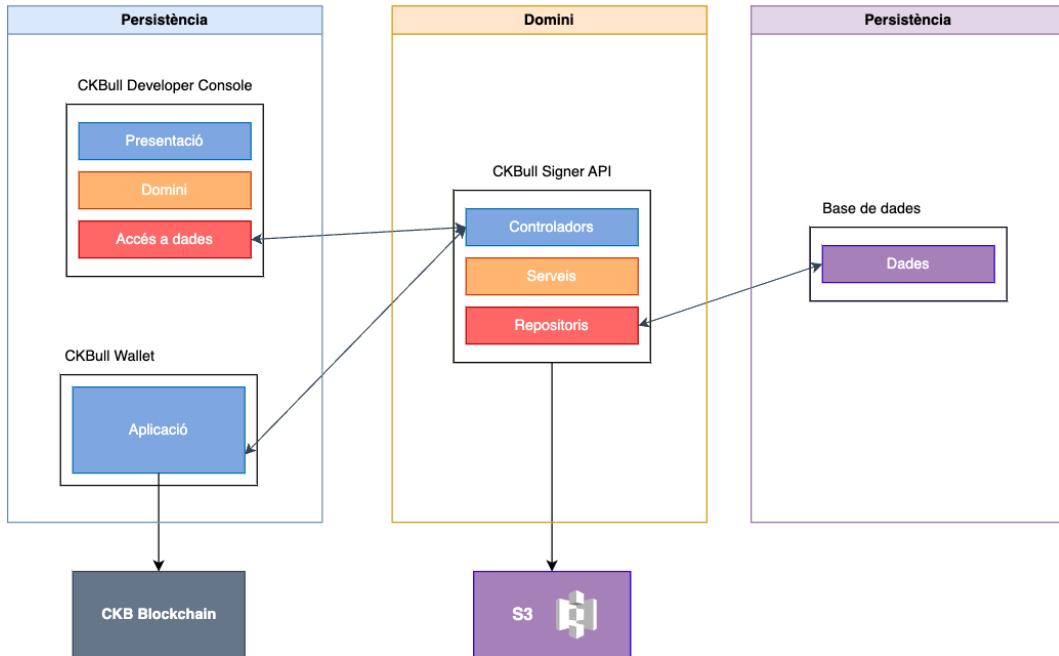


Figura 21: Arquitectura dels components a dissenyar.

Font: elaboració pròpia.

Els serveis que es troben fora de les capes (CKB Blockchain i el contenidor S3) queden fora degut al fet que **no s'han d'implementar** perquè són sistemes ja existents dels que únicament és **consumeixen serveis**. A les següents seccions s'aprofundirà en el disseny de cada component, visualitzant la seva arquitectura i quins patrons de disseny s'han aplicat.

## 10.2 Disseny de les interfícies

Com s'ha descrit a la secció anterior, a la capa de presentació existeixen dos components que permeten connectar els actors amb el sistema. A continuació, es descriu el disseny de cada component.

### 10.2.1 Disseny de CKBull Developer Console

Existeixen certs casos d'ús a l'escenari CKBull Developer Console que han d'estar presents dins de la interfície. Els casos d'ús són els relacionats amb l'actor Desenvolupador de dApp pel fet que és l'únic actor que podrà interactuar amb la interfície. Per satisfer tots els casos d'ús, s'han utilitzat diverses eines per aconseguir un disseny complet de la interfície.

#### Arquitectura de CKBull Developer Console

L'arquitectura del client CKBull Developer Console és basada en l'arquitectura en capes. En aquest component existeixen tres capes:

- **Presentació:** Engloba tots els elements que es visualitzaran a la interfície. Només gestiona la lògica a nivell de component o element d'interfície.
- **Domini:** Gestiona l'estat i la lògica del client, fent també de capa comunicadora entre la capa de presentació i la capa d'accés a dades.

- **Accés a dades:** Capa encarregada de la petició de dades als altres serveis del sistema i gestió de dades locals.

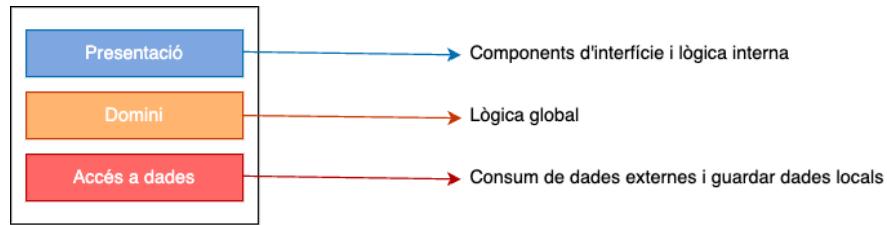


Figura 22: Arquitectura de CKBull Developer Console.

Font: elaboració pròpia.

### Diagrama de navegabilitat

El diagrama de navegabilitat és una estructura que permet representar de manera visual com pot interactuar l'usuari amb el sistema, mostrant les navegacions i els processos que pot realitzar.

A la figura següent es mostra el mapa de navegabilitat de la interfície amb les pàgines i components que es trobarà l'usuari. Les pàgines són referenciades amb el color verd i els modals són referenciats amb el color lila. Cada connexió té com a etiqueta el desencadenant que provoca la navegació.

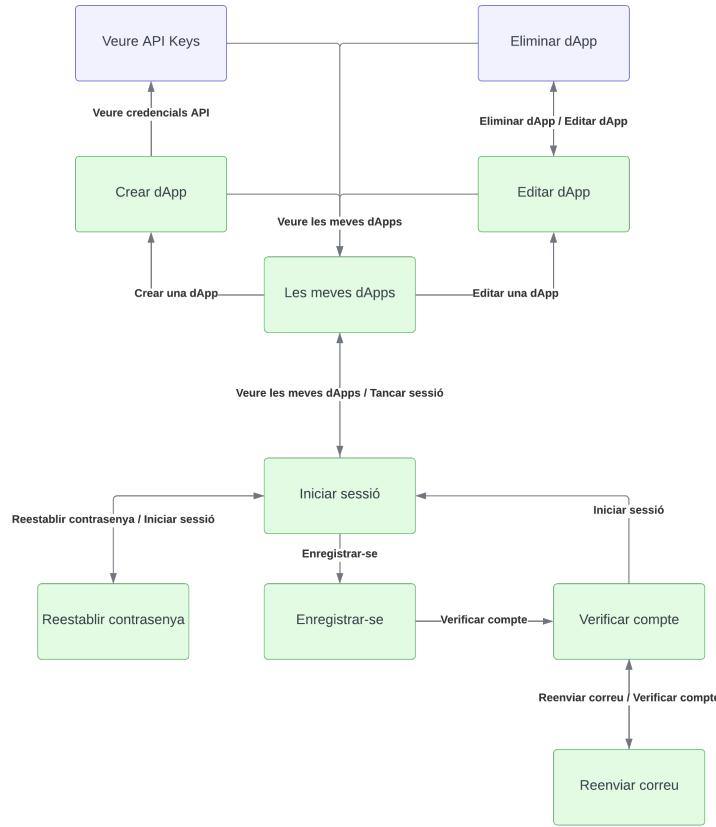


Figura 23: Mapa de navegabilitat de CKBull Developer Console.

Font: elaboració pròpia.

## Dissenys externs

L'empresa de dissenyadors Urano va ser contractada per a generar uns dissenys de la interfície de CKBull Developer Console acord amb la imatge de la *blockchain* CKB.

A continuació, es llisten les pàgines dissenyades per Urano. Els dissenys no inclouen totes les pantalles i modals, però són suficients per determinar l'aspecte visual de la interfície i poder generar nous components en cas de necessitat.

## Pàgina d'inici de sessió

Correspon a la pàgina on el desenvolupador de dApps podrà identificar-se. Consta d'un formulari d'inici de sessió amb dos camps, un per a la adreça de correu electrònic amb el qual el desenvolupador s'hagi enregistrat, i la contrasenya. A partir dels *links* proporcionats al formulari pot navegar cap a la pàgina de restablir contrasenya o a la pàgina d'enregistrament. A més, també pot iniciar la sessió amb el seu compte de Google.

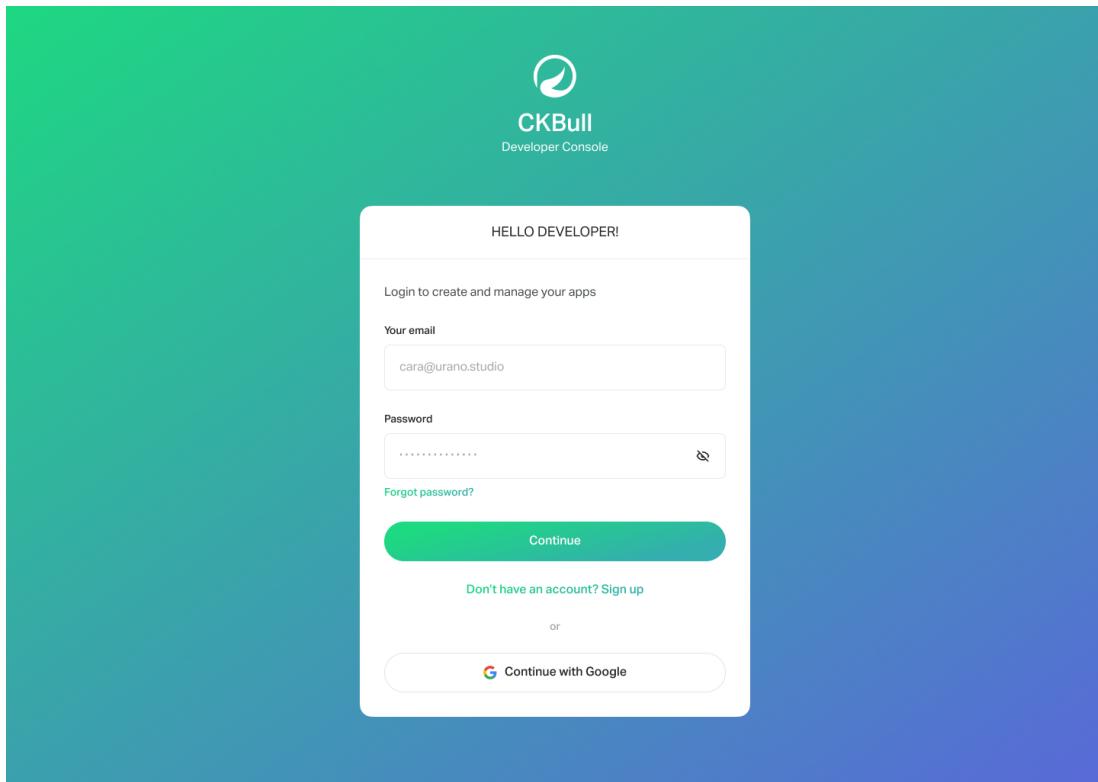


Figura 24: Disseny de pàgina d'inici de sessió.  
Font: Disseny d'URANO.

## Pàgina de verificació de compte

Correspon a la pàgina on un desenvolupador que s'ha enregistrat valida el seu compte mitjançant el codi enviat per correu electrònic.

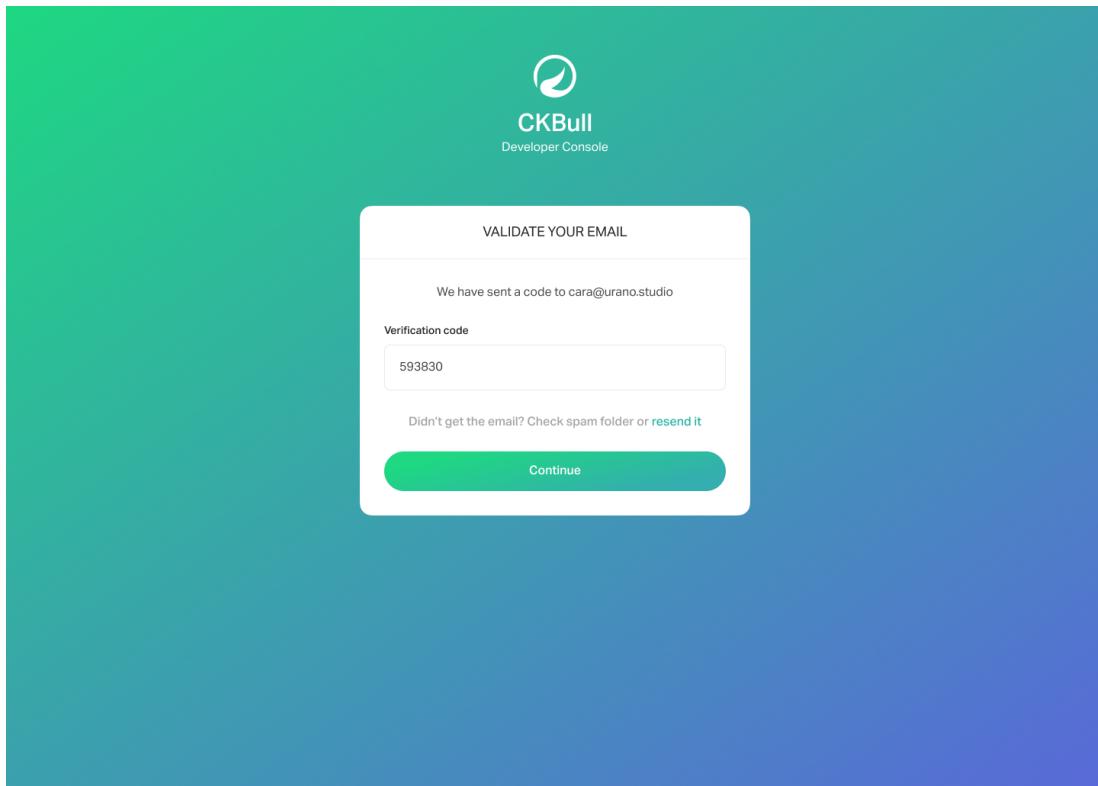


Figura 25: Disseny de pàgina de verificació de compte.  
Font: Disseny d'URANO.

## Pàgina de les meves dApps

Pàgina on el desenvolupador de dApps podrà llistar les dApps que tingui enregistrades al sistema. També podrà crear noves dApps o sortir de la sessió. Per accedir dins d'aquesta pantalla el desenvolupador ha d'estar prèviament identificat.

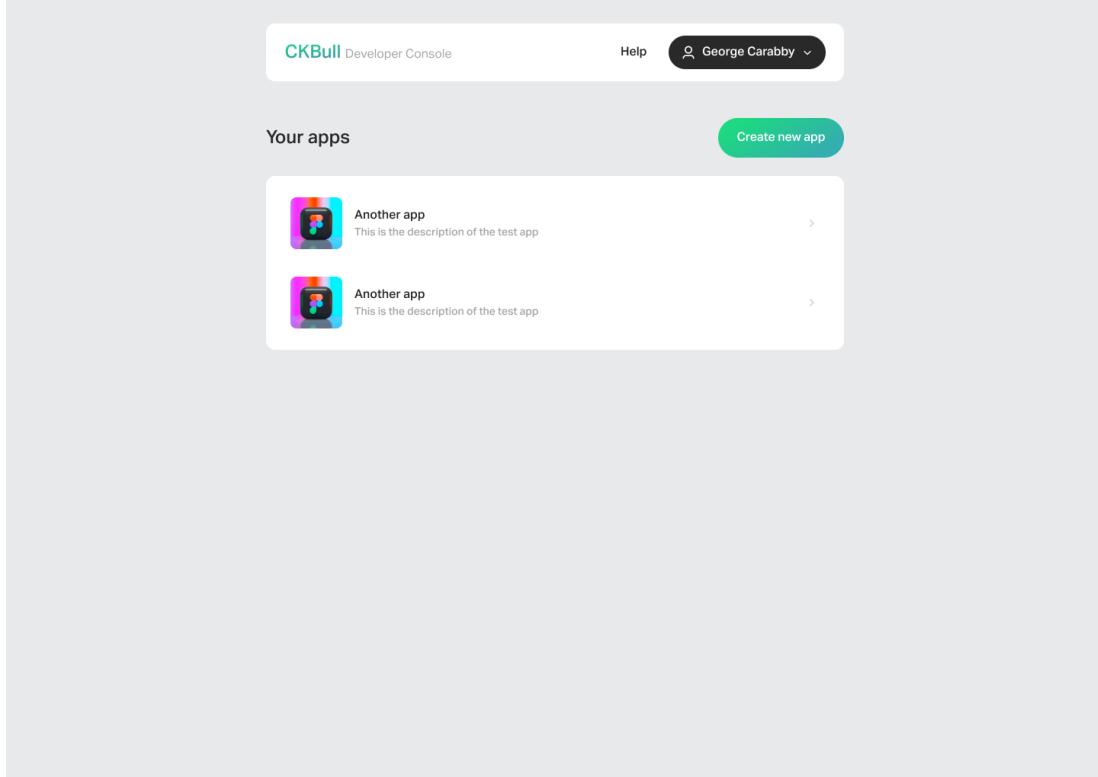


Figura 26: Disseny de pàgina de les meves dApps.  
Font: Disseny d'URANO.

## Pàgina de creació de dApp

Pàgina on el desenvolupador podrà afegir la informació necessària per a enregistrar una nova dApp al sistema. Està composta per un formulari per enregistrar la informació de la dApp.

The screenshot shows the 'Create application' page of the CKBull Developer Console. At the top, there's a header with the CKBull logo, a 'Help' link, and a user profile for 'George Carabby'. Below the header, the title 'Create application' is displayed, followed by a subtitle: 'Register your app and you will get the API key and secret to connect. You only need to enter a few details of your app.' The main form area is titled 'Settings' and contains the following fields:

- App logo:** A placeholder box with a camera icon, labeled 'App logo'.
- Name:** An input field labeled 'Name'.
- Description:** An input field labeled 'Description'.
- Email:** An input field labeled 'Contact email to receive important communications'.
- Project URL:** An input field labeled 'Project url'.
- Support URL:** An input field labeled 'Can also be your Help Center URL, Discord, etc.'
- Terms of Service URL (Optional):** An input field labeled 'Terms of Service URL'.
- Privacy Policy URL (Optional):** An input field labeled 'Privacy Policy URL'.

At the bottom right of the form are two buttons: 'Cancel' and 'Create application' (highlighted in green).

Figura 27: Disseny de creació de dApps.  
Font: Disseny d'URANO.

## Modal per visualitzar les credencials d'una dApp

Aquest modal es troba dins de la pàgina de creació d'una nova dApp i es mostra a l'usuari quan s'ha creat amb èxit la dApp. El modal mostra les credencials que de la dApp haurà d'utilitzar per a comunicar-se posteriorment amb l'API per a generar peticions.

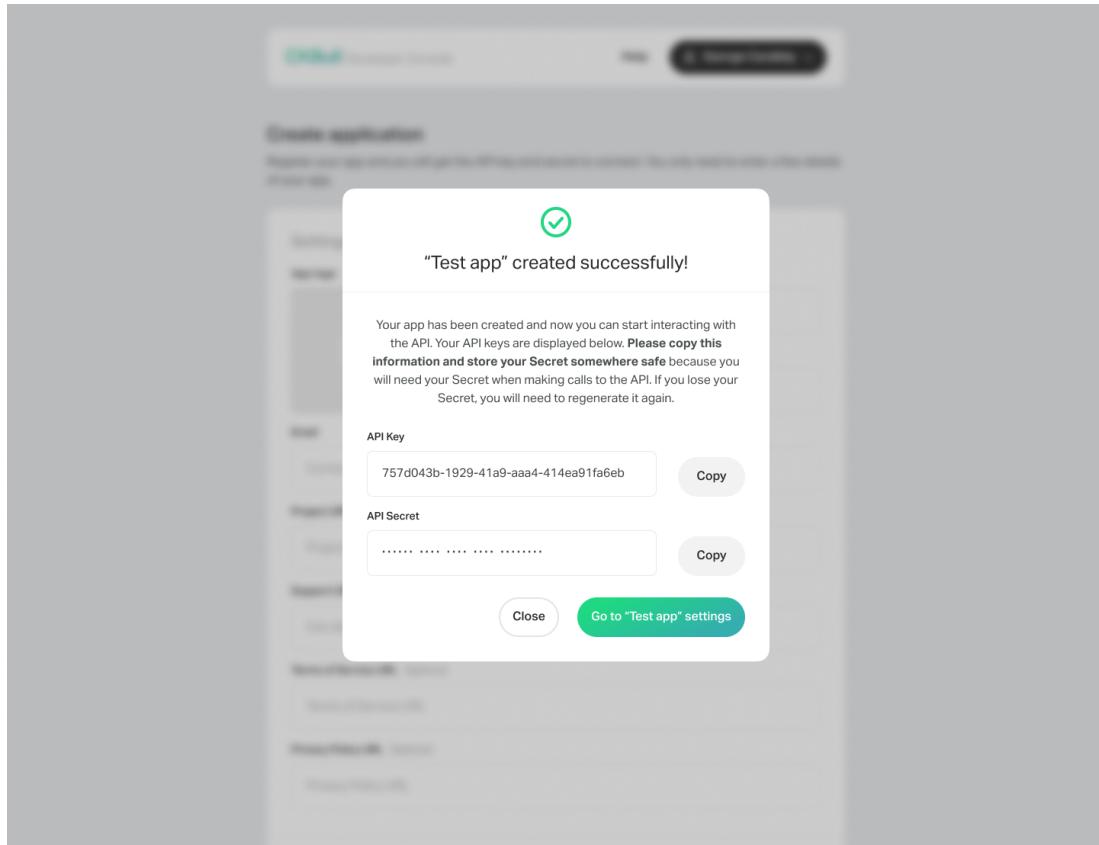


Figura 28: Disseny del modal de credencials de la dApp.  
Font: Dissenys d'URANO.

## Pàgina d'edició de dApp

Correspon a la pàgina on el desenvolupador podrà modificar parcial o completament la informació de la seva dApp. Aquesta pàgina inclou la regeneració de l'API *secret*.

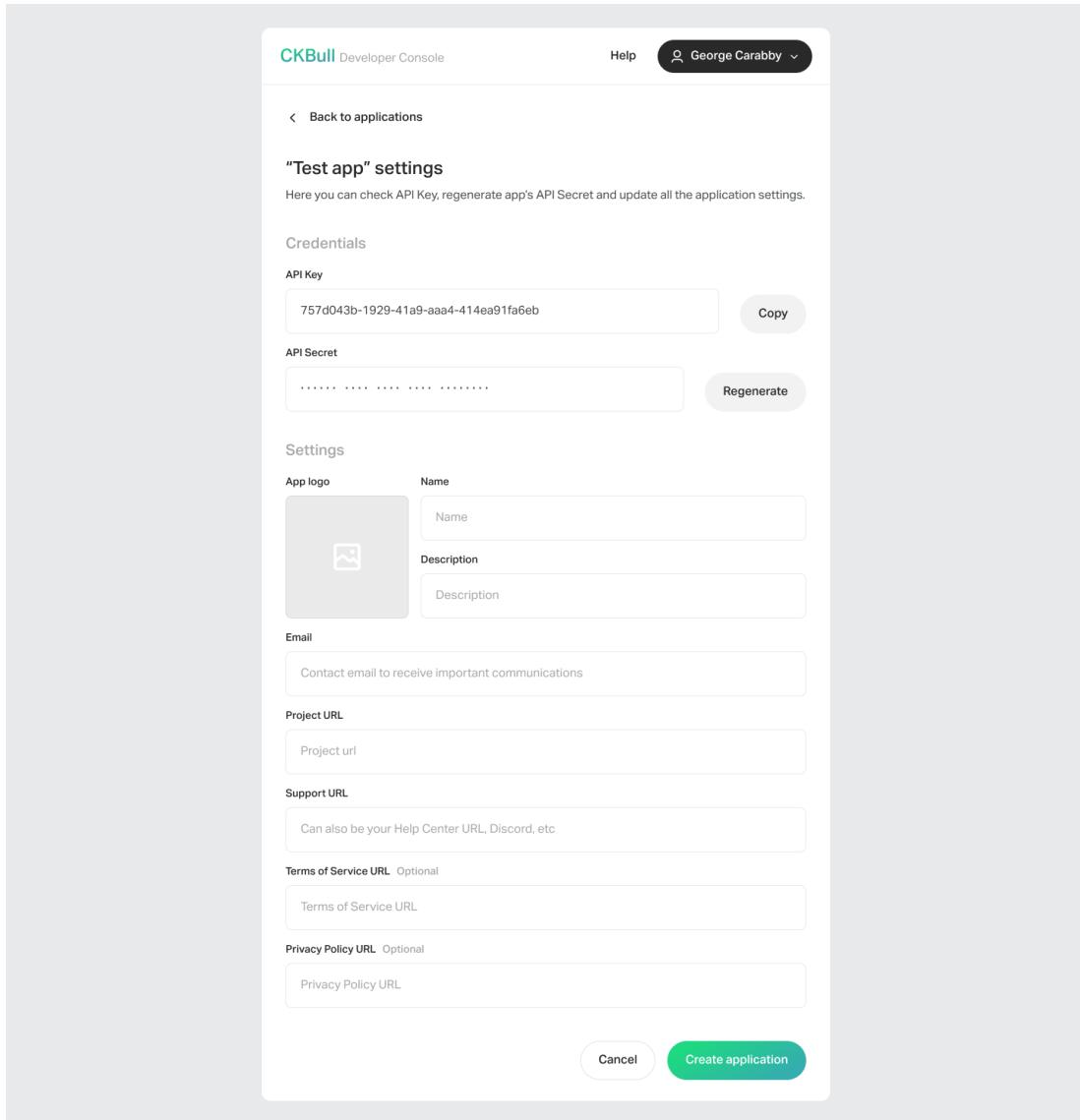


Figura 29: Disseny de pàgina d'edició de dApp.  
Font: Disseny d'URANO.

## Disseny de la capa de domini

Un cop definida la capa de presentació, la següent capa a definir és la capa de domini. En ser CKB Developer Console un producte nou, s'ha realitzat un disseny des de 0, fent possible dissenyar una millor arquitectura en comparació a *CKBull Wallet*. Així doncs, la capa de domini es defineix a la Figura 30.

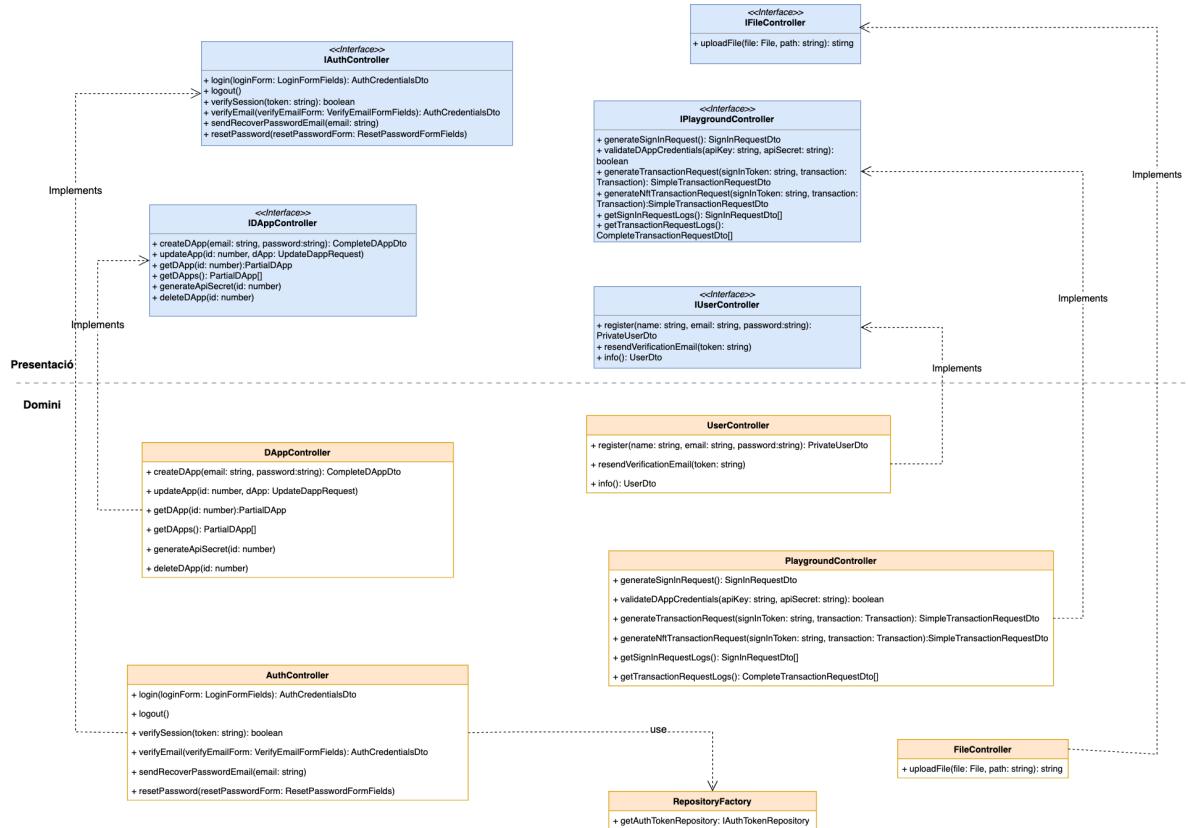


Figura 30: Connexió entre capa de presentació i domini CKBull Developer Console.

Font: elaboració pròpria.

La comunicació entre capes dins de l'arquitectura es realitza mitjançant interfícies. D'aquesta manera aconseguim que el sistema segueixi el **príncipi de segregació d'interfícies** (*Interface Segregation Principle*) dels principis **SOLID** [36]. Cada capa conté les interfícies dels objectes a connectar de la capa inferior. En aquest cas, la capa de presentació utilitzà les interfícies per utilitzar la lògica de domini, definida amb controladors.

La lògica de domini s'agrupa segons la funcionalitat, aplicant el **príncipi de responsabilitat única** (*Single Responsibility Principle*) dels principis SOLID. A la Figura 30 es visualitzen 5 controladors de domini:

- **AuthController**: Gestiona la lògica d'autenticació i autorització de l'usuari.
- **UserController**: Encarregat de gestionar la lògica relacionada amb la informació de l'usuari.
- **DAppController**: Gestiona la lògica relacionada amb les dApps generades per l'usuari.
- **FileController**: Encarregat de gestionar lògica entorn a fitxers que se'n pugui a la web.
- **PlaygroundController**: Encarregat de gestionar totes les peticions que es realitzin al *playground* de la web.

Mantenint el principi de segregació d'interfícies, els controladors de domini també utilitzant les interfícies de la capa d'accés a dades. A la següent figura podem observar quines interfícies connectem amb els controladors definits anteriorment i la resta de classes que trobem a domini.

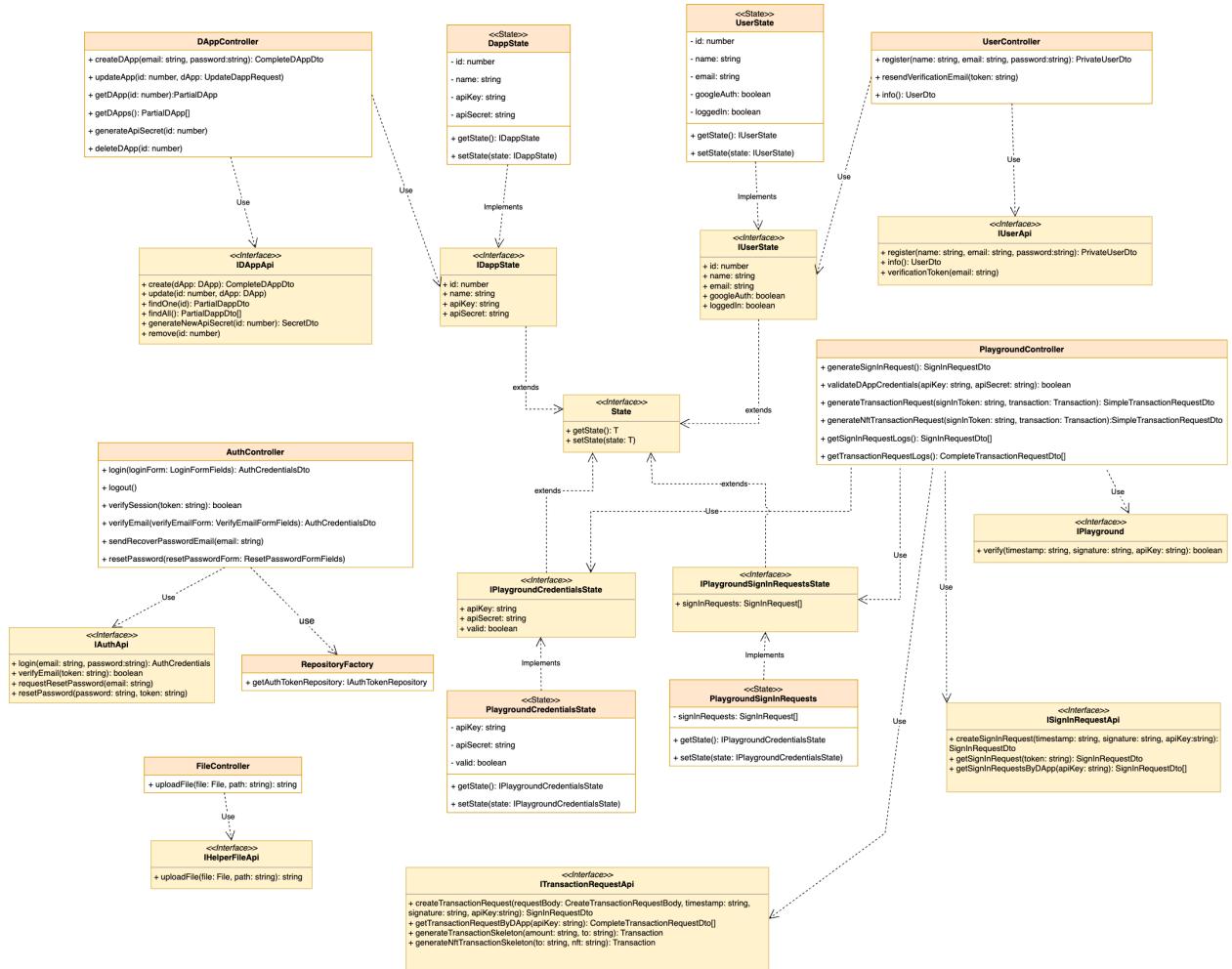


Figura 31: Disseny de capa de domini de CKBull Developer Console.  
Font: elaboració pròpia.

A més dels controladors, també trobem els estats o *States*, que correspon a la informació que gestiona la capa de domini que és utilitzada per les interfícies a través dels controladors, i la classe *RepositoryFactory*. Aquesta classe correspon a una factoria per obtenir instàncies de les classes de la capa de persistència. Només és usat pel *AuthController* per a aconseguir el repositori encarregat d'emmagatzemar el *token* d'accés de l'usuari.

## Disseny de capa d'accés a dades

Com a última capa de la interfície trobem la capa d'accés a dades. En aquesta capa es troben principalment classes que comuniquen amb l'API de domini (CKBull Signer API) i repositoris que emmagatzemen informació a la mateixa interfície.

Tal com està definida la capa de domini, la capa d'accés a dades també usa interfícies per a definir la comunicació entre les dues capes. La Figura 32 mostra aquesta connexió.

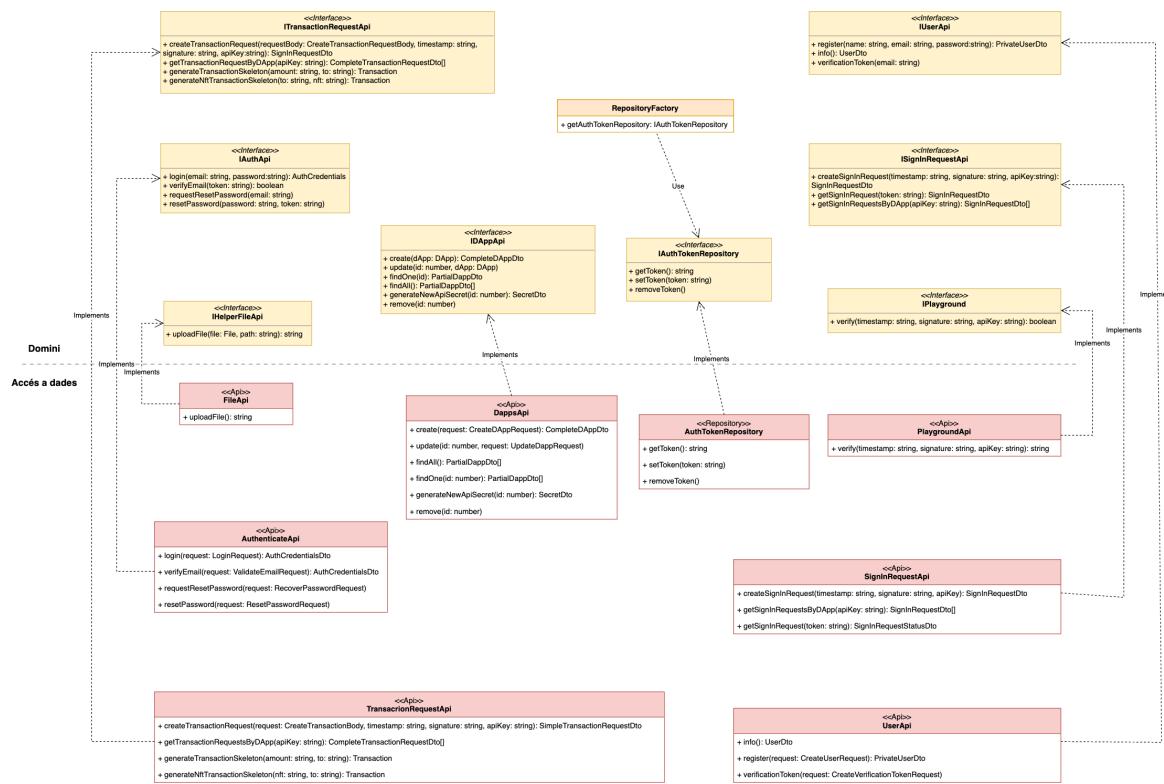


Figura 32: Conexió entre capa de domini i accés a dades CKBull Developer Console.  
Font: elaboració pròpria.

## 10.2.2 Disseny de CKBull Wallet

A diferència del client CKBull Developer Console, CKBull Wallet és un *software* que ja existia prèviament a la data d'inici del projecte i per al que només cal implementar la nova funcionalitat. Per tant, el disseny de la nova funcionalitat s'ha d'adaptar a tant a l'arquitectura implementada com a l'aspecte visual de l'aplicació.

### Arquitectura de CKBull Wallet

CKBull Wallet és una cartera digital *blockchain* implementada com a aplicació mòbil que es va iniciar a principis de Febrer de l'any 2022. Aquesta aplicació no es basa en cap arquitectura en capes i connecta directament la interfície, la lògica i l'accés a dades. Tot i poder fer un *refactor* de l'arquitectura, no s'ha decidit incloure aquesta tasca degut als esforços que suposa. Així doncs, per al desenvolupament de la nova funcionalitat s'ha decidit seguir les pautes del disseny actual de l'aplicació.

### Diagrama de navegabilitat

A continuació, es llisten totes les navegabilitats que es produueixen dins de l'aplicació involucrada en la nova funcionalitat. La llegenda de colors dels blocs del diagrama correspon a la següent:

- **Vermell:** Pàgines que existien prèviament a CKBull Wallet abans d'implementar la nova funcionalitat.

- **Verd:** Pàgines noves creades per a la funcionalitat.
- **Lila:** Modals i diàlegs nous utilitzats dins de la funcionalitat.
- **Groc:** Pestanyes (*tabs*), incloses dins d'una pantalla, que contenen components o informació.
- **Blanc:** Vista d'un servei extern a l'aplicació.

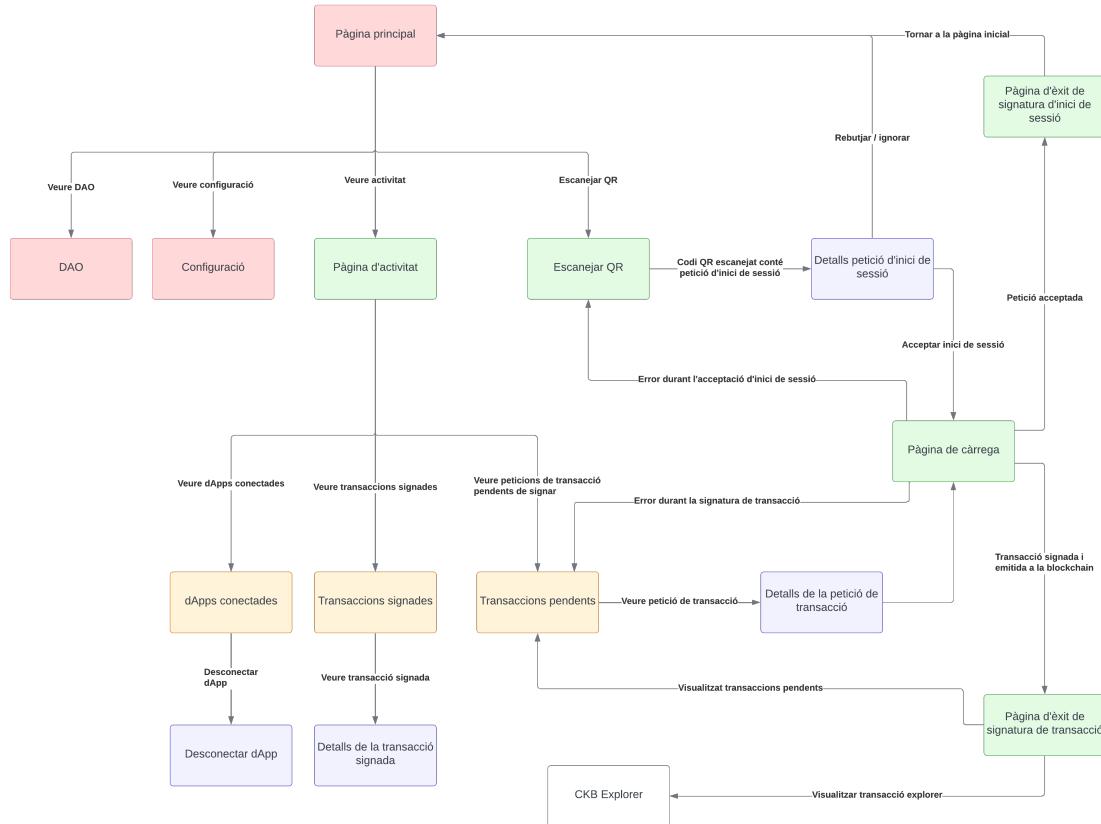


Figura 33: Diagrama de navegació de CKBull Wallet.

Font: elaboració pròpia.

La navegació es divideix principalment en 2 camins: la visualització de la informació (dApps connectades, transaccions pendents i transaccions signades) i la signatura de peticions (inici de sessió i transacció). Tal com indica la Figura 33, gran part de la informació és representada a través de modals o diàlegs i no amb pàgines. D'aquesta manera, una navegabilitat que pot semblar complexa es limita únicament a 4 pàgines completes, que corresponen a la pàgina d'activitat, escanear el QR, la pàgina de càrrega i les pàgines d'exit de signatures d'inici de sessió i transacció.

## Dissenys externs

Per a les funcionalitats a desenvolupar a CKBull Wallet, l'empresa Urano també va facilitar una sèrie de dissenys externs de la interfície com a estàndards. A continuació es mostren la col·lecció parcial de dissenys. En alguns casos s'adjunten dos dissenys de la mateixa funcionalitat, a causa dels diferents modes que disposa l'aplicació: mode clar i mode fosc.

## Pàgina d'activitat amb pestanya de transaccions pendents

Pàgina accessible des de la pàgina principal. Conté les 3 pestanyes (dApps connectades, transaccions signades i transaccions pendents) a més d'informació de la cartera.

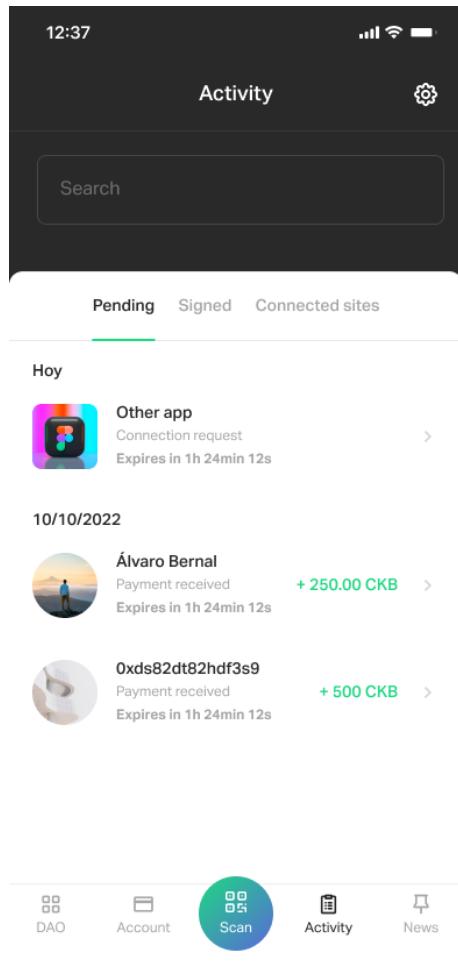
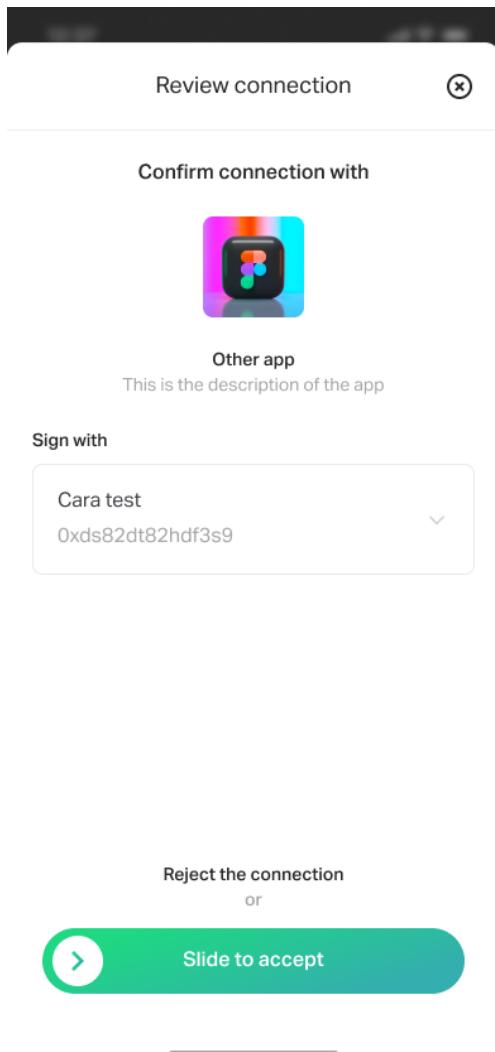


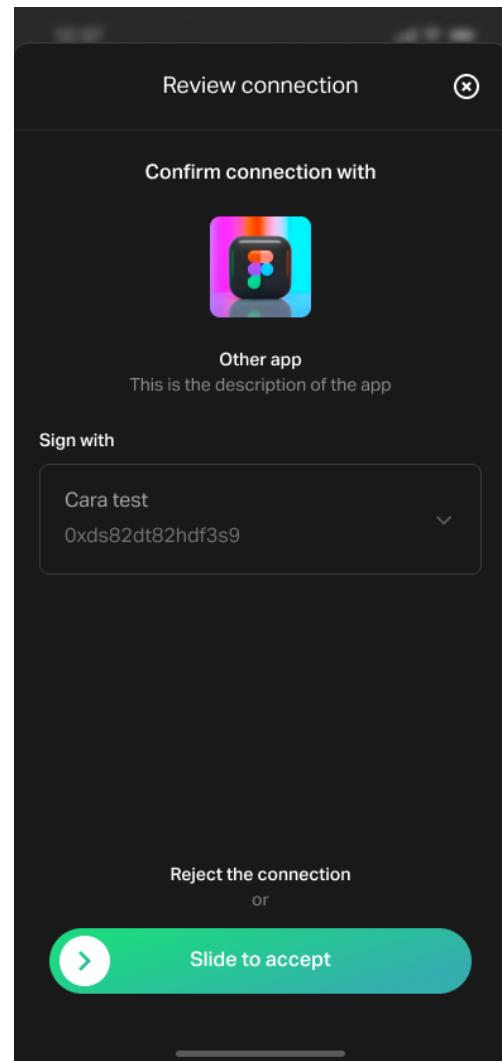
Figura 34: Disseny de la pàgina d'activitat.  
Font: Disseny d'URANO.

## Modal d'acceptació d'inici de sessió

Modal que es mostra després d'escanejar un codi QR que conté una petició d'inici de sessió. Es mostren les dades principals de la dApp a la que es vol iniciar sessió i es permet seleccionar amb quin compte es vol iniciar sessió.



(a) Disseny en mode clar

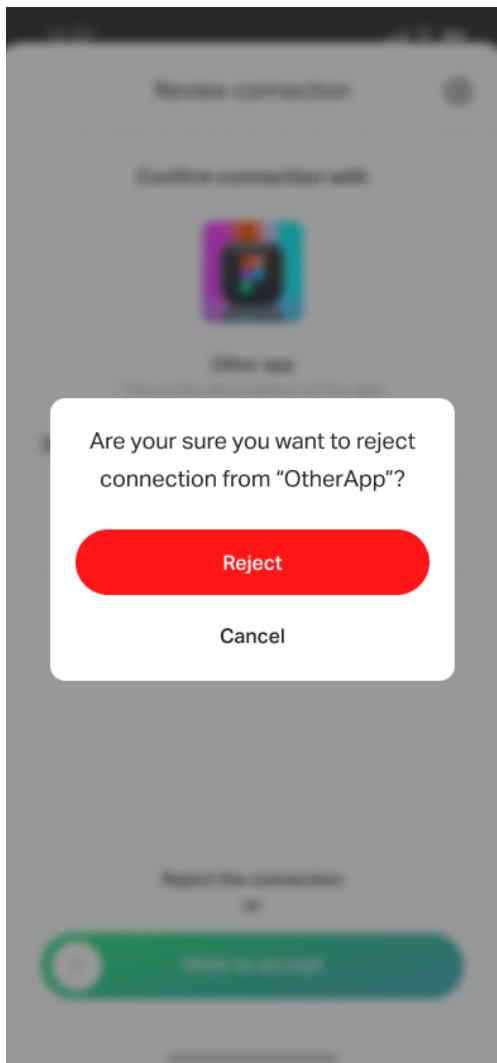


(b) Disseny en mode fosc

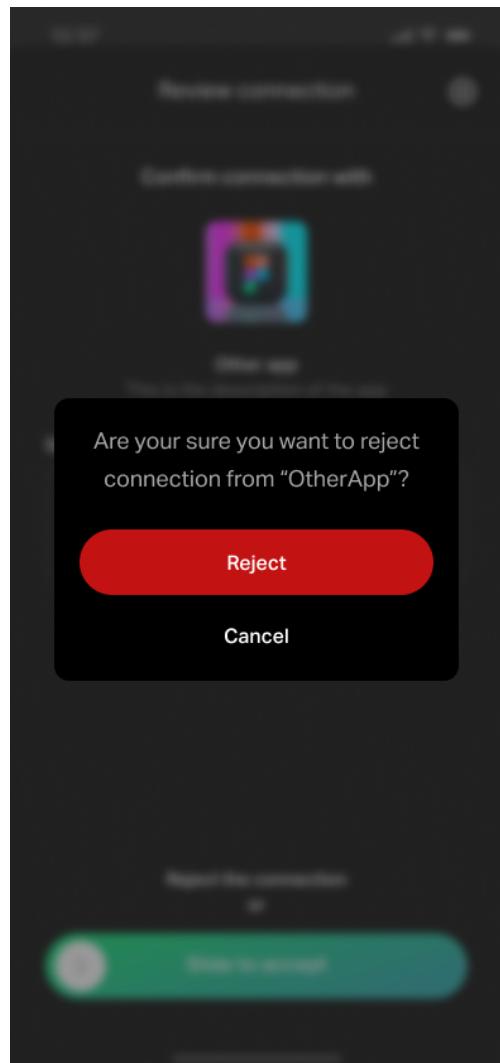
Figura 35: Dissensys del modal d'inici de sessió.  
Font: Dissensys d'URANO.

### Dissenys del diàleg de reacció de d'inici de sessió

Diàleg que permet rebutjar una petició d'inici de sessió. S'inclou dins del modal de petició d'inici de sessió.



(a) Disseny en mode clar



(b) Disseny en mode fosc

Figura 36: Dissenys del diàleg de declinació d'inici de sessió.  
Font: Dissenys d'URANO.

#### Disseny de pàgina d'exit de signatura de transacció

Dissenys de la pàgina que es mostra en signar i emetre la transacció amb èxit. Inclou un botó d'enllaç cap al servei extern *CKB Explorer* que permetrà comprovar la transacció dins de la *blockchain*.

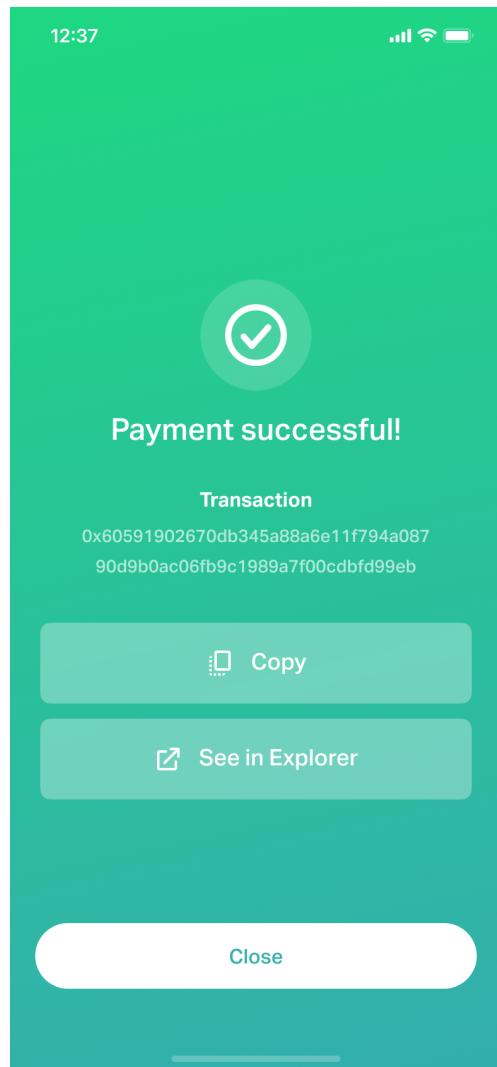


Figura 37: Disseny de pàgina d'èxit de signatura de transacció.  
Font: Dissenys d'URANO.

## 10.3 Disseny de domini

La capa de domini del sistema conté únicament un component, CKBull Signer API, que comunica les interfícies i les dades a la capa de persistència.

### 10.3.1 Disseny de l'API

El següent component a dissenyar és CKBull Signer API, l'únic component de la capa de domini del sistema. Com a CKBull Developer Console, aquest component s'ha creat des de 0 per al projecte i, per tant, s'ha hagut de dissenyar totalment.

#### Arquitectura de l'API

Com s'ha definit a anteriors components, s'ha utilitzat l'arquitectura de capes per a estructurar l'API. S'ha decidit agrupar les capes per funció per a mantenir el principi de responsabilitat única.

- **Controladors:** Punts d'entrada de l'API. Connecten les peticions de les interfícies amb els serveis de l'API.
- **Serveis:** Classes que proporcionen accions sobre els recursos que gestiona l'API.
- **Repositoris:** Objectes que permeten accedir a les dades emmagatzemades a la capa de persistència.

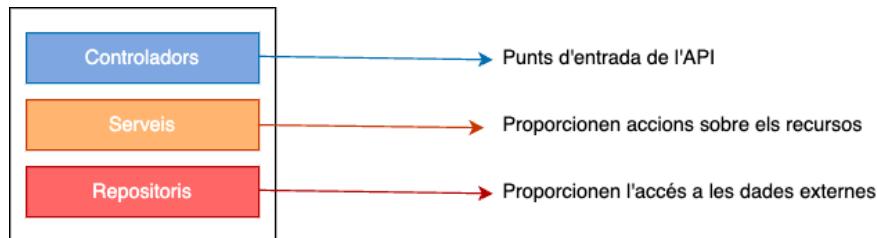


Figura 38: Arquitectura de CKBull Signer API.

Font: elaboració pròpia.

#### Diagrama de la capa de controladors

Com s'ha descrit a l'apartat anterior, la capa de controladors representa el punt d'entrada de les peticions de l'API. Aquesta capa no ha de ser la responsable de realitzar les peticions de la crida, sinó únicament verificar l'autenticació i autorització de les peticions i cridar els serveis que faran les accions corresponents.

Els controladors s'han agrupat segons el recurs sobre el qual actuen. A la Figura 39 es veuen tots els controladors i la connexió amb els interfícies dels serveis que utilitzen.

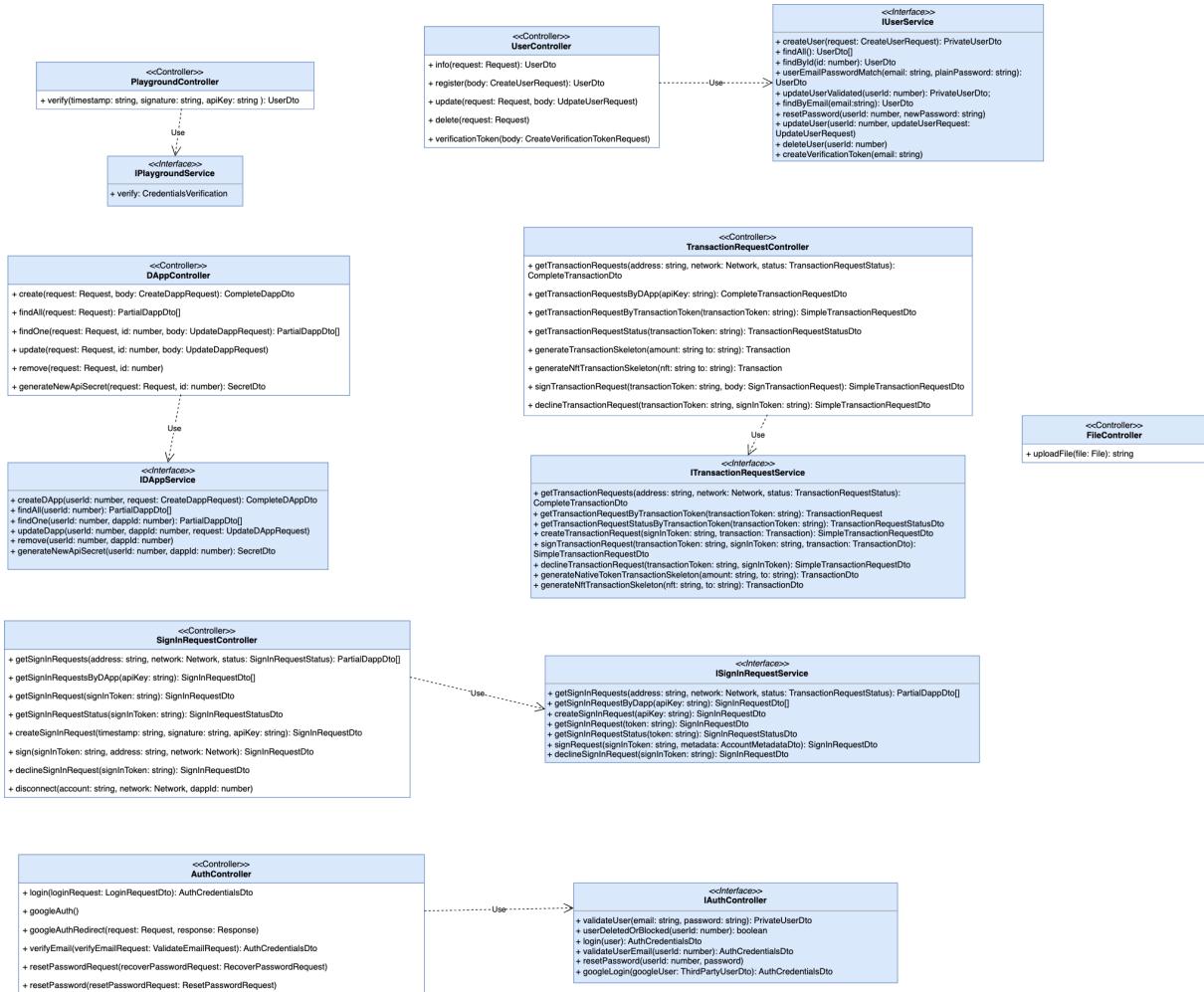


Figura 39: Diagrama de la capa de controladors de l'API.  
Font: elaboració pròpria.

Ja que l'API s'implementarà sobre el protocol HTTP, els controladors tindran unes direccions on els usuaris podran demanar els recursos. Aquestes rutes es poden trobar a l'annex Secció B.1

## Diagrama de la capa de serveis

Les classes incloses dins d'aquesta capa corresponen als serveis de l'API que efectuen operacions sobre els recursos. Dins d'aquesta capa no s'haurien de fer comprovacions a nivell d'autorització i autenticació. El diagrama resultant d'aquesta capa és el següent.

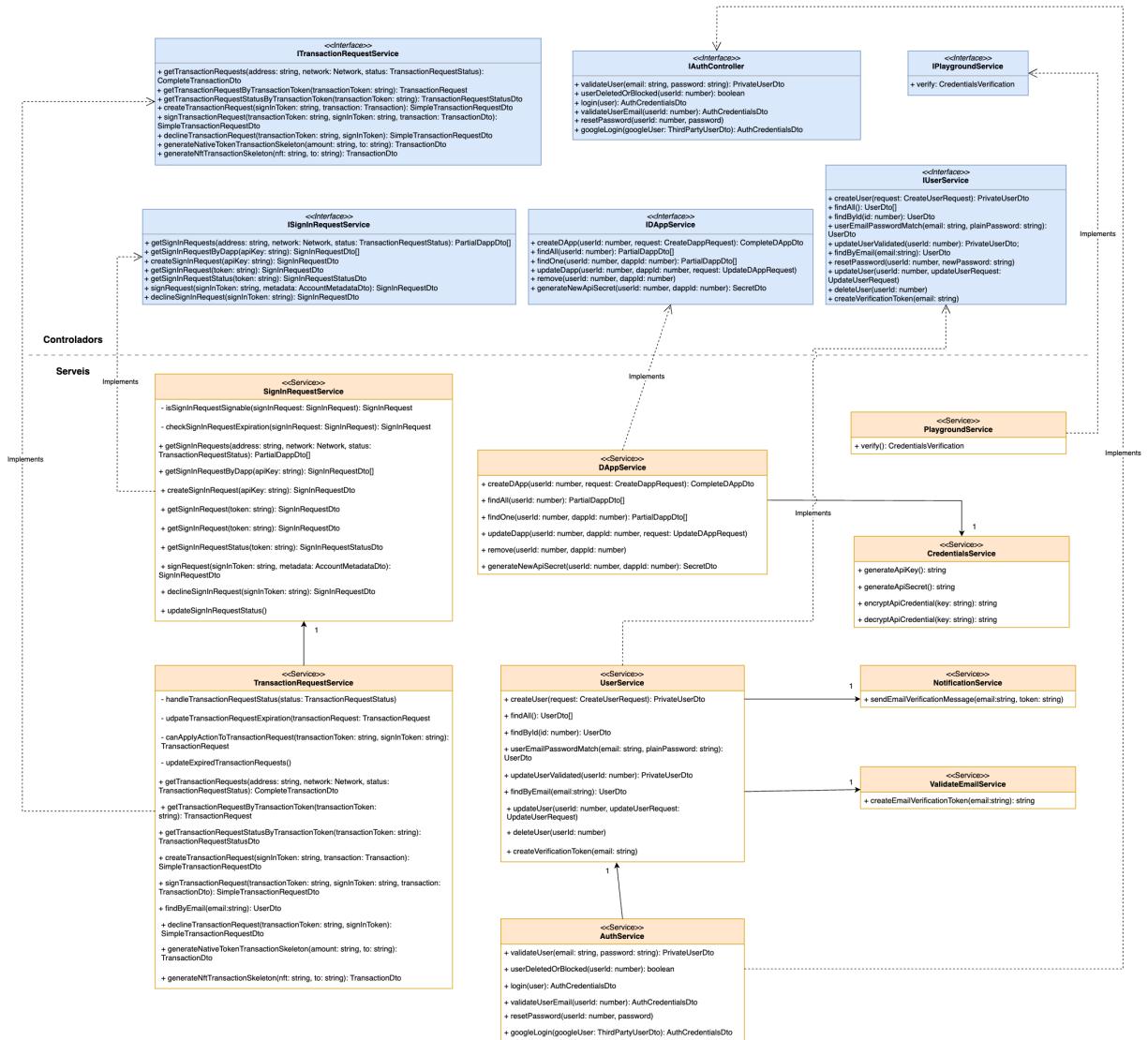


Figura 40: Diagrama de serveis de l'API.

Font: elaboració pròpria.

Com es pot veure a la Figura 40, s'ha intentat produir el menor acoblament entre serveis. Únicament TransactionRequestService, DappService i UserService utilitzen altres serveis per a fer crides. No obstant això, els serveis connecten també amb la capa d'accés a dades. Les connexions dels serveis amb les interfícies es defineix a la següent figura.

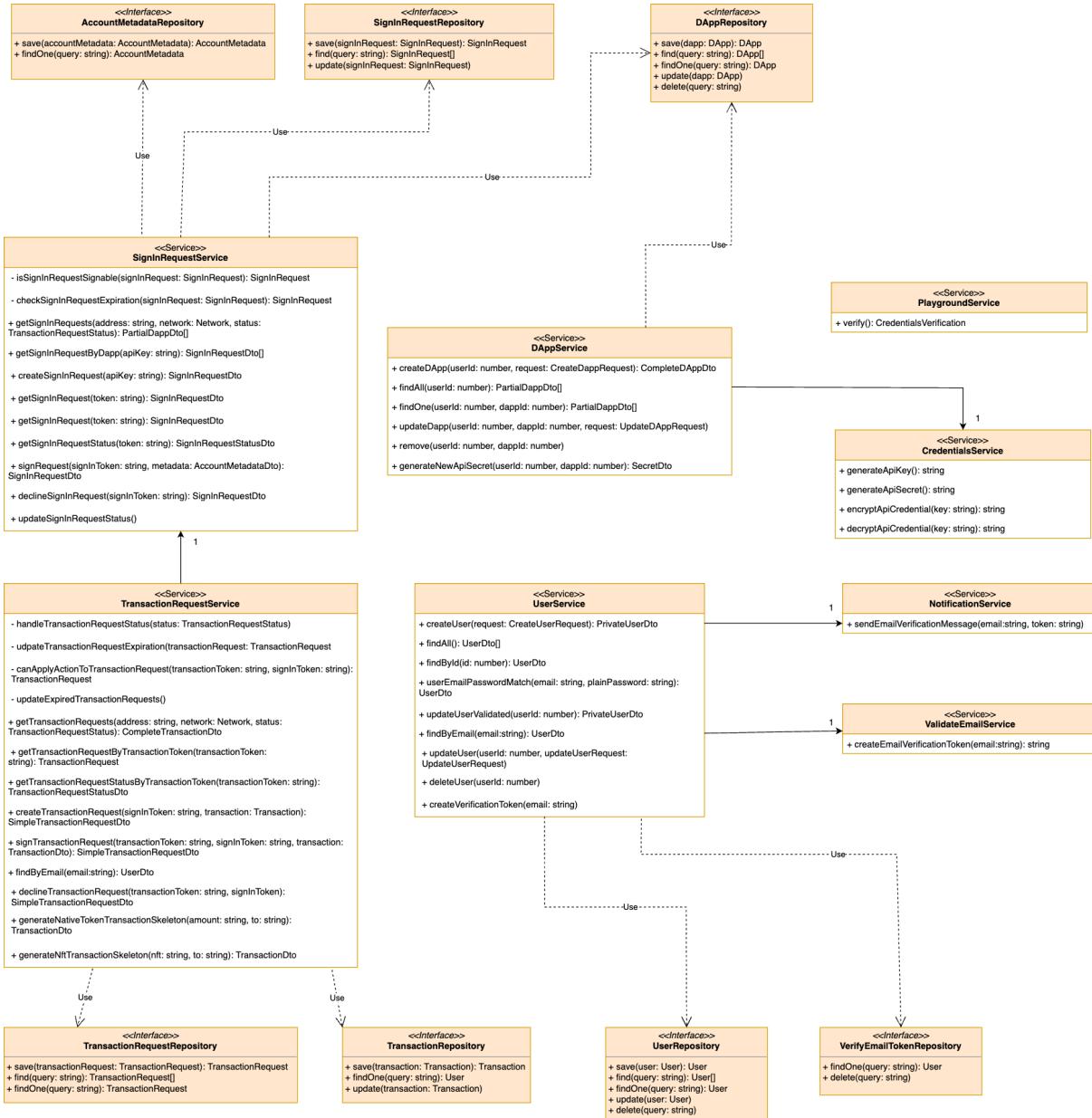


Figura 41: Diagrama de serveis amb interfícies de repositoris.

Font: elaboració pròpia.

## Diagrama de de la capa de repositoris

Per finalitzar l'arquitectura de CKBull Signer API, definim la capa de repositoris. Aquesta capa únicament conté classes repositori, encarregades de l'accés a les dades de la capa de persistència (del sistema). A més, conté classes *entity* (entitat) que representen quins recursos utilitzen els repositoris.

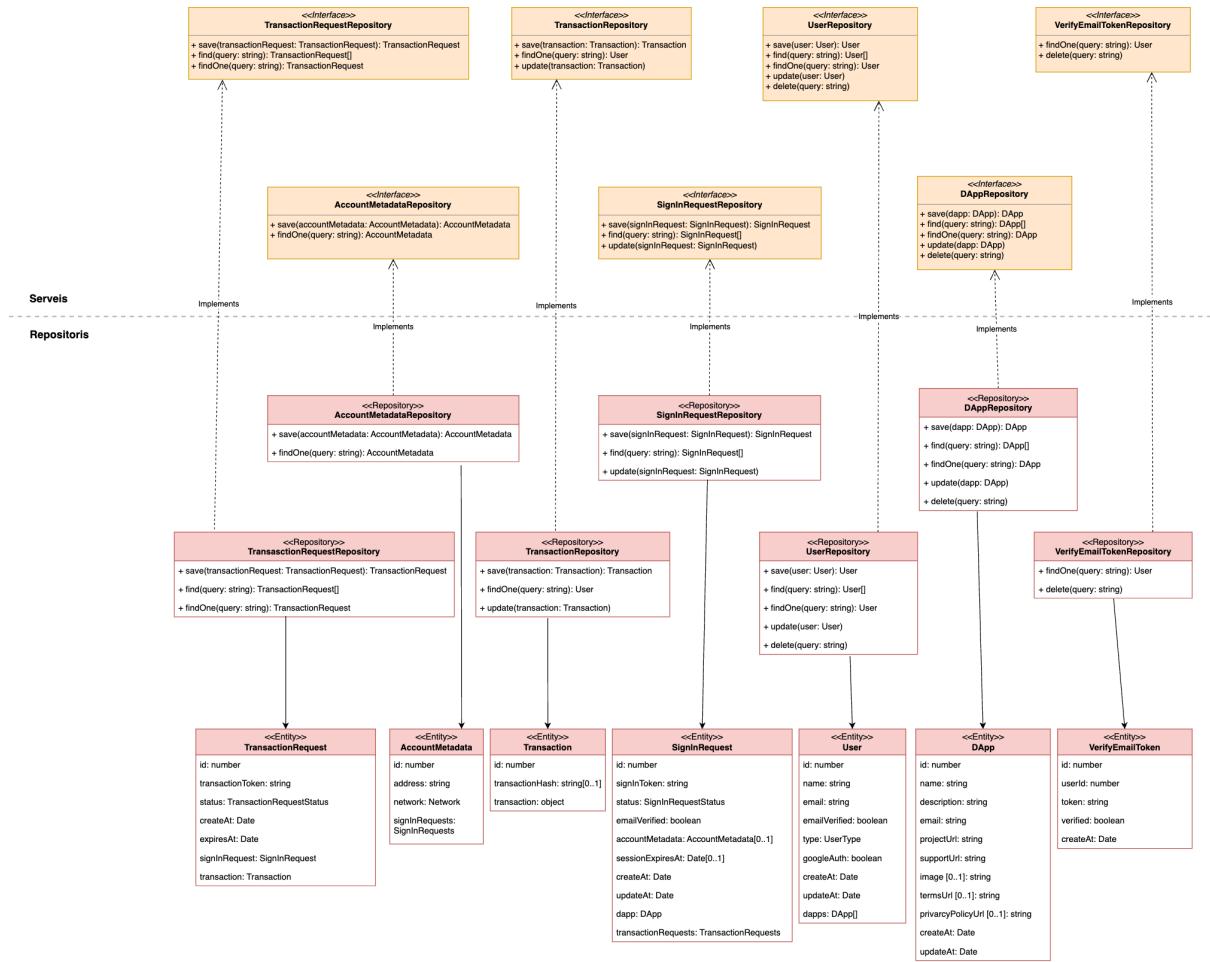


Figura 42: Diagrama de repositoris de l'API.  
Font: elaboració pròpia.

## 10.4 Disseny de la persistència

L'última capa de l'arquitectura del sistema que queda per dissenyar és la capa de persistència. Dins d'aquesta capa es troben tots els components que emmagatzemem dades importants per al sistema que són utilitzades per les interfícies o per l'API. L'únic component de la capa que cal dissenyar és la base de dades, ja que el contenidor S3 d'AWS ja es trobava prèviament configurat i s'usa en altres projectes de Peersyst com a repositori per emmagatzemar continguts com imatges i fitxers.

### 10.4.1 Disseny de la base de dades

Per a fer un disseny que compleixi els requisits funcionals del sistema, cal tindre en compte els següents factors:

- **Rapidessa:** La transmissió d'informació entre l'API i la base de dades ha de produir-se de la manera més ràpida possible.
- **Flexibilitat a l'accés de dades:** En ser la primera versió d'un producte, és possible que durant el temps s'afegeixin funcionalitats addicionals a les definides a la memòria d'aquest projecte. Així doncs, cal fer servir una base de dades que ofereixi flexibilitat sobre les consultes als recursos.

Dels diferents tipus de base de dades que existeixen, l'opció seleccionada ha resultat una **base de dades relacional**. Amb aquesta decisió, aconseguim satisfer per a aquesta versió del projecte la rapidesa de la informació i la flexibilitat a l'accés de dades, ja que aquestes s'emmagatzemem en un model de taules (amb files i columnes).

Les dades que emmagatzemarem corresponen a les entitats usades pels repositoris que s'han descrit a la capa de domini (Figura (42)).

#### Descripció general de les taules

A continuació, es descriuràn les restriccions sobre els atributs de cada taula de la base de dades. La Figura 43 mostra tota la informació completa de les taules.

La taula AccountMetadata emmagatzema la informació que aporta un usuari de la *wallet* quan accepta una petició d'inici de sessió per part d'una dApp. La seva **clau primària** és l'atribut *id* (autogenerat) i consta dels atributs *address* (adreça del compte de l'usuari) i *network* (mainnet o testnet). La combinació **d'address i network ha de ser única per evitar valors repetits**.

Taula	AccountMetadata
Atributs	
<i>id</i>	PRIMARY KEY
<i>address</i>	NOT NULL
<i>network</i>	NOT NULL
<i>address + network</i>	UNIQUE

Taula 50: Restriccions de la taula AccountMetadata.

La taula SignInRequest emmagatzema les peticions d'inici de sessió generades per les dApps i actualitza el seu estat segons l'acció dels usuaris de les *wallet*. La seva **clau primària** correspon

a l'atribut *id* (autogenerat) i l'atribut *signInToken* sent únic, ja que fa referència al SignInToken. Conté **dues claus externes** que fan referència a les taules DApp i AccountMetadata, que referencien a quina dApp correspon la petició d'inici de sessió i qui ha acceptat la petició.

Taula	SignInRequest
Atributs	
<i>id</i>	PRIMARY KEY
<i>signInToken</i>	NOT NULL UNIQUE
<i>status</i>	NOT NULL
<i>created_at</i>	NOT NULL
<i>expires_at</i>	NOT NULL
<i>dapp_api_key</i>	NOT NULL REFERENCES DApp
<i>metadataId</i>	REFERENCES AccountMetadata

Taula 51: Restriccions de la taula SignInRequest.

La taula TransactionRequest guarda les peticions de transacció generades per una dApp a partir d'una SignInRequest. Per aquesta raó, podem trobar que la taula conté una **clau externa cap a SignInRequest** que indica a quina SignInRequest està lligada la petició. A més, la seva **clau primària** també correspon a un identificador *id* (autogenerat) i conté una altra **clau externa cap a la taula Transaction**, que referencia la transacció a signar.

Taula	TransactionRequest
Atributs	
<i>id</i>	PRIMARY KEY
<i>transactionToken</i>	NOT NULL UNIQUE
<i>status</i>	NOT NULL
<i>created_at</i>	NOT NULL
<i>expires_at</i>	NOT NULL
<i>signInRequestToken</i>	NOT NULL REFERENCES SignInRequest
<i>transactionId</i>	NOT NULL REFERENCES Transaction

Taula 52: Restriccions de la taula TransactionRequest.

La taula Transaction conté informació rellevant sobre les transaccions que se signen mitjançant la plataforma. La seva **clau primària** correspon a un identificador *id* (autogenerat). També disposa del camp *transaction*, que emmagatzema l'objecte que conté tota la informació de la transacció, i un camp *transactionHash* que deixa de ser *null* quan la transacció és signada.

Taula	Transaction
Atributs	
<i>id</i>	PRIMARY KEY
<i>transaction</i>	NOT NULL

Taula 53: Restriccions de la taula Transaction.

La taula DApp és l'encarregada d'emmagatzemar la informació de la dApp, proporcionada pel desenvolupador (objecte de la taula User). Com a la resta de taules, la seva **clau primària**

correspon a l'atribut *id* (autogenerat). Disposa de diferents atributs únics, com l'atribut *name* i *apiKey*. Per referenciar l'usuari creador de la dApp, conté una **clau externa sobre l'atribut *userId*** cap a la taula User.

Taula	DApp
Atributs	
<i>id</i>	PRIMARY KEY
<i>name</i>	NOT NULL UNIQUE
<i>description</i>	NOT NULL
<i>email</i>	NOT NULL
<i>projectUrl</i>	NOT NULL
<i>supportUrl</i>	NOT NULL
<i>created_at</i>	NOT NULL
<i>updated_at</i>	NOT NULL
<i>apiKey</i>	NOT NULL UNIQUE
<i>apiSecret</i>	NOT NULL
<i>userId</i>	NOT NULL REFERENCES USER

Taula 54: Restriccions de la taula DApp.

La taula User fa referència a l'actor *Desenvolupador de dApps*. La seva **clau primària** correspon a un identificador *id* (autogenerat). A més, l'atribut *email* conté una restricció d'únic.

Taula	User
Atributs	
<i>id</i>	PRIMARY KEY
<i>name</i>	NOT NULL
<i>email</i>	NOT NULL UNIQUE
<i>email_verified</i>	NOT NULL
<i>password</i>	NOT NULL
<i>type</i>	NOT NULL
<i>googleAuth</i>	NOT NULL
<i>created_at</i>	NOT NULL
<i>updated_at</i>	NOT NULL

Taula 55: Restriccions de la taula User.

La taula VerifyEmailToken emmagatzema els tokens de verificació dels usuaris quan aquests creen un compte. La seva **clau primària** correspon a un identificador *id* (autogenerat) i compta amb una **clau externa sobre l'atribut *userId*** cap a la taula User, que indica quin usuari ha de verificar el seu compte. Un cop l'usuari es verificat, l'atribut *verified* es torna *true*.

Taula	VerifyEmailToken
Atributs	
id	PRIMARY KEY
userId	NOT NULL UNIQUE REFERENCES USER
token	NOT NULL
verified	NOT NULL
created_at	NOT NULL

Taula 56: Restriccions de la taula VerifyEmailToken.

Per finalitzar, la taula ResetToken emmagatzema els *tokens* que permeten als usuaris (User) la seva contrasenya (*password*). La seva **clau primària** és un identificador id (autogenerat) i, com a la taula anterior, també disposa d'una clau externa sobre l'atribut *userId* cap a la taula User.

Taula	ResetToken
Atributs	
id	PRIMARY KEY
userId	NOT NULL UNIQUE REFERENCES USER
token	NOT NULL
expiration	NOT NULL
verified	NOT NULL
created_at	NOT NULL
updated_at	NOT NULL

Taula 57: Restriccions de la taula ResetToken.

## Diagrama de les taules de la base de dades

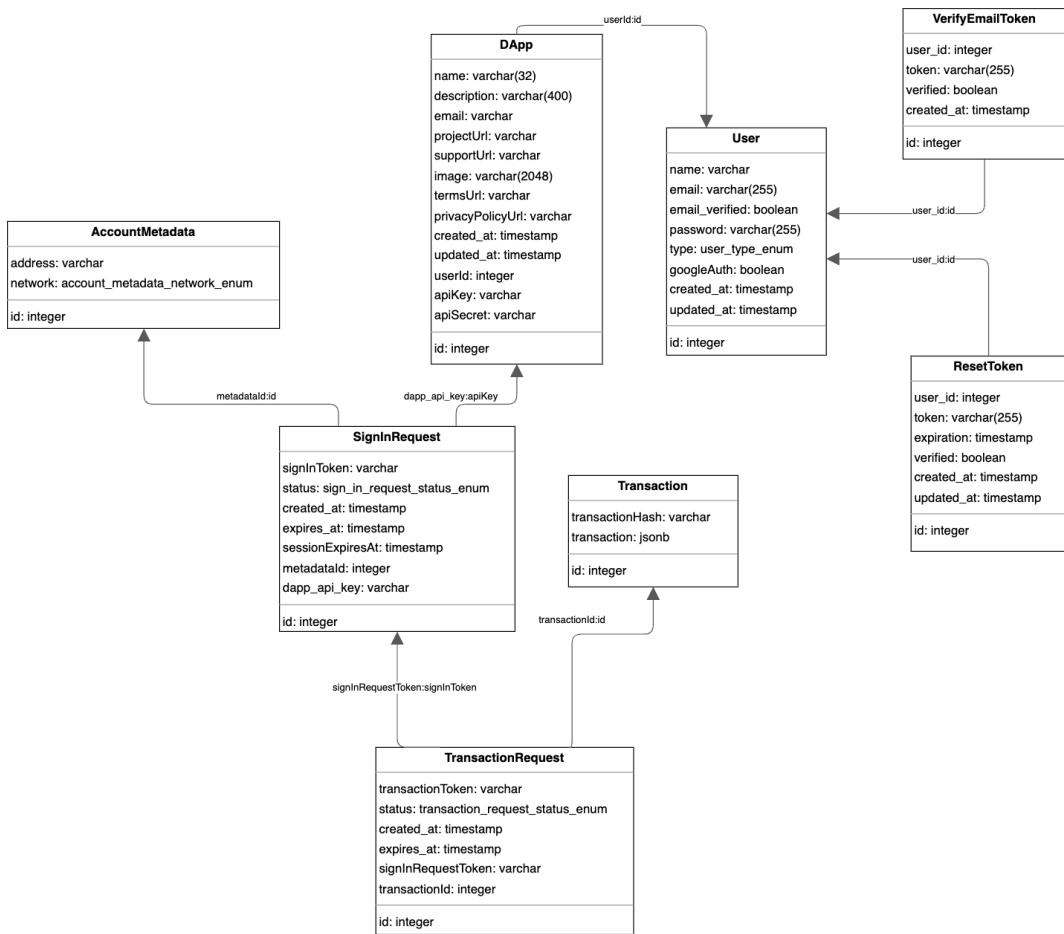


Figura 43: Disseny de les taules de la base de dades.  
Font: elaboració pròpia.

## 10.5 Patrons de disseny

A mode de resum del disseny del sistema, s'inclouen a continuació tots els patrons de disseny utilitzats. L'ús dels patrons de disseny dins de l'arquitectura del projecte és degut a l'aplicació de bones pràctiques que permeten produir un codi escalable i flexible. Aquests patrons es veuran categoritzats segons el tipus de patró que siguin: **creacionals** o **estructurals**.

### 10.5.1 Patrons creacionals

- **Singleton**: Patró que permet a una classe tindre únicament una instància que pugui ser accessible globalment. [37]. Aquest patró s'ha usat en combinació amb el següent per a la creació de les factories de CKBull Developer Console (*ControllerFactory* i *RepositoryFactory*).
- **Factoría simple**: Patró que usa una classe per a la creació d'objectes d'altres classes. Utilitzada, com s'ha descrit al patró anterior, per a les factories dels controladors de domini i els repositoris de CKBull Developer Console [38].

En aquesta secció únicament s'han descrit els patrons usats en el disseny del sistema. Posteriorment al Capítol 11 s'afegiran nous patrons segons la tecnologia que es faci servir per a desenvolupar el sistema.

# Capítol 11

## Implementació

Un cop definit el disseny del sistema, amb els seus components i les seves interaccions, és hora d'implementar la solució. Aquest capítol es divideix en tres seccions principals, on cadascuna descriu la implementació de cada component del sistema.

### 11.1 Tecnologies comunes

Per a afavorir una implementació àgil acord amb les tecnologies utilitzades per Peersyst, s'han definit unes tecnologies comunes entre les tres implementacions. D'aquesta manera, el desenvolupament del projecte no es veurà afectat en gran mesura pel risc **Limitacions tècniques** definit al Capítol 7. A continuació, es llisten les tecnologies comunes entre les tres solucions amb una breu descripció.

#### Javascript

Llenguatge de programació interpretat fet servir principalment per al *scripting* en pàgines web. Durant els anys s'ha estès més enllà del navegador i, gràcies a altres tecnologies com Node [39], es pot usar per a altres usos, com desenvolupament de servidors [40].

#### TypeScript

Llenguatge de programació tipat i compilat construït sobre Javascript. Entre els seus usos es troba la inferència de tipus i la creació de tipus personalitzats. S'utilitza al projecte per mantenir una sintàxi llegible i evitar el nombre més gran d'errors possibles (objectiu alineat amb evitar el risc **Aparició d'errors i bugs** [41]).

#### Yarn

Gestor de paquets utilitzat per a gestionar les dependències dels projectes. És compatible amb els llenguatges esmentats i s'usa a totes les implementacions. A més, és fet servir per Peersyst i, per tant, ajuda a mantenir la coherència tecnològica entre projectes [42].

#### Docker

Docker és un programari de codi obert que permet, entre altres funcionalitats, desplegar projectes de manera automàtica incloent tots els recursos necessaris abstraient la capa física [43]. Aquesta tecnologia ha sigut utilitzada per tots els components del sistema per agilitzar el desplegament de les solucions i poder emular aquests durant el desenvolupament.

## 11.2 CKBull Developer Console

Dintre de les múltiples possibilitats que existeixen per a construir CKBull Developer Console, s'ha escollit la solució d'implementar una **aplicació web**. Els principals factors que han determinat aquesta decisió han estat el **coneixement de la tecnologia** a utilitzar, la **complexitat de la solució i satisfer els requisits funcionals i funcionals del sistema**.

Gràcies a realitzar assignatures durant el grau relacionades amb el desenvolupament web, com Arquitectura i Serveis Web, i l'aprenentatge continu durant el desenvolupament del projecte, he disposat de coneixement per a fer una solució que s'adapti als requisits del problema.

### 11.2.1 Tecnologies

CKBull Developer Console s'ha desenvolupat com una aplicació web *client-side rendering*<sup>1</sup> amb les tecnologies que s'usen a Peersyst per al desenvolupament.

#### React

Llibreria de Javascript feta servir per a la construcció dels components de la interfície web. React permet tant el desenvolupament d'interfícies web i natives i és pensada per segmentar la interfície en components, que poden encapsular estat i lògica. A més, a diferència del *server-side rendering* permet actualitzar els components de la interfície sense refrescar la pàgina [44].

#### React components (Peersyst)

Llibreria de components per React desenvolupada per Peersyst. Inclou diferents components d'interfície que permeten un desenvolupament més àgil. En tractar-se d'una llibreria creada per Peersyst, durant el desenvolupament del projecte s'han afegit altres elements necessaris per a la construcció de les solucions [45].

#### React Router

Llibreria per utilitzar la navegació *client-side routing* amb React. Aquest tipus de navegació permet poder renderitzar components a la interfície sense demanar al servidor la informació. S'ha fet servir per a tota la navegació de l'aplicació [46].

#### React Query

Llibreria que permet realitzar peticions de dades a altres serveis, abstraient la implementació de la lògica i l'estat de la petició. Fent ús aquesta llibreria s'ha aconseguit gestionar de manera eficient l'estat de les peticions i els errors [47].

#### Styled components

Llibreria que permet estilitzar components d'una interfície amb CSS. S'ha usat per a modificar i maquetar els components de la interfície d'acord amb els dissenys proporcionats per Urano [48].

#### Zustand

Utilitat lleugera que permet gestionar els estats d'un software dels *frameworks* utilitzats. En aquest projecte, s'ha fet servir per a gestionar estats globals del sistema com les credencials al *playground* i informació del desenvolupador [49].

---

<sup>1</sup>Permet desenvolupar aplicacions web que renderitzen completament al navegador.

## **i18next**

Un dels requisits no funcionals del projecte és la **internacionalització**, que correspon a proporcionar als usuaris les funcionalitats en diferents idiomes, en aquest cas castellà i anglès. La llibreria i18next permet afegir múltiples idiomes a l'aplicació [50].

## **11.3 CKBull Signer API i base de dades**

Ja que les interfícies amb la que es comunicarà CKBull Signer API són aplicacions web i natives, l'opció més encertada per a un desenvolupament àgil de l'API és la construcció d'aquesta sobre el protocol HTTP. A més, és un protocol amb el qual s'han dut a terme projectes anteriorment a Peersyst i també durant algunes assignatures al grau.

### **11.3.1 Tecnologies de l'API**

Peersyst també disposa d'un conjunt de tecnologies (*stack*) per a la producció d'APIs sobre HTTP. Les tecnologies utilitzades per aquest *stack* són les descrites a continuació.

#### **NestJS**

*Framework open-source* de Node.js que permet desenvolupar de manera progressiva aplicacions sobre servidors. El seu desenvolupament és basat en mòduls i la injecció de dependències, permetent així un desenvolupament a mesura, segons les funcionalitats a implementar, i eficient. Usa per sota la llibrería *express*, un altre *framework* de Node.js per a construir aplicacions de servidors [51].

#### **typeorm**

ORM (Object-relational mapping) multiplataforma que permet abstraure les crides a base de dades. Es pot fer servir amb Javascript i Typescript. S'ha usat a la capa de repositoris, especificada a la Figura 42, per facilitar les crides a la base de dades i gestionar les entitats[52].

#### **class-validator**

Llibreria de decoradors per a la validació d'atributs a classes. S'ha fet servir principalment per a la validació d'objectes a les crides de l'API, models de la base de dades i DTOs [53].

#### **passport amb jwt**

Llibreria que ofereix *middleware* d'autenticació utilitzada sobre Node.js. Permet fer servir diferents estratègies d'autentificació. En aquest cas, s'ha fet servir l'estratègia amb JWT (json-web-tokens) per a l'autenticació dels desenvolupadors mitjançant CKBull Developer Console [54].

#### **ckb-peersyst-sdk**

Mateix SDK fet servir a CKBull Wallet per a agilitzar la integració de crides de CKB. Únicament utilitzat per a la producció del *playground* per a la generació de transaccions parcials de criptomonedes natives i NFTs.

### 11.3.2 Tecnologies de la base de dades

Com s'ha definit a l'apartat **Disseny de la base de dades**, s'ha optat per a l'ús d'una base de dades relacional. Dintre d'aquest tipus trobem moltes opcions disponibles com ara MySQL, MariaDB, Oracle... No obstant això, la base de dades que s'ha utilitzat ha sigut **PostgreSQL**, pel fet que s'ha utilitzat anteriorment a altres projectes a Peersyst. Gràcies a assignatures com Base de dades (BD) i Disseny de Bases de dades (DBD) no s'han trobat impediments, com dificultat d'aprenentatge de la tecnologia, durant el desenvolupament.

### 11.3.3 Seguretat

Com ja s'ha especificat anteriorment existeixen diferents tipus d'actors que interactuen amb l'API: els desenvolupadors de dApps (que interactuen mitjançant l'ús de CKBull Developer Console) i la dApp creada pel desenvolupador (que interactua mitjançant les credencials API). Per a poder asegurar que els actors que es connecten a l'API són qui representen, s'han aplicat diferents tipus d'autenticació.

#### Autenticació per a desenvolupadors de dApps

Ja que la interacció dels desenvolupadors amb l'API es realitza mitjançant la interfície CKBull Developer Console, l'autenticació implementada es basa en *JSON Web Tokens*. Els **JSON Web Tokens** corresponen a l'estàndard RFC 7519 [55] i la seva funció és transmetre informació de manera segura entre dues entitats mitjançant un objecte JSON.

El procés d'autenticació inicia quan el desenvolupador vol identificar-se iniciant sessió a CKBull Developer Console, proporcionant les seves credencials. Aquestes són enviades a l'API on es comprova si són correctes i, en cas afirmatiu, es retorna un *string* que contindrà la informació encriptada del JSON Web Token. La informació transmesa pel JSON correspon a:

- **alg**: Algoritme utilitzat per encriptar el JWT.
- **typ**: Tipus de token.
- **email**: Adreça de correu electrònic del desenvolupador que vol identificar-se.
- **id**: Nombre enter identificador de l'usuari.
- **iat**: Data en format *timestamp* de la creació del token.
- **exp**: Data d'expiració del token en format *timestamp*.

Amb aquest identificador retornat a la interfície, el desenvolupador es tornarà validat i podrà realitzar la resta de crides del sistema, com crear, editar i esborrar dApps. Cada crida contindrà el token que es verificarà abans d'actuar sobre l'acció demandada pel desenvolupador. Un cop s'arribi a la data d'expiració, el token romanirà expirat i el desenvolupador haurà de tornar a identificar-se per tornar a fer crides protegides.

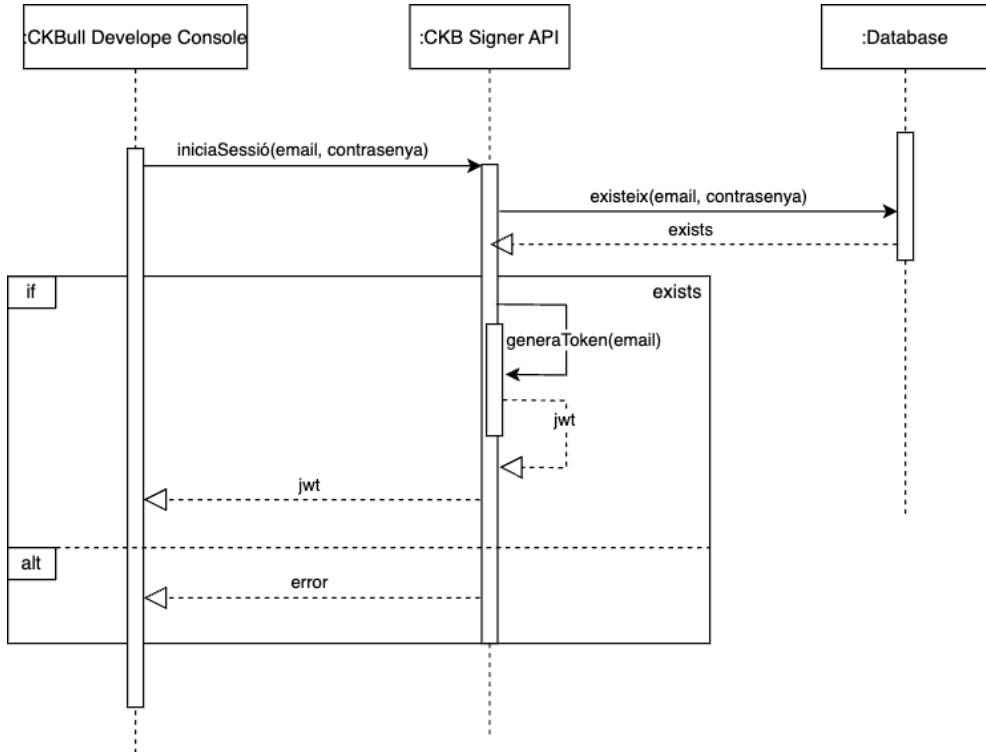


Figura 44: Procés d'autenticació amb JWT.  
Font: elaboració pròpia.

### Autenticació per dApps

Per a validar l'autenticació d'una dApp quan aquesta vol interactuar amb l'API s'utilitza un altre mètode diferent de l'utilitzat amb els desenvolupadors. Com s'ha detallat a l'especificació i al disseny, les dApps, un cop enregistrades pels desenvolupadors a CKBull Developer Console, obtenen unes credencials per a identificar-se. Les credencials corresponen a dues claus API: API Key i ***API secret***. L'ús d'aquestes credencials és semblant a l'ús fet servir a la criptografia de clau pública i clau privada. Per a autenticar els usuaris mitjançant claus API, s'ha implementat l'**algoritme HMAC-SHA512** [56] per a produir missatges signats mitjançant l'***API secret***. El funcionament de l'autenticació segueix les següents passes:

1. dApp: Generar un *timestamp* que actuarà com a *payload* per a signar amb l'*API secret* mitjançant l'algoritme HMAC-SHA512.
2. dApp: Enviar a la capçalera de la petició HTTP l'*API key*, el *timestamp* i la signatura obtinguda.
3. API: Obtenir l'*API secret* de la base de dades mitjançant l'*API key*.
4. API: Executar el primer pas amb el *timestamp* enviat per l'API. En cas que la signatura sigui la mateixa, es verifica la dApp. Altrament es denega la petició.

La figura següent resumeix l'autenticació a les peticions de les dApps amb l'API.

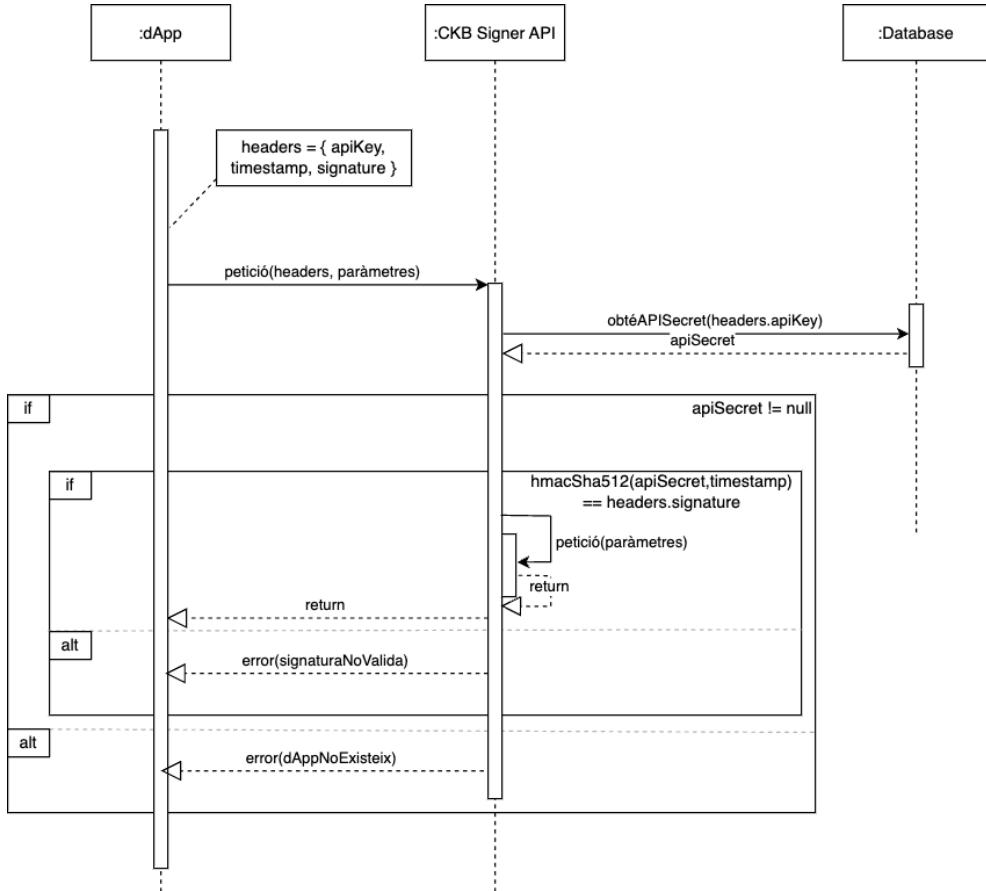


Figura 45: Procés d'autenticació per dApps.  
Font: elaboració pròpia.

La dApp és **totalment responsable de no compartir o exposar l'*API secret*** ja que és l'objecte que permet identificar-lo. En cas de filtració o pèrdua de la credencial, el desenvolupador podrà regenerar l'*API secret* a CKBull Developer Console.

A més d'aplicar processos d'autenticació per garantir la seguretat, també s'han aplicat criptografia per a encriptar informació crítica de la base de dades. Per a veure en detall els motius i com s'ha encriptat aquesta informació es recomana llegir el Capítol 14.

## 11.4 CKBull Wallet

En ser CKBull Wallet un projecte ja en estat de producció, s'ha partit d'una tecnologia i tècniques ja definides. CKBull Wallet és una aplicació mòbil multiplataforma disponible per a *iOS* i *Android*. Per tant, el desenvolupament de les noves funcionalitats encara utilitzarà les següents tecnologies.

### React native

És un *framework* de Javascript usat per a la construcció d'interfícies multiplataforma, popularment utilitzada a *iOS* i *Android*, mitjançant React. Segueix l'esquema *learn once, apply anywhere*, referint-se a la capacitat d'aprendre únicament el framework per a poder produir interfícies a les dues plataformes. A més, és fet servir per Peersyst per al desenvolupament de qualsevol aplicació

nativa [57].

### **Expo**

Plataforma *open-source* que funciona sobre React Native per potenciar el desenvolupament d'aplicacions natives multiplataforma. Permet el desenvolupament mitjançant l'ús de Javascript i Typescript i proporciona als desenvolupadors un servei de construcció d'aplicacions automàtic [58].

### **React components (Peersyst)**

Com a CKBull Developer Console, la llibreria React components inclou també diferents components d'interfície per al seu ús amb React Native. Gran part dels components de React Native comparteixen la mateixa lògica o esquelet amb els components de React fets servir a CKBull Developer Console [45].

### **React Navigation**

React Navigation és una llibreria de React Native que permet gestionar les pantalles d'una aplicació nativa i mantenir un historial de pantalles visitades. El seu ús és pràcticament idèntic al de la llibreria React Router utilitzada a CKBull Developer Console, però utilitzant una estructura de dades diferent per adaptar-ho a l'ecosistema natiu [59].

### **React Query**

Llibreria que permet realitzar peticions de dades a altres serveis, abstraient la implementació de la lògica i l'estat de la petició. Utilitzant aquesta llibreria s'ha aconseguit gestionar de manera eficient l'estat de les peticions i els errors [47].

### **Styled components**

Com s'ha descrit anteriorment, Styled components és una llibreria que permet estilitzar components d'una interfície amb CSS i, en aquest cas, els estils propis de React Native. S'ha fet servir per a modificar i maquetar els components de la interfície d'acord amb els dissenys proporcionats per Urano [48].

### **Recoil**

Llibreria externa que permet gestionar l'estat dels components de la interfície d'una manera senzilla i eficient. Tot i disposar del *context* de React Native, s'ha decidit usar aquesta llibreria a causa de les poques línies de codi que calen per gestionar un estat i l'eficiència de la gestió [60].

### **i18next**

Mateixa llibreria usada a CKBull Developer Console per a afegir multiidioma a l'aplicació. Prèviament al desenvolupament del projecte ja es troava aplicada la funcionalitat amb diferents idiomes [50].

### **ckb-lumos**

Framework *open-source* per a Javascript i Typescript per al desenvolupament de dApps dins de l'ecosistema de Nervos CKB. Inclou crides com enviament de transaccions, creació de comptes, etc. S'ha fet servir per a desenvolupar un SDK explícit al següent punt [61].

### **ckb-peersyst-sdk**

SDK desenvolupat per Peersyst per afavorir la integració de la llibreria ckb-lumos i agilitzar les crides a la *blockchain*. Les operacions de signatura i enviament són exemples de crides realitzades mitjançant el SDK. Durant el procés de desenvolupament del projecte, aquesta llibreria ha sigut

modificada per a poder adaptar certes crides a la solució plantejada.

## 11.5 Técniques i pràctiques

Per a la implementació dels diferents components del sistema s'han seguit una sèrie de tècniques i pràctiques agnòstiques a la tecnologia utilitzada a la implementació. A continuació es llisten les tècniques emprades amb una breu descripció que són i que aporten al desenvolupament.

### 11.5.1 Monorepo

Monorepo correspon al concepte d'**usar un mateix repositori de codi per a emmagatzemar diferents projectes**. Un exemple d'aquest tipus de tècnica correspon a tindre un directori dins del repositori per a cada projecte. Existeixen diversos beneficis i inconvenients en comparació amb Polyrepo, la tècnica contrària, però l'opció el principal avantatge que aconseguim en aquest projecte es dona a la fase del desenvolupament del projecte.

Amb l'estructura monorepo disposem tant de **CKBull Developer Console** com de **CKBull Signer API** dins del mateix repositori, facilitant el desenvolupament en local i el desplegament dels dos projectes. CKBull Wallet queda exclòs d'aquesta estructura pel fet que és un projecte desenvolupat anteriorment i disposa del seu propi repositori. Dins d'aquest projecte CKBull Developer Console i CKBull Signer API es troben dins de la carpeta *packages* sota els aliasses de *frontend* i *backend* respectivament.

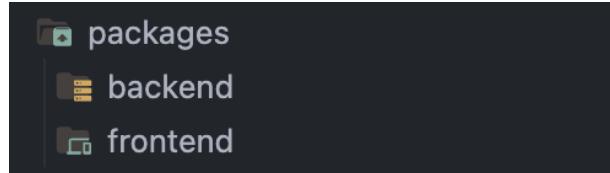
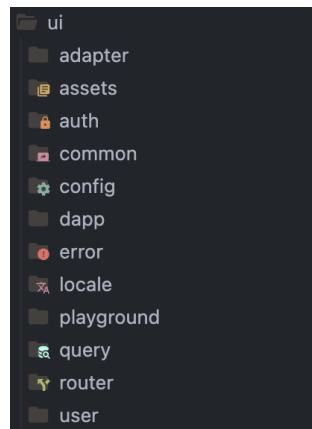


Figura 46: Estructura de carpetes del monorepo.  
Font: elaboració pròpia.

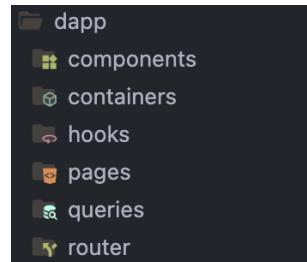
### 11.5.2 Estructura modular

Una altra tècnica que s'ha aplicat al desenvolupament de tots els components del projecte és l'estructura modular. Aquesta tècnica consisteix a encapsular els components, estats, crides i errors entre altres segons la seva funcionalitat. El principal benefici d'aquest d'aquesta tècnica és un acoblament reduït entre els components i una major organització del codi.

Segons el tipus de mòdul i de projecte pot seguir diferents estructures. Per exemple, la figura següent representa un exemple de com s'agrupen els components a CKBull Developer Console.



(a) Estructura de mòdul UI



(b) Estructura de mòdul DApp

Figura 47: Exemples d'estructura modular a CKBull Developer Console.  
Font: elaboració pròpia.

### 11.5.3 Entorns

Durant el desenvolupament del projecte, s'han definit entorns en el que es pot trobar un component del sistema segons la fase a la qual es trobi. En específic, s'han definit els 3 següents:

- **Development** (o desenvolupament): Entorn on es desenvolupa el *software* del projecte. Dins d'aquest entorn es realitza tant desenvolupament com proves del sistema.
- **Staging** (o rèplica de producció): Entorn on es troben totes les funcionalitats desenvolupades a l'entorn *development*. Aquest entorn actua com una rèplica quasi exacta del següent entorn.
- **Production** (o producció): Entorn on es troba la versió més estable del *software*. En aquest entorn interactuen els usuaris directament al *software*.

La relació que s'estableix entre els tres entorns és l'especificada a la següent figura:

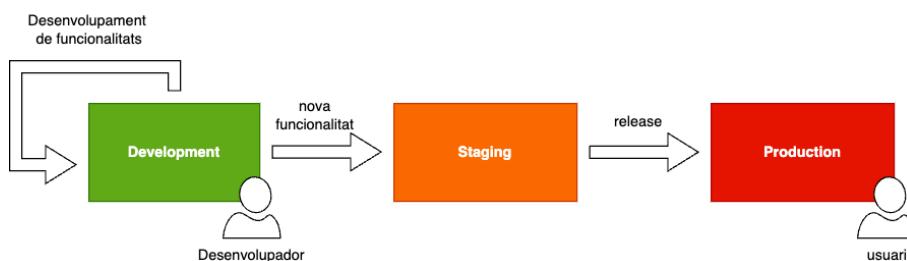


Figura 48: Relació entre entorns.  
Font: elaboració pròpia.

Idealment, cada cop que es desenvolupa una **nova funcionalitat**, aquesta es replica a l'entorn de *staging*. Aquesta replicació no es limita a únicament noves funcionalitats, sinó que s'estén també a *fixes* i solucions a errors o *bugs*. Un cop la funcionalitat ha sigut testejada a l'entorn de

*staging* aquesta pot passar a l'estat de producció mitjançant una *release*<sup>2</sup>, on serà utilitzada pels usuaris del *software*. Respecte a l'ús d'aquests entorns durant la vida del projecte, s'han fet servir des de l'inici els entorns de *development* i l'entorn de *staging*. L'entorn de producció ha variat segons els components. CKBull Wallet ja disposava d'un entorn de producció actiu prèviament al desenvolupament de la funcionalitat. En canvi, CKBull Developer Console i CKBull Signer API van entrar a l'entorn de *production* a la fase final del projecte, un cop estaven fetes totes les funcionalitats, a causa de la dependència de CKBull Signer API sobre la resta de components.

## 11.6 Patrons de disseny

En aquesta secció, s'afegeiran als patrons de disseny definits al Capítol 10 els patrons que s'han utilitzat a la implementació. Seguint la definició, es dividiran segons la seva categoria: creacionals, estructurals i de comportament.

### 11.6.1 Patrons creacionals

- **Singleton:** Patró que permet a una classe tindre únicament una instància que pugui ser accessible globalment. [37]. Aquest patró s'ha utilitzat en combinació amb el següent per a la creació de les factories de CKBull Developer Console (*ControllerFactory* i *RepositoryFactory*).
- **Factoría simple:** Patró que usa una classe per a la creació d'objectes d'altres classes. Utilitzada, com s'ha descrit al patró anterior, per a les factories dels controladors de domini i els repositoris de CKBull Developer Console [38].
- **Data Transfer Object** (o objecte de transmissió de dades): Patró utilitzat per a transferir informació entre processos [62]. A diferència dels objectes de negoci (com les entitats), els DTOs només han d'utilitzar-se per a la transmissió d'informació i no per a emmagatzemar-la. S'ha fet servir per a la transmissió de les entitats, emmagatzemades a base de dades, entre els components del sistema.
- **Dependency Injection** (o injecció de dependències): Patró de disseny en el qual un objecte o funció rep altres objectes dels quals depèn per funcionar [63], sense saber com es construeixen. La injecció de dependències ajuda a mantenir una arquitectura amb poc acoblament. Com a exemple, s'ha utilitzat als controladors de domini de CKBull Developer Console i als components de CKBull Signer API.

### 11.6.2 Patrons estructurals

- **Repository** (o repositori): Patró de disseny centrat a mantenir tota la lògica de persistència de dades agnòstica a la capa de domini [64]. S'utilitza tant a CKBull Developer Console, per a emmagatzemar el JWT del desenvolupador, i a CKBull Signer API, per gestionar l'emmagatzemament d'entitats.
- **Data Mapper** (o mapeig de dades): Patró de disseny semblant a l'anterior, on per cada objecte existeix una interfície que presenta operacions sobre l'objecte [65]. És fet servir per a la comunicació entre l'API i la base de dades per gestionar les taules i les entitats.

---

<sup>2</sup>Release: Treure, estrenar, fer pública una informació.

# Capítol 12

## Tests i validació de requisits

Per a poder garantir que el sistema funciona correctament és necessari aplicar mecanismes de validació. En aquest capítol s'especifiquen les tècniques utilitzades durant el procés de validació i proves.

### 12.1 Tècniques de validació

Prèviament a l'explicació de la validació dels requisits del sistema, és convenient explicar quines tècniques s'han aplicat per proves sobre el *software*, com s'han aplicat i a quins components.

Com s'ha descrit durant el projecte, aquest sistema consta de parts diferents que es comuniquen entre elles. A nivell de desenvolupament, s'han hagut de fer proves sobre els tres components descrits a la implementació: la interfície web CKBull Developer Console, l'aplicació mòbil CKBull Wallet i l'API CKBull Signer API. A causa dels recursos temporals que dels que es disposaven s'ha optat per a fer els següents tipus de proves.

#### 12.1.1 Tests unitaris

Els tests unitaris [66] són proves de *software* que es realitzen sobre una peça individual de codi per a determinar si aquesta funciona correctament. Aquest tipus de tests s'ha utilitzat als 3 components desenvolupats. Un exemple d'un test unitari a la interfície CKBull Developer Console es pot veure representat a la Figura 49. Aquest tipus de proves s'han utilitzat per als components d'interfície de CKBull Developer Console i CKBull Wallet i per als serveis de CKBull Signer API.

```

describe("DAppList", () => {
  Run | Debug
  test("Renders correctly with items", () => {
    const mockApps = [new CompleteDAppDtoMock(), new CompleteDAppDtoMock()];

    render(<DAppList apps={mockApps} />);

    expect(screen.getAllByText("name")).toHaveLength(2);
    expect(screen.getAllByText("description")).toHaveLength(2);
    expect(screen.getAllByRole("img")).toHaveLength(2);
  });

  Run | Debug
  test("Renders correctly with no items", () => {
    render(<DAppList apps={[]} />);

    expect(screen.getByText(translate("youHaveNoApps"))).toBeInTheDocument();
  });
});

```

Figura 49: Test unitari del component DAppList.

Font: elaboració pròpia.

En aquest exemple es mostren dos tests que determinen si el component de la interfície *DAppList* es comporta com s'espera. La funcionalitat d'aquest component és mostrar les dApps que li arriben per paràmetre en format de vector i en cas de ser aquest vector buit mostrar un missatge avisant que l'usuari no té dApps. En el primer test, el component renderitza amb dos dApps i comprova que es visualitza el nom, la descripció i la imatge de la dApp. Altrament, al segon test el component renderitza un vector buit i, a continuació, es comprova amb la funció *expect* que es mostra el missatge esperat. No obstant això, als tests unitaris no es proporciona cap mena de dades reals, sinó que s'utilitza la tècnica següent.

### 12.1.2 Mock objects

Els *mock objects* [67] (o objectes simulats) són objectes que repliquen el comportament i els atributs d'un objecte real o de negoci. Amb aquesta tècnica, garantim que els components que usin aquests objectes simulats funcionin tal com ho farien amb un objecte real. Com es pot veure a la Figura 49 les dApps proporcionades al component *DAppList* són *mocks* del DTO *CompleteDAppDto*.

```

You, 3 months ago | 1 author (You)
export default class CompleteDAppDtoMock implements CompleteDAppDto {
  id: number;
  name: string;
  description: string;
  email: string;
  projectUrl: string;
  supportUrl: string;
  apiKey: string;
  apiSecret: string;
  termsUrl?: string;
  privacyPolicyUrl?: string;
  image?: string;
  createdAt: string;
  updatedAt: string;
  constructor({...
})
}

```

Figura 50: Mock de la classe CompleteDAppDto.

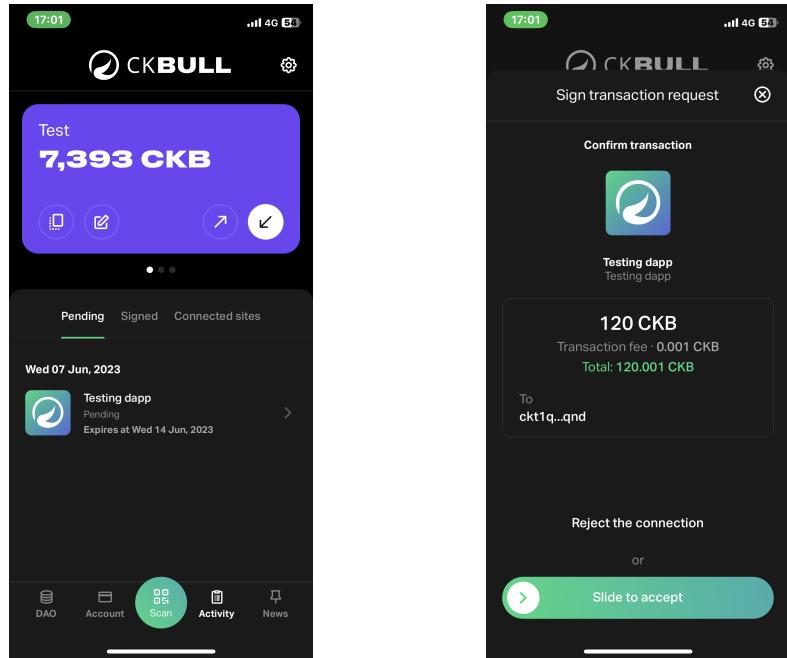
Font: elaboració pròpia.

Per a garantir que l'objecte simulat es comporti igual que l'objecte real, s'utilitzen les interfícies per a garantir que la classe implementa tots els atributs i mètodes. A la Figura 50 es troba l'exemple de com s'implementa un objecte *mock* de la classe *CompleteDAppDtoMock*.

### 12.1.3 Proves no automatitzades end-2-end

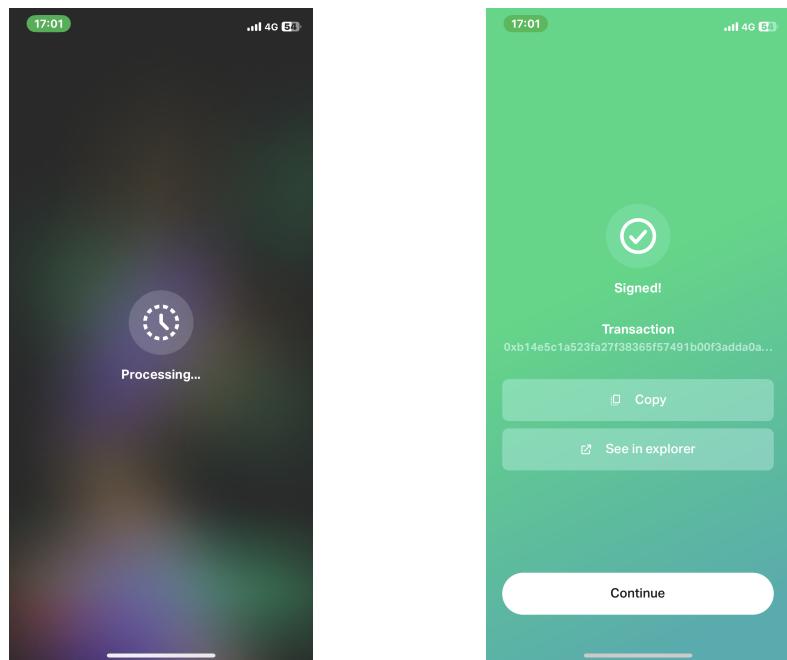
Ja aplicades tècniques de validació a nivell individual de codi, la següent validació que s'ha d'aplicar és en l'àmbit de comunicació entre els components del sistema. Per a poder demostrar que una funcionalitat entre components funciona s'ha fet servir el *testing end-2-end*, que correspon a la **validació de funcionalitats d'inici a fi en un entorn real** (o pràcticament real).

Aquests tipus de proves han estat aplicades per a provar els casos d'ús del sistema que necessiten dos o més components d'aquests. Un exemple correspon a les proves de validació fetes sobre el cas d'ús *Signar transacció*. Aquest cas d'ús involucra CKBull Signer API i CKBull Wallet i el cas d'ús previ *Veure peticions de transacció*. Com que les proves *end-2-end* s'han de realitzar a entorns de proves gairebé reals, **s'ha utilitzat la xarxa *testnet*** de la *blockchain* per a poder provar els casos d'ús abans d'utilitzar la funcionalitat a la xarxa *mainnet*. Les següents figures mostren pas a pas la validació del cas d'ús fent servir tant l'API com l'aplicació mòbil.



(a) Pestanya de transaccions pendents

(b) Detalls de transacció



(c) Pantalla de signatura signant-se

(d) Pantalla d'èxit de signatura

Figura 51: Prova del cas d'ús Signar transacció.  
Font: CKBull Wallet.

## 12.2 Validació de requisits

Un cop definides les tècniques de validació que s'han emprat per a la validació, el següent pas es validar tots els requisits que s'han definit al Capítol 4, tant per als requisits funcionals i els no funcionals. Per a cada requisit, s'ha elaborat una taula on s'explica quin era el resultat esperat, quin resultat s'ha obtingut i quines tècniques de validació s'han aplicat per a validar-lo.

### 12.2.1 Requisits funcionals

Identificador	RF1
Casos d'ús involucrats	Enregistrar-se, iniciar sessió, iniciar sessió amb Google, validar compte
Resultats	
Esperat	Obtingut
Com a desenvolupador m'he de poder enregistrar al sistema amb una adreça de correu personal o amb un compte de Google	Com a desenvolupador puc crear un compte a CKBull Developer Console amb una adreça de correu personal o amb un compte de Google.
Tècniques utilitzades	
Tests unitaris	Components d'interfícies i lAuthService i UserService de l'API.
<i>Mock objects</i>	Replicació de DTOs i serveis
Proves <i>end-2-end</i>	Procés d'enregistrament entre CKBull Developer Console i CKBull Signer API

Taula 58: Validació de RF1.

Identificador	RF2
Casos d'ús involucrats	Veure dApps, crear dApp, editar dApp, generar claus API, eliminar dApp
Resultats	
Esperat	Obtingut
Com a desenvolupador vull poder enregistrar la meva dApp al sistema, editar la seva informació, eliminar-la i veure les dApps enregistrades	Un desenvolupador pot crear múltiples dApps, editar la seva informació, eliminar-les i visualitzar-les, generant també les claus API d'aquesta.
Tècniques utilitzades	
Tests unitaris	Components d'ínterficie i DAppService de l'API.
<i>Mock objects</i>	Replicació de DTOs i serveis
Proves <i>end-2-end</i>	Creació, edició, eliminació i visualització de dApps mitjançant la interfície CKBull Developer Console en connexió a l'API.

Taula 59: Validació de RF2.

<b>Identificador</b>	<b>RF3</b>
Casos d'ús involucrats	Enregistrar-se, iniciar sessió, iniciar sessió amb Google, validar compte
<b>Resultats</b>	
<b>Esperat</b>	<b>Obtingut</b>
Com a desenvolupador no identificat no puc realitzar cap acció que involucri alguna dApp sobre CKBull Developer Console	Sense cap JWT proporcionat per l'API, no es poden accedir a les urls que permeten realitzar accions sobre dApps
<b>Tècniques utilitzades</b>	
Tests unitaris	Components d'interfície i DAppService de l'API.
<i>Mock objects</i>	Replicació de DTOs i serveis
Proves <i>end-2-end</i>	Càrrega de rutes de CKBull Developer Console sense haver-me identificat com a desenvolupador. No s'ha permès cap accés a recursos no autoritzats.

Taula 60: Validació de RF3.

<b>Identificador</b>	<b>RF4</b>
Casos d'ús involucrats	Generar claus API, regenerar clau secreta API
<b>Resultats</b>	
<b>Esperat</b>	<b>Obtingut</b>
Com a desenvolupador identificat, a l'hora d'enregistrar una dApp es creen credencials API per la dApp.	Al crear una dApp mitjançant CKBull Developer Console es retornen unes credencials API per a la dApp. A més, es pot regenerar la clau secreta.
<b>Tècniques utilitzades</b>	
Tests unitaris	Components d'interfície i DAppService de l'API.
<i>Mock objects</i>	Replicació de DTOs i serveis
Proves <i>end-2-end</i>	Mostrar les claus generades a l'API a CK-Bull Developer Console.

Taula 61: Validació de RF4.

<b>Identificador</b>	<b>RF5</b>
Casos d'ús involucrats	Escanejar codi inici de sessió, acceptar inici de sessió, rebutjar inici de sessió
<b>Resultats</b>	
<b>Esperat</b>	<b>Obtingut</b>
Com a usuari de CKBull Wallet he de poder visualitzar, acceptar i rebutjar peticions d'inici de sessió d'una dApp enregistrada.	Amb CKBull Wallet, l'usuari pot escanear la petició d'inici de sessió i, a continuació, acceptar-la o rebutjar-la.
<b>Tècniques utilitzades</b>	
Tests unitaris	Components d'interfície mòbil i SignInRequestService de l'API.
<i>Mock objects</i>	Replicació de DTOs i serveis
Proves <i>end-2-end</i>	Visualitzar, acceptar o rebutjar una petició d'inici de sessió mitjançant CKBull Wallet en connexió amb l'API.

Taula 62: Validació de RF5.

<b>Identificador</b>	<b>RF6</b>
Casos d'ús involucrats	Veure peticions de transacció, signar transacció, rebutjar transacció
<b>Resultats</b>	
<b>Esperat</b>	<b>Obtingut</b>
Com a usuari de CKBull Wallet he de poder visualitzar, acceptar i rebutjar peticions de transacció pendents d'una dApp a la que hagi iniciat sessió prèviament.	Amb CKBull Wallet, l'usuari pot visualitzar peticions pendents de transaccions, signar la transacció o rebutjar-la i visualitzar el procés de signatura amb el resultat final.
<b>Tècniques utilitzades</b>	
Tests unitaris	Components d'interfície mòbil i TransactionRequestService de l'API.
<i>Mock objects</i>	Replicació de DTOs i serveis
Proves <i>end-2-end</i>	Visualitzar, acceptar o rebutjar una petició de transacció mitjançant CKBull Wallet en connexió amb l'API i la <i>blockchain</i> .

Taula 63: Validació de RF6.

<b>Identificador</b>	<b>RF7</b>
Casos d'ús involucrats	Consultar petició de transacció
<b>Resultats</b>	
<b>Esperat</b>	<b>Obtingut</b>
Com a dApp enregistrada, vull poder consultar al sistema l'estat d'una petició de transacció creada per mí.	Utilitzant l' <i>endpoint</i> de CKBull Signer API i un <i>transactionToken</i> es pot demanar l'estat d'una petició de transacció.
<b>Tècniques utilitzades</b>	
Tests unitaris	TransactionRequestService de l'API.
<i>Mock objects</i>	Replicació de DTOs i serveis

Taula 64: Validació de RF7.

<b>Identificador</b>	<b>RF8</b>
Casos d'ús involucrats	Consultar petició d'inici de sessió
<b>Resultats</b>	
<b>Esperat</b>	<b>Obtingut</b>
Com a dApp enregistrada, vull poder consultar al sistema l'estat d'una petició d'inici de sessió creada per mí.	Utilitzant l' <i>endpoint</i> de CKBull Signer API i un <i>transactionToken</i> es pot demanar l'estat d'una petició d'inici de sessió.
<b>Tècniques utilitzades</b>	
Tests unitaris	SignInRequestService de l'API.
<i>Mock objects</i>	Replicació de DTOs i serveis

Taula 65: Validació de RF8.

<b>Identificador</b>	<b>RF9</b>
<b>Resultats</b>	
<b>Esperat</b>	<b>Obtingut</b>
Com hi ha desenvolupador de dApps he de disposar de documentació que permeti integrar-me amb el sistema.	El desenvolupador disposa de documentació amb la pàgina creada amb <i>Gitbook</i> , on s'especifica com utilitzar la interfície CKBull Developer Console i com utilitzar CKBull Signer API amb les credencials API.

Taula 66: Validació de RF9.

### 12.2.2 Requisits no funcionals

Per finalitzar la validació, a continuació es descriu per a cada requisit no funcional quin és el criteri de validació i com s'ha resolt.

<b>Identificador</b>	<b>RNF1 - Aparença</b>
<b>Criteri de validació</b>	
Les interfícies del sistema han de tindre una aparença semblant amb altres productes de Nervos.	
<b>Resol·lució</b>	Els dissenys han estat creats amb les paletes de color proposades per Nervos. Així doncs, aquests dissenys s'han maquetat amb l'exactitud més gran possible, mostrant tots els colors acordats amb el sistema.

Taula 67: Validació de RNF1.

<b>Identificador</b>	<b>RNF2 - Usabilitat</b>
<b>Criteri de validació</b>	
El sistema ha d'evitar que l'usuari produueixi errors durant l'ús d'aquest.	
<b>Resol·lució</b>	Per a garantir aquest requisit s'ha de segmentar el concepte d'usuari i el desenvolupador de dApp, ja que cadascun realitza diferents accions. Respecte a l'usuari de CKBull Wallet, aquest té limitades les seves accions sobre el sistema pel fet que només pot fer signatures o rebutjaments de transaccions mitjançant l'aplicació mòbil. Altrament, el desenvolupador de dApps té un major rang d'accions que pot realitzar. No obstant això, el sistema limita únicament les accions a dur a terme del desenvolupador gràcies a l'autenticació i ajuda a resoldre confusions mitjançant la documentació generada. A més, els dos usuaris disposen d'una interfície senzilla i guiada per a evitar confusions d'accions.

Taula 68: Validació de RNF2.

<b>Identificador</b>	<b>RNF3 - Internacionalització</b>
<b>Criteri de validació</b>	
El sistema ha de disposar de suport d'idioma d'anglès i castellà.	
<b>Resol·lució</b>	Durant el desenvolupament, s'han aplicat traduccions per als dos idiomes específics a totes les interfícies del sistema. En el cas de la interfície web CKBull Developer Console, l'idioma utilitzat serà el que detecti del mateix navegador (per defecte anglès). Altrament, la interfície CKBull Wallet disposa de diferents idiomes (inclosos anglès i castellà) que poden aplicar-se dins de l'aplicació.

Taula 69: Validació de RNF3.

<b>Identificador</b>	<b>RNF4 - Fiabilitat i disponibilitat</b>
<b>Criteri de validació</b>	
El sistema ha de ser usable el 90% del temps que estigui actiu.	
<b>Resol·lució</b>	Ja que CKBull Wallet és una aplicació nativa i es trobarà disponible sempre que es trobi instalada a un dispositiu, els components que s'han de posar a prova en aquest requisit són CKBull Developer Console i CKBull Signer API. Per a fer-ho s'han realitzat crides a la web i a un <i>endpoint</i> de l'API durant 24 hores en períodes d'1 minut.

Taula 70: Validació de RNF4.

Identificador	RNF5 - Accés
Criteri de validació	
El sistema només ha de permetre l'accés a usuaris mitjançant credencials vàlides.	
Resolució	
La resolució d'aquest requisit ha sigut especificada prèviament al Capítol 11 on s'han especificat quins mètodes d'autenticació s'han aplicat al sistema.	

Taula 71: Validació de RNF5.

Identificador	RNF6 - Privacitat
Criteri de validació	
El sistema ha de complir els requisits necessaris detectats al RGPD.	
Resolució	
La resolució d'aquest requisit ha sigut especificada prèviament al Capítol 14 on s'han detectat quines pràctiques s'han d'implementar al projecte.	

Taula 72: Validació de RNF6.

### 12.3 Resum de validació

Per finalitzar el capítol, a continuació es fa resum de totes les accions de validació aplicades al projecte. Respecte a CKBull Developer Console s'han realitzat 85 test unitaris, sobre CKBull Signer API s'han realitzat 78 tests unitaris per als serveis de l'API. En canvi, a CKBull Wallet existeixen 275 test unitaris, ja que ja existien test prèviament a l'inici del projecte, però s'han fet tests unitaris per a cada component.

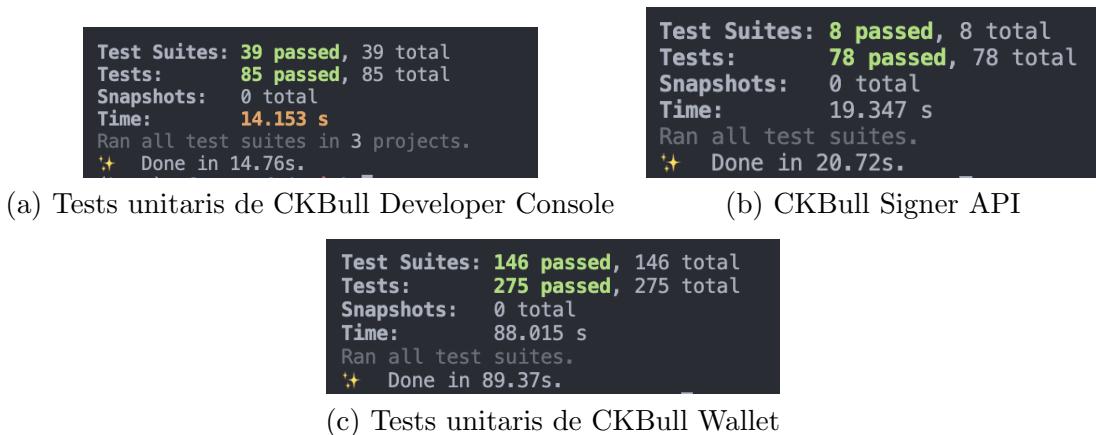


Figura 52: Tests unitaris realitzats sobre els components del sistema.

Font: elaboració pròpria.

## Capítol 13

# Desplegament i documentació

L'última fase del projecte correspon al desplegament dels components del sistema a l'entorn de producció. A més d'aquest procediment, s'ha generat una documentació addicional per als desenvolupadors de dApps per a facilitar als desenvolupadors la integració de les seves dApps amb el sistema.

### 13.1 Desplegament

La primera passa de la fase final és desplegar a l'entorn de producció tots els components del sistema. No obstant això, no tots els components s'han de desplegar amb el mateix procediment. A continuació, es descriu com han sigut els dos procediments diferents de desplegament.

#### 13.1.1 Desplegament de CKBull Developer Console i CKBull Signer API

Tant CKBull Developer Console com CKBull Signer API són components que s'han d'allotjar a un servidor per a poder accedir als recursos d'aquests mitjançant el protocol HTTP. Per aconseguir-ho, s'ha fet un desplegament al servidor FS3, definit a la Secció 6.2, on s'allotjaran tant CKBull Developer Console, CKBull Signer API i la base de dades utilitzada per emmagatzemar la informació del sistema.

Com la resta de projectes *web* i APIs HTTP desenvolupats a Peersyst, aquests projectes s'allotjaran dins d'una **instància EC2** de l'empresa *cloud* **AWS**. Tot i ser dos components a desplegar, la complexitat del desplegament es veu reduïda degut a l'ús de Docker durant el procés de desenvolupament i a l'entorn de *staging*. D'aquesta manera, podem pràcticament garantir que si el projecte funciona a l'entorn de *staging* funcionarà al de producció.

Així doncs, les passes que s'han seguit per a posar en producció els dos projectes corresponen als següents:

1. Clonar el repositori de codi al servidor FS3.
2. Aplicar variables d'entorn de producció (variables que s'hagin de protegir per no ser exposades).
3. Construir els contenidors de les imatges de CKBull Developer Console i CKBull Signer API.

### 13.1.2 Desplegament de CKBull Wallet

Respecte al desplegament d'aplicacions mòbils, aquest s'ha realitzat mitjançant les eines proporcionades d'Expo per a fer les construccions de les aplicacions i la publicació d'aquestes a l'App Store (iOS) i Play Store (Android).

Per als projectes d'aplicacions mòbils, hi ha configurat el desplegament continuo mitjançant el recurs Github Actions. Quan la branca *dev* del projecte fa *merge* amb la branca principal (*main*), Github Actions té programat que faci un desplegament directe cap a Expo. La Figura 53 mostra un exemple de desplegament a producció de l'aplicació per als dos sistemes operatius.

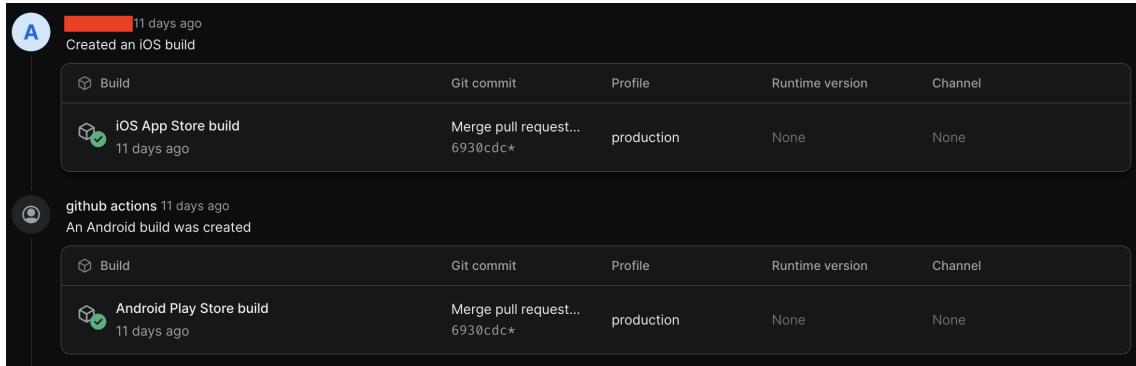


Figura 53: Exemple desplegament gestionat desde Expo.

Font: Pàgina interna de Peersyst d'Expo.

## 13.2 Documentació

L'última secció del projecte consisteix en la documentació. La producció de CKBull Console Developer, CKBull Signer API i CKBull Wallet no obtindrà sentit si no s'adoprt com a solució. Per això, cal fer especial èmfasi en reduir la resistència tecnològica perquè els desenvolupadors de dApps puguin integrar el nou sistema amb les seves aplicacions.

Per afavorir aquest procés, s'ha generat una documentació que explica pas a pas com integrar el nou sistema a qualsevol dApp. Aquesta documentació s'ha generat amb el *software* Gitbook [68], una plataforma per a generar documentació en format web. La documentació es troba en anglès pel fet que és la llengua més utilitzada arreu del món. A mode de resum, els punts que conté la documentació són els següents:

1. **Getting started** (o introducció): Secció on es guia al desenvolupador de dApps pas a pas com crear un compte a CKBull Developer Console, com enregistrar la primera dApp i obtenir i regenerar credencials API.
2. **Guides and concepts** (o guies i conceptes): Secció on s'introduceix al desenvolupador sobre els conceptes més importants de la plataforma. Entre aquests punts es troben conceptes com les *sign in request* (petició d'inici de sessió) i *transaction request* (petició de transacció).
3. **API**: Documentació relacionada amb les crides que pot realitzar la dApp amb les credencials generades amb CKBull Developer Console. Entre elles s'inclou com implementar l'autenticació a les peticions HTTP o les crides *polling* per a comprovar l'estat de les peticions.

La documentació es troba sota el nom de *CKBull Signer API* i es troba disponible en aquesta web.

# Capítol 14

## Lleis i regulacions

### 14.1 Identificació de lleis

Segons el **REGLAMENTO (UE) 2016/679 DEL PARLAMENTO EUROPEO Y DEL CONSEJO de 27 de abril de 2016 relativo a la protección de las personas físicas en lo que respecta al tratamiento de datos personales y a la libre circulación de estos datos y por el que se deroga la Directiva 95/46/CE (Reglamento general de protección de datos)** la definició dels **datos personales** es la següent:

«datos personales»: toda información sobre una persona física identificada o identifiable («el interesado»); se considerará persona física identifiable toda persona cuya identidad pueda determinarse, directa o indirectamente, en particular mediante un identificador, como por ejemplo un nombre, un número de identificación, datos de localización, un identificador en línea o uno o varios elementos propios de la identidad física, fisiológica, genética, psíquica, económica, cultural o social de dicha persona;

Dins de la implementació de la nostra solució trobem diferents tipus de dades personals que poden mantenir una similitud amb la definició de dades personals. Aquestes dades únicament fan referència a les aportades pel desenvolupador de dApps, que són:

- Adreça de correu electrònic.
- Contrasenyes.

Per a poder garantir que aquesta informació personal és gestionada de manera segura, s'han aplicat diferents tipus d'encriptació perquè, en cas de sostracció de les dades, aquestes no puguin ser accessibles. Els tipus d'encriptació que s'ha aplicat varia segons l'estat de les dades. Els estats en el que es poden trobar les dades dins de la solució corresponen a l'**estat de trànsit**, on les dades són enviades d'un servei a un altre, i l'**estat de repòs**, on aquestes dades són emmagatzemades.

Per a les dades en estat de trànsit s'aplica la **Transport Layer Security (TLS)**, un **protocol criptogràfic** sobre la capa de transport del model OSI, per a **garantir l'autenticació i la privacitat de la informació compartida entre extrems** [69]. Aquest protocol és utilitzat pel protocol d'aplicació **HTTPS**, protocol sobre el qual funciona la CKBull Signer API i la comunicació amb les interfícies del sistema, en específic la versió 1.3. Exemples de l'ús d'aquest protocol són la comunicació i transmissió de dades entre el client web CKBull Developer Console i CKBull Signer

API.

Per a les dades en estat de repòs s'ha fet servir l'algoritme criptogràfic **Advanced Encryption Standard** (AES). Les dades que es troben en aquest estat són les emmagatzemades a la base de dades, on es troben tant la adreça de correu electrònic com la contrasenya de l'usuari.

## 14.2 Llicències

Com s'ha pogut observar al Capítol 11, el desenvolupament de la solució ha fet servir altres components *software*, com l'ús de llibreries i eines. Gairebé tots els elements són eines *open-source*<sup>1</sup> que permeten tant el seu ús, com el desenvolupament, sota diferents condicions. Aquestes condicions venen determinades per la **llicència** sobre la que es trobi el *software*.

La llicència més freqüent entre els repositoris i eines de codi obert és la llicència **MIT**[70], una llicència que permet l'ús, còpia, modificació i distribució del *software*, així com la no garantia que el programari funcioni com es descriu, fent absents de tota responsabilitat als autors.

No obstant això, no tot el programari fet servir durant el desenvolupament ha sigut *open-source*. Com es van definir als recursos digitals emprats al projecte, **certs programaris** com ara l'entorn de desenvolupament integrat IntelliJ es troba **sobre la seva pròpia llicència**, com també es troba l'ecosistema Github, usat per a la planificació i per a l'ús d'integració contínua.

Per finalitzar, l'objectiu d'aquest projecte és, un cop sigui estable, que passi a ser un software *open-source*, fent així que la comunitat de desenvolupadors de CKB puguin millorar i fer servir-ho per a créixer de manera orgànica l'adopció a la *blockchain* CKB.

---

<sup>1</sup>Open-source: Programari que pot ser estudiat, modificat i usat de manera lliure.

## Capítol 15

# Resultats de desenvolupament

Ja tancada la fase de desenvolupament, és moment per analitzar i extreure conclusions sobre el desenvolupament del projecte. En aquest capítol s'identificaran els esdeveniments que han succeït durant la construcció del projecte, com s'ha respost davant d'aquest i quin impacte han suposat.

### 15.1 Desviacions sobre el pla inicial

El pla inicial consistia en la divisió temporal del projecte en un nombre total de 6 *sprints* de dues setmanes cadascun, conformant la següent divisió:

- **Sprint 1:** del 23 de gener al 3 de febrer.
- **Sprint 2:** del 6 de febrer al 17 de febrer.
- **Sprint 3:** del 20 de febrer al 3 de març.
- **Sprint 4:** del 6 de març al 17 de març.
- **Sprint 5:** del 20 de març al 31 de març.
- **Sprint 6:** del 3 d'abril al 14 d'abril.

Aquesta planificació venia donada per les dates especificades al contracte amb Nervos, tot i que s'han acabat estenen per diversos factors. Així doncs, el període de desenvolupament iniciaria el dia 23 de gener i finalitzaria el 14 d'abril. No obstant això, a continuació es visualitzen les desviacions patides, el seu tipus i en quin *sprint* van succeir.

Sprint	Desviació	Tipus
<i>Sprint 1</i>	No	-
<i>Sprint 2</i>	No	-
<i>Sprint 3</i>	No	-
<i>Sprint 4</i>	No	-
<i>Sprint 5</i>	Sí	Manca o error a la definició de les tasques.
<i>Sprint 6</i>	Sí	Nova tasca

Taula 73: Desviacions per *sprints*.  
Font: elaboració pròpria.

Com es pot observar a la Taula 73, el ritme de desenvolupament durant els primers *sprints* va ser l'esperat, mentre que als dos últims *sprints* van succeir 2 esdeveniments que van provocar certa desviació sobre la planificació.

El primer tipus de desviació que es va patir va ser classificada com a risc prèviament a la planificació inicial com a **Manca o error a la definició de les tasques**, sent el risc amb més probabilitats de succeir durant el procés de desenvolupament. La causa d'aquesta desviació és deguda al desconeixement de l'existència prèvia de conceptes a la tasca amb codi **DCW6.3**, corresponent a **Obtenir, acceptar o declinar peticions de transaccions**. La planificació original no contemplava quins tipus de transaccions serien suportades (a part de la transacció base de tokens natius), arribant a la tasca mencionada anteriorment amb certa incertesa sobre quins tipus de transaccions hauríem de suportar. Per a mitigar l'error, es van detectar 4 tipus de transaccions que podrien ser suportades dins del sistema:

- Tokens natius (CKB).
- NFTs.
- Tokens.
- Transaccions DAO.

de les quals es van decidir implementar els tres primers a causa dels recursos temporals i humans disponibles en el moment de la decisió. Aquesta implementació va suposar una desviació temporal d'un 50% aproximadament sobre el temps de la tasca (un 20% menys de l'esperat segons la planificació inicial), passant de 24 hores de dedicació a un total de 36. Les conseqüències finals no només van comportar **12 hores més de desenvolupament**, sinó el desplaçament de les tasques posteriors a un sisè i setè *sprint* (aquest últim d'una setmana), deixant la divisió per *sprints* de la següent forma.

- **Sprint 1:** del 23 de gener al 3 de febrer.
- **Sprint 2:** del 6 de febrer al 17 de febrer.
- **Sprint 3:** del 20 de febrer al 3 de març.
- **Sprint 4:** del 6 de març al 17 de març.
- **Sprint 5:** del 20 de març al 31 de març.
- **Sprint 6:** del 3 d'abril al 14 d'abril.
- **Sprint 7:** del 17 d'abril al 21 d'abril.

El segon tipus d'esdeveniment que va succeir no va ser cap risc, sinó la creació d'una nova tasca a causa de una idea que va sorgir durant el desenvolupament. Com s'ha descrit durant la memòria, aquest projecte busca poder millorar l'experiència dels usuaris a l'hora d'interactuar amb les dApps de CKB, en específic amb les transaccions entre els dos actors. No obstant això, tot havent produït la plataforma amb totes les interfícies, l'API i la documentació, va sorgir la idea de crear un **nou cas d'ús** per als desenvolupadors que permetés **provar el funcionament sencer de la dApp dins la plataforma sense que aquesta hagués d'implementar totes les crides**. Amb aquesta idea es va materialitzar *playground* o *entorn de proves* que permetés al desenvolupador provar tots els

casos d'ús que impliquessin la seva dApp i ell mateix, sense la necessitat d'integrar la plataforma a la seva dApp amb les credencials API.

Per aconseguir-ho, la nova tasca anomenada **DCDC6.1 Playground** hauria de generar una vista dins del client CKBull Developer Console on un desenvolupador, proporcionant les credencials API generades a l'hora d'enregistrar una dApp, pogués generar peticions d'inici de sessió i peticions de transacció que poguessin ser signades amb CKBull Wallet. El resultat d'aquesta tasca es mostra a la següent figura:

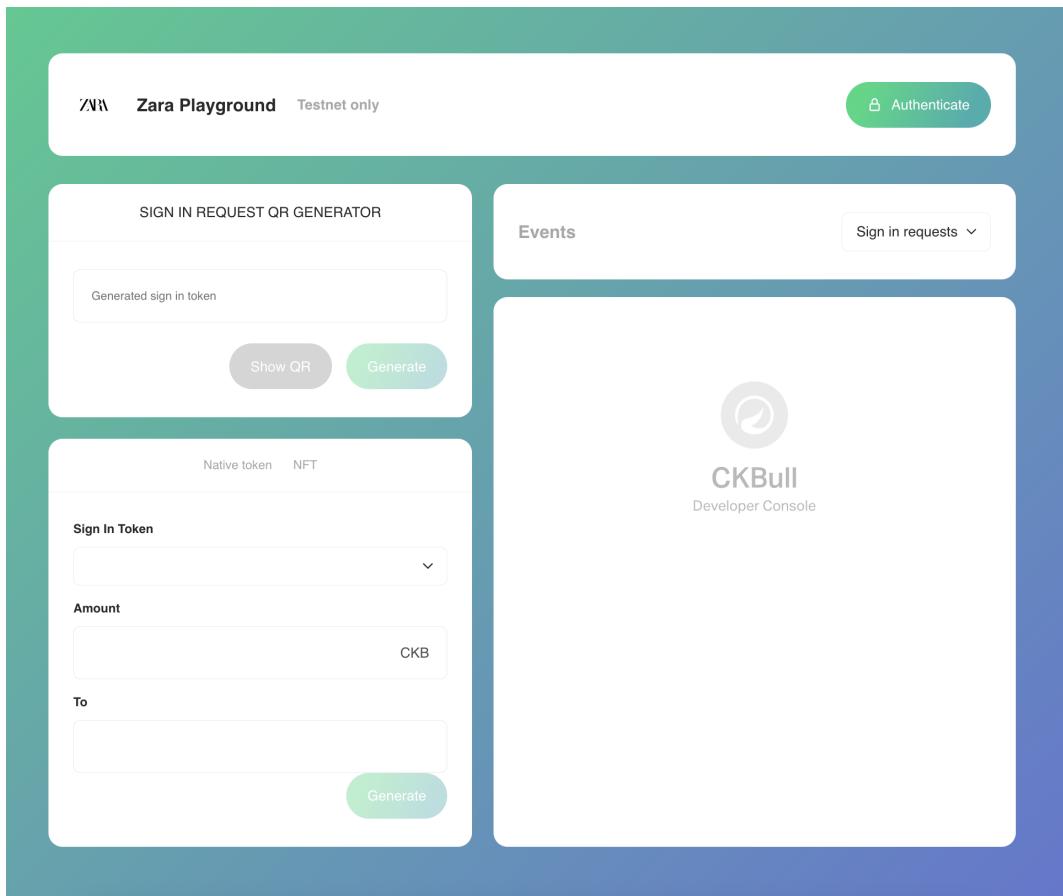


Figura 54: Resultat de la tasca *Playground*.

Font: CKBull Developer Console *playground*.

Amb la interfície mostrada a la figura anterior, un desenvolupador pot testejar com funcionaria la seva dApp. Podria **generar peticions d'inici de sessió** (mitjançant el formulari a la part superior esquerra) que es mostrarien en forma de QR per a després ser escanejades mitjançant la cartera i **generar noves peticions de transacció** (part inferior esquerra). A més, també pot **consultar l'estat de les peticions** mitjançant les llistes amb desplegables de la part dreta de la interfície, on pot llistar tant peticions d'inici de sessió com peticions de transacció. Les següents figures mostren les funcionalitats esmentades.

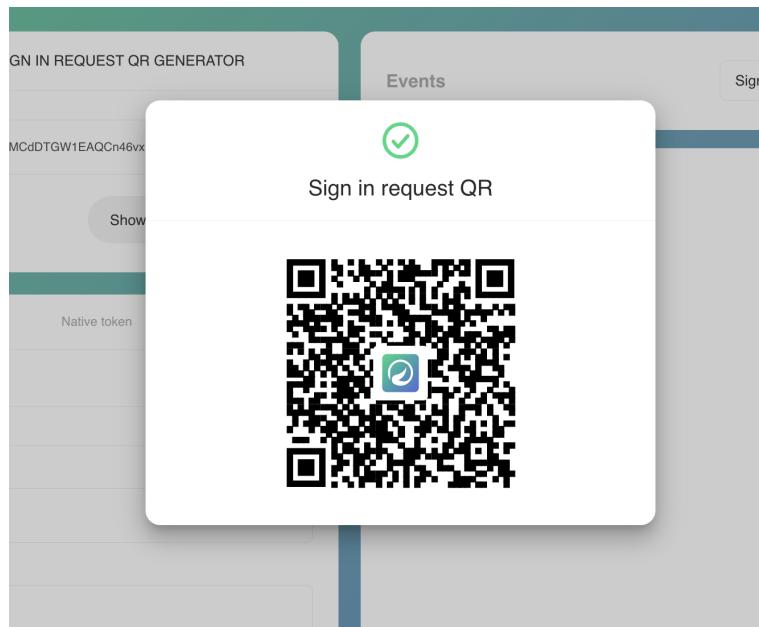
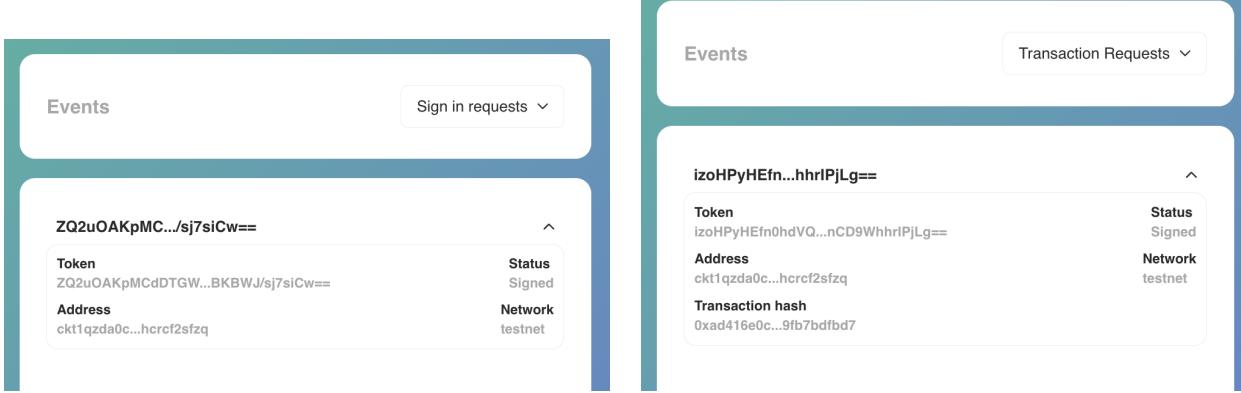


Figura 55: Codi QR de petició d'inici de sessió.  
Font: CKBull Developer Console *playground*.

A screenshot of a mobile application interface titled "Native token NFT". At the top, there are tabs for "Native token" and "NFT". Below the tabs, there is a section labeled "Sign In Token" containing a dropdown menu with the value "ZQ2uOAKpMCdDTGW1...". Below this, there is a section labeled "Amount" with a input field containing "300" and a unit indicator "CKB". Further down, there is a section labeled "To" with an input field containing a long address: "ckt1qzda0cr08m85hc8jlnfp3zer7xulejywt49kt2rr0vthywaa50xwsq". At the bottom right, there is a green button labeled "Generate".

Figura 56: Formulari per crear una petició de transacció  
Font: CKBull Developer Console *playground*.



(a) Detalls d'una SignInRequest

(b) Detalls d'una TransactionRequest

Figura 57: Elements de la llista d'events.  
Font: CKBull Developer Console *playground*.

L'impacte temporal que va suposar la implementació d'aquesta tasca va ser d'un **total de 16 hores**. Aquestes 16 hores inclouen tant el desenvolupament com la validació de la nova funcionalitat, on es van aplicar les tres tècniques descrites a la Secció 12.1. En ser una tasca "gran", aquesta va modificar per segon cop la planificació, fent que el setè *sprint* augmentés la seva durada una setmana més (2 en total), deixant la planificació final amb la següent forma:

- **Sprint 1:** del 23 de gener al 3 de febrer.
- **Sprint 2:** del 6 de febrer al 17 de febrer.
- **Sprint 3:** del 20 de febrer al 3 de març.
- **Sprint 4:** del 6 de març al 17 de març.
- **Sprint 5:** del 20 de març al 31 de març.
- **Sprint 6:** del 3 d'abril al 14 d'abril.
- **Sprint 7:** del 17 d'abril al 28 d'abril.

A continuació es descriurà en detall com aquestes desviacions temporals han afectat la partida pressupostària inicial.

## 15.2 Afectacions sobre el pressupost

En haver-se produït aquesta demora temporal en el desenvolupament, s'han reflectit canvis sobre el pressupost del projecte.

Com el primer esdeveniment descrit a l'apartat anterior va suposar una extensió de 12 hores de desenvolupament per part de l'autor del TFG, la tasca passa de tindre 26 hores de desenvolupament a 38 hores, on 36 hores han sigut dedicades per part del DP (desenvolupador del projecte) i 2 hores per part del PM (*Project Manager*).

No obstant això, al pressupost inicial es van definir 212,76 €dins dels costos associats als riscos que corresponen a 20 hores de desenvolupament per part del DP per a la "Manca o error a la definició". Per tant, el cost general del projecte no es veu modificat respecte al primer esdeveniment pel fet que el temps dedicat sobre el risc patit a l'apartat anterior no ha superat les 12 hores de desenvolupament.

Respecte al segon esdeveniment, la realització de la tasca **DCDC6.1 Playground** va suposar un total de 16 hores per part de l'autor del TFG, suposant una despesa de 283,68 €. Tot i que a l'apartat de riscos no es va definir cap risc relacionat amb la creació de noves tasques, aquesta despesa suposaria un **cost extra**. No obstant això, aquesta quantitat no s'ha d'afegir al pressupost general inicial perquè **no se superen els costos de contingència**. Com aquests costos no s'han utilitzat per al seu propòsit, la despesa d'aquesta nova tasca queda coberta per les despeses de contingència no utilitzades.

# Capítol 16

## Informe de sostenibilitat

Per finalitzar la memòria d'aquest Treball de Fi de Grau, aquest capítol recull l'anàlisi i reflexions relacionades amb la sostenibilitat del projecte en aquestes tres àrees:

- Sostenibilitat ambiental.
- Sostenibilitat econòmica.
- Sostenibilitat social.

### 16.1 Sostenibilitat ambiental

Dintre d'aquest apartat s'analitzarà la sostenibilitat ambiental del projecte en les tres dimensions definides a l'apartat anterior. Cal destacar que a causa de la falta de disponibilitat d'informació alguns dels següents càlculs són estimacions i no costos reals. Els costos ambientals seran estimats en kilowatts hora i la massa emitida de  $CO_2$ .

#### 16.1.1 Posada en producció del projecte (PPP)

Per a quantificar quin cost comporta la posada en producció del projecte cal tindre en compte quins elements i factors són rellevants que són:

1. **Recursos utilitzats en el desenvolupament:** Tots els recursos que poden produir una empremta ecològica utilitzada per al desenvolupament del projecte prèviament a la posada en producció d'aquest. Entre aquests recursos es troben els especificats al Capítol 6. Recursos com els *recursos humans interns* i recursos humans externs, recursos físics hardware i recursos espacials presenten consums d'energia que s'han de tindre en compte.
2. **Recursos del projecte a producció:** Els recursos necessaris per al funcionament correcte del projecte que comportin una empremta mediambiental. Dintre d'aquests recursos únicament destacarem el servidor FS3 especificat a la Secció 13.1 i altres recursos externs com la *blockchain* CKB sobre la que funciona el projecte.

#### Recursos utilitzats en el desenvolupament

Un cop definits quins elements formen part de l'empremta mediambiental del projecte és moment de calcular quin cost comporten. Respecte als recursos físics *hardware* cal esmentar que els dispositius

com el portàtil i els perifèrics (ratolí i teclat) no realitzen un consum continuo sinó per càrregues de bateria. Per a implicar els càlculs, ja que no s'ha portat un seguiment de cada càrrega per dispositiu, s'ha estimat que el nombre de càrregues per a un portàtil és d'una càrrega per cada 10 hores d'ús on el temps de càrrega és de 90 minuts, mentre que la càrrega de la resta de perifèrics es redueix a una càrrega de 30 minuts per cada 25 hores a causa de un consum menor d'energia. Així doncs, el resultat dels costos sobre els recursos hardware físics es calcula a partir de les següents fòrmules.

$$\text{Consum de carrega de portatil} = \frac{67 \text{ W} * 1.5 \text{ hores} * 530 \text{ hores}}{10 \text{ hores} * 1000 \text{ Wh}} = 5,3265 \text{ kWh}$$

$$\text{Consum de carrega de periferics} = \frac{67 \text{ W} * 0.5 \text{ hores} * 530 \text{ hores}}{25 \text{ hores} * 1000 \text{ Wh}} = 0,7102 \text{ kWh}$$

Per a calcular quina quantitat de  $CO_2$  comporten els kilowatts consumits per les càrregues utilitzarem com a referència les dades oferides per la Generalitat de Catalunya a l'article *Factor d'emissió de l'energia elèctrica: el mix elèctric* on el consum de  $CO_2$  per kilowatt hora l'any 2021 és de 259 g $CO_2$ /kWh [71].

Així doncs, el consum en kilograms de  $CO_2$  de les càrregues dels dispositius és la següent.

$$\text{Consum } CO_2 \text{ de periferics} = (5,3265 + 0,7102) \text{ kWh} * 0,259 \text{ kg } CO_2/\text{kWh} = 1,5635 \text{ kg } CO_2$$

Respecte al consum del servidor FS1 utilitzat per a la fase de *staging* el seu consum s'ha calculat mitjançant un estimador de consum de màquines d'AWS. [72]. El resultat és de 0,172kg $CO_2$ . En canvi, per al servidor d'emmagatzemament d'imatges FS2 s'ha neglidit el cost a causa del poc consum que s'ha fet del servei on no s'ha pujat més de 50 imatges durant el desenvolupament.

$$\text{Consum servidor FS1} = 0,172 \text{ kg } CO_2$$

En total, el consum de  $CO_2$  fet per els recursos físics hardware es 1,7355kg $CO_2$ .

Respecte al consum dels recursos humans interns i externs, s'ha de tindre en compte principalment el consum que realitzen els tipus de recursos. En el cas dels recursos humans interns es desconeix el consum en kWh que han sigut necessaris per a fer els dissenys de la UI del sistema. Així que per a proporcionar una estimació, el consum es calcularà de la mateixa manera que el càlcul dels recursos físics del projecte, tenint en compte únicament el portàtil com a recurs principal.

$$\text{Consum de recursos externs} = \frac{67 \text{ W} * 1.5 \text{ hores} * 20 \text{ hores}}{10 \text{ hores} * 1000 \text{ Wh}} = 0,201 \text{ kWh}$$

$$\text{Consum en kg } CO_2 \text{ de recursos externs} = 0,201 \text{ kWh} * 0,259 \text{ kg } CO_2/\text{kwh} = 0,052059 \text{ kg } CO_2$$

Respecte als recursos humans interns, s'han de tindre en compte els recursos espacials definits a Capítol 8 que s'han utilitzat. En aquest cas es disposava del recurs IDT, un espai de *coworking* per a simular el desenvolupament a una oficina. No obstant això, aquest cos es pot veure reflectit al consum fet pels recursos físics com el portàtil i perifèrics, ja que les càrregues s'han fet utilitzant aquest recurs. Així i tot, altres factors com la llum de l'edifici s'han de tindre en compte. Simulant que la llum necessària per a una persona consta d'una bombeta LED [73], el consum en llum del recurs IDT és el següent:

$$\text{Consum del recurs IDT} = 0,013 \text{ kWh} * 530 \text{ hores} = 6,89 \text{ kW}$$

$$Consum\ de\ CO_2\ del\ recurs\ IDT = 6,89\ kW * 0,259\ kgCO_2/kWh = 1,7845\ kgCO_2$$

Així doncs, el consum de  $CO_2$  resultant de l'ús dels recursos de desenvolupament del projecte es calculat per la següent fórmula.

$$Consum\ CO_2\ recursos\ desenvolupament = 6,89\ kW * 0,259\ kgCO_2/kWh = 1,7845\ kgCO_2$$

### Recursos del projecte a producció

Per finalitzar el càlcul del PPP, cal afegir el cost que han comportat la infraestructura de producció. En aquest cas únicament contemplarem el consum mitjà del servidor on es troben allotjades tant CKBull Developer Console com CKBull Signer API. El recurs FS3 és l'utilitzat per allotjar els dos serveis, i tenint en compte que és una instància *t3.large* d'AWS el seu **consum per hora és de 14,1 kWh** produint 6,3g $CO_2$  [74]. Aquest valor no s'inclou en el PPP degut a que el consum d'aquest recurs es produirà durant la vida útil del projecte.

#### 16.1.2 Vida útil

Per determinar la sostenibilitat ambiental del projecte durant la vida útil d'aquest cal tindre en compte quins recursos formen part d'aquest. El recurs principal del projecte i del qual depèn és de la *blockchain* CKB. *Nervos* no ofereix cap informació sobre el consum mitjà d'una transacció de la seva *blockchain*, així que per estimar el cost mitjà d'una transacció s'ha comparat el consum d'una xarxa *blockchain* amb el mateix algoritme de consens (*Proof of Work*) com per exemple Bitcoin. No obstant això, el volum de transaccions de Bitcoin és major que el volum de CKB, amb més nodes minant les transaccions, fent que el cost d'una transacció s'elevi de manera exagerada. Segons *Cambridge Bitcoin Electricity Consumption Index* [75], el consum total de les transaccions de l'any 2022 va ser de 102.51 TWh, amb un nombre de transaccions realitzades de 109,54 milions. No obstant això, per a poder simular un cost que pugui aproximar-se al real, el cost per transacció es multiplicarà pel percentatge de transaccions totals de CKB [76] entre les transaccions totals de Bitcoin durant el mateix any. Així doncs, el cost simulat d'una transacció a CKB suposa:

$$Consum\ d'\ una\ transacció = \frac{102,51\ TWh * 10^9}{109,54 * 10^6\ transaccions} * \frac{8.164.025\ transaccions}{109,54 * 10^6\ transaccions} = 69,7469\ kWh$$

$$Consum\ d'\ una\ transacció\ CO_2 = 69,7469\ kWh * 0,259\ kgCO_2/kWh = 18,0644\ kgCO_2$$

Com es pot observar, tot i ser **una estimació i no una dada oficial**, el cost d'una transacció és molt elevat. Aquest cost equivaldria a un **viatge aproximat de 400 km de distància amb un Tesla Model S 60** [77]. No obstant això, la construcció de la nostra solució aconsegueix no augmentar la de manera dràstica l'empremta ecològica. Al no haver realitzat la solució de manera directa sobre el protocol de CKB, aquesta s'executa únicament sobre un dispositiu (instància del servidor de producció).

Suposant que la solució proposada pot gestionar 40.000 transaccions per hora (unes 11 transaccions

per segon), i que el consum de la instància de producció equival a 6,3gCO<sub>2</sub> el consum de gestionar una petició de transacció amb la solució és el següent:

$$\text{Consum de signatura transacció} = \frac{6,3 \text{ gCO}_2}{40.000 \text{ transaccions}} = 0,0001575 \text{ gCO}_2$$

Si comparem el cost en massa de CO<sub>2</sub> emitida pel nostre projecte en comparació amb el cost d'una transacció de CKB podem observar que el cost d'utilitzar la nostra solució és pràcticament ínfim. Perquè CKBull Signer App arribés a consumir tant com una transacció de CKB, caldrien fer aproximadament 11,42 milions de transaccions, havent-se dut a terme unes 8 milions l'any 2022.

$$\text{Transaccions} = \frac{18,0644 \text{ kgCO}_2 * 10^3}{0,0001575 \text{ gCO}_2} = 11.428.571,42 \text{ transaccions}$$

### 16.1.3 Riscos

Un dels factors que eviten que el projecte augmenti la seva empremta ecològica és la capacitat de mantenir un consum baix respecte a les transaccions realitzades a la *blockchain* CKB. Com s'ha especificat a l'apartat anterior, caldrien fer moltes transaccions mitjançant la plataforma perquè aquesta arribés a consumir tant com ho faria una transacció.

No obstant això, si la plataforma aconsegueix el seu objectiu, que és augmentar l'adopció de CKB, pot generar que augmentin el nombre de transaccions, fent així que es consumeixin més recursos i que les emissions augmentin. Però aquest factor depèn de com actuïn els usuaris. Tot i això, perquè es generi aquesta adopció ha de passar cert temps fins que la plataforma es trobi estable i s'hagin desenvolupat dApps dins de l'ecosistema.

## 16.2 Sostenibilitat econòmica

El següent domini d'anàlisi dins d'aquest informe de sostenibilitat correspon a l'economia relacionada amb el projecte.

### 16.2.1 Posada en producció del projecte (PPP)

Ja que l'estimació pressupostària es va dur a terme de manera detallada i que s'han produït desviacions durant el desenvolupament, no s'ha pogut actuar per a poder reduir els costos del projecte de manera significativa. No obstant això, amb aquest pressupost s'ha intentat **minimitzar reduint la quantitat de recursos al mínim possible**. Un exemple d'aquesta optimització econòmica ha estat utilitzar simuladors de dispositius mòbils en comptes de dispositius mòbils físics, donat que aquests podrien comportar una despesa que pot ser evitada. A més, gran part dels recursos digitals fets servir al desenvolupament són *open-source* o s'ha fet servir amb la subscripció més econòmica.

Tot i haver superat algun cost de contingència la desviació final del projecte no ha sigut rellevant i no ha suposat cap risc real sobre la solució.

### 16.2.2 Vida útil

Un cop finalitzat el projecte i assumits els costos de desenvolupament, és moment de reflexionar sobre el cost que pot comportar aquest durant la seva vida útil. Com s'ha descrit anteriorment, la solució es troba sobre una instància EC2 dins d'AWS. Aquesta instància comporta un cost mensual

aproximat de 120 €. Així doncs, el cost estimat del projecte durant la vida útil és de 120 € al mes. No obstant això, aquesta quantitat pot variar segons la demanda del servei per part dels nous usuaris. En cas que amb la instància actual es produueixi un coll d'ampolla, s'hauria d'estudiar quin factor provoca aquest esdeveniment i, en cas que sigui falta de recursos, s'haurà de seleccionar una nova instància amb més prestacions per a poder satisfer la demanda del sistema.

## 16.3 Sostenibilitat social

Per finalitzar l'informe, aquesta secció fa referència a l'impacte en l'àmbit personal i social que ha comportat el desenvolupament del projecte.

### 16.3.1 Posada en producció del projecte

Personalment considero que he tingut sort d'haver pogut formar part d'un projecte dins d'un domini tecnològic molt especial, que és la *blockchain*. A l'inici del projecte, encara que havia sentit parlar de la tecnologia, gairebé no coneixia pràcticament res sobre la tecnologia *blockchain* i, de fet, tenia la sensació que no tenia l'experiència o coneixements necessaris per embarcar-me dins d'aquest món. No obstant això, gràcies als estàndards de desenvolupament de Peersyst i la informació disponible que ofereix Nervos, he pogut adonar-me que és possible aprendre, entendre i desenvolupar, amb el temps suficient, els recursos adequats i una gran quantitat d'esforç, un projecte com aquest.

Gràcies a aquest projecte, he pogut avançar un pas més, dels molts que queden, com a enginyer, sent capaç d'enfrontar-me a problemes complexos i oferir solucions que s'adaptin al món real.

### 16.3.2 Vida útil

Un dels objectius del projecte és poder oferir als usuaris de CKB una millor experiència d'usuari que pugui **augmentar l'adopció de la tecnologia**. La tecnologia *blockchain* comporta una gran llista de beneficis (incloent-hi també inconvenients) com la descentralització del capital, l'automatització de transaccions i l'eliminació d'intermediaris. Però **una tecnologia sense adopció no és útil**. D'aquesta manera, la plataforma fomenta la capacitat de crear una **experiència d'usuari més satisfactòria tant pels desenvolupadors de dApps com els usuaris regulars de la blockchain**.

No obstant això, altres projectes de la *blockchain* CKB, com els esmentats a la Secció 3.1, poden arribar a veure's afectats per una possible pèrdua d'usuaris que vulguin utilitzar dApps enregistrades a la plataforma.

### 16.3.3 Riscos

Ja que la plataforma és un projecte que s'ha produït sobre la *blockchain* CKB i que, per tant, actua com un valor afegit sobre aquesta, no existeixen grans riscos que puguin afectar els usuaris de la *blockchain*. Els afectats serien els usuaris que usessin la cartera de CKBull Wallet i, en cas de desmantellament de la funcionalitat, només es veurien afectats a escala de la funcionalitat de signatura amb l'API, sent capaços d'utilitzar totes les funcionalitats prèvies com rebre, enviar, etc.

No obstant això, els desenvolupadors de dApps serien el sector més afectat, degut a queno existiria cap altre solució amb un comportament semblant dins de l'ecosistema CKB. Encara que aquest projecte es trobi a la primera versió i requereixi l'adopció per a generar aquest risc, cal tindre'l en compte per al desenvolupament de futures funcionalitats i la seguretat associades a aquestes.

# Capítol 17

## Conclusions

Per finalitzar aquesta memòria del projecte del Treball de Fi de Grau, he dedicat aquest últim apartat per a compartir les conclusions i reflexions que he obtingut durant el desenvolupament d'aquest projecte.

Respecte al projecte i la memòria, considero que una solució com és CKBull Signer App està preparada per a poder ser utilitzada de manera introductòria en un entorn de producció. Aquest projecte pot ajudar a molta gent que vulgui poder expandir més enllà l'adopció d'una tecnologia que podria ajudar molt a la societat, tal com és la *blockchain*. També és cert que, tot i ser una tecnologia prometedora i amb molts avantatges, disposa que certs inconvenients com la falta de sostenibilitat en l'àmbit ambiental. No obstant això, una solució com la que proposa aquest projecte és capaç de poder evitar aquest nivell baix de sostenibilitat i fer que una nova funcionalitat dins CKB sigui sostenible. A més, la feina no finalitza amb l'entrega d'aquesta memòria ni amb el llançament a producció del projecte. És pràcticament impossible evitar que sorgeixin nous inconvenients, com ha succeït durant el desenvolupament, encara que s'hagi volgut minimitzar la seva aparició. Però la clau resideix en la cerca de la millora constant, tant a nivell de qualitat del projecte actual per a minimitzar l'impacte com les possibles funcionalitats que es puguin afegir al futur.

En l'àmbit personal, aquest projecte m'ha ajudat a introduir-me al món laboral. No únicament a nivell tècnic, on he sigut conscient de tot l'aprenentatge que he experimentat durant el procés de desenvolupament, sinó a guanyar consciència que qualsevol concepte, amb el temps i esforç, pot aprendre's. Si fa un any hagués sabut que faria un projecte com el Treball de Fi de Grau sobre la tecnologia *blockchain*, no m'ho hauria cregut. I, tot i haver sigut un projecte amb una càrrega de treball considerable i moments difícils, he pogut gaudir tant del desenvolupament com del resultat del projecte.

# Referències

- [1] Peersyst. *Peersyst*. URL: <https://peersyst.com/> (cons. 28-02-2023).
- [2] Nervos. *Nervos*. URL: <https://www.nervos.org/> (cons. 28-02-2023).
- [3] Wikipedia. *Cadena de bloques*. URL: [https://es.wikipedia.org/wiki/Cadena\\_de\\_bloques](https://es.wikipedia.org/wiki/Cadena_de_bloques) (cons. 28-02-2023).
- [4] IBM. *¿Tecnología Blockchain?* URL: <https://www.ibm.com/es-es/topics/what-is-blockchain> (cons. 28-02-2023).
- [5] Nervos Docs. *Nervos Blockchain*. URL: <https://docs.nervos.org/docs/basics/concepts/nervos-blockchain> (cons. 28-02-2023).
- [6] Greg Maxwell Pieter Wuille. *BIP173*. URL: <https://github.com/bitcoin/bips/blob/master/bip-0173.mediawiki> (cons. 30-05-2023).
- [7] Pieter Wuille. *BIP350*. URL: <https://github.com/bitcoin/bips/blob/master/bip-0350.mediawiki> (cons. 30-05-2023).
- [8] Wikipedia. *Criptomonedas*. URL: <https://es.wikipedia.org/wiki/Criptomonedas> (cons. 28-02-2023).
- [9] Ethereum. *Introducción a las dapps*. URL: <https://ethereum.org/es/developers/docs/dapps/> (cons. 28-02-2023).
- [10] Wikipedia. *Cartera de criptodivisa*. URL: [https://es.wikipedia.org/wiki/Cartera\\_de\\_cryptodivisa](https://es.wikipedia.org/wiki/Cartera_de_cryptodivisa) (cons. 28-02-2023).
- [11] Wikipedia. *Unspent transaction output*. URL: [https://en.wikipedia.org/wiki/Unspent\\_transaction\\_output](https://en.wikipedia.org/wiki/Unspent_transaction_output) (cons. 30-05-2023).
- [12] Nervos. *Repositori de codi Neuron*. URL: <https://github.com/nervosnetwork/neuron/releases> (cons. 14-03-2023).
- [13] imToken. *imToken — Ethereum Bitcoin Wallet*. URL: <https://token.im/> (cons. 14-03-2023).
- [14] SafePal. *SafePal Crypto Hardware Wallet (Official) — The best wallet to protect your assets*. URL: <https://www.safepal.com/en/> (cons. 14-03-2023).
- [15] JoyID. *JoyID - Universal Account Protocol for Web3 Mass-adoption*. URL: <https://joy.id/> (cons. 14-03-2023).
- [16] *What is software testing?* URL: <https://www.ibm.com/topics/software-testing> (cons. 20-03-2023).
- [17] James Robertson & Suzanne Robertson. *Volere Requirements Especification Template*. URL: <https://www.cs.uic.edu/~i440/VolereMaterials/templateArchive16/c%5C%20Volere%5C%20template16.pdf> (cons. 28-02-2023).

- [18] scrum.org. *What is Scrum?* URL: <https://www.scrum.org/resources/what-is-scrum> (cons. 28-02-2023).
- [19] Slack. *Slack*. URL: <https://slack.com/intl/es-es> (cons. 28-02-2023).
- [20] Github. *Github*. URL: <https://github.com/> (cons. 28-02-2023).
- [21] Github. *About Projects*. URL: <https://docs.github.com/en/issues/planning-and-tracking-with-projects/learning-about-projects/about-projects> (cons. 28-02-2023).
- [22] RedHat. *What is CI/CD?* URL: <https://www.redhat.com/en/topics/devops/what-is-ci-cd> (cons. 28-02-2023).
- [23] *Features · Github Actions*. URL: <https://github.com/features/actions> (cons. 20-03-2023).
- [24] Wikipedia. *Interfaz de usuario*. URL: [https://es.wikipedia.org/wiki/Interfaz\\_de\\_usuario](https://es.wikipedia.org/wiki/Interfaz_de_usuario) (cons. 06-03-2023).
- [25] Wikipedia. *API*. URL: <https://es.wikipedia.org/wiki/API> (cons. 06-03-2023).
- [26] *Urano Página Principal*. URL: <https://www.urano.studio/> (cons. 12-03-2023).
- [27] *MacBook Pro de 14 pulgadas - Gris espacial - Apple (ES)*. URL: <https://www.apple.com/es/shop/buy-mac/macbook-pro/14-pulgadas-gris-espacial-chip-m2-pro-de-apple-con-cpu-de-10-n%C3%A9gros-y-gpu-de-16-n%C3%A9gros-512gb> (cons. 12-03-2023).
- [28] *Magic Keyboard - Español - Apple (ES)*. URL: <https://www.apple.com/es/shop/product/MK2A3Y/A/magic-keyboard-espaa%C3%BCnol> (cons. 12-03-2023).
- [29] *Logitech MX Master 3 Ratón Inalámbrico Avanzado 4000DPI Grafito — PcComponentes.com*. URL: <https://www.pccomponentes.com/logitech-mx-master-3-raton-inalambrico-avanzado-4000dpi-grafito> (cons. 12-03-2023).
- [30] *Comprar IntelliJ IDEA Ultimate: Precios y licencias, Descuentos - Suscripción a JetBrains Toolbox*. URL: <https://www.jetbrains.com/es-es/idea/buy/?section=commercial&billing=monthly> (cons. 12-03-2023).
- [31] *Pricing · Plans for every developer*. URL: <https://github.com/pricing> (cons. 12-03-2023).
- [32] *Salario en España 2023*. URL: <https://es.talent.com/salary> (cons. 12-03-2023).
- [33] AWS. *My Estimante - FS1*. URL: <https://calculator.aws/#/estimate> (cons. 14-03-2023).
- [34] *CREC Gràcia — Coworking en Barcelona*. URL: <https://coworkingspain.es/espacios/coworking/barcelona/crec-gracia> (cons. 12-03-2023).
- [35] Ian Yang. *CKB Transaction Structure*. URL: <https://github.com/nervosnetwork/rfcs/blob/master/rfcs/0022-transaction-structure/0022-transaction-structure.md> (cons. 30-05-2023).
- [36] Wikipedia. *SOLID*. URL: <https://es.wikipedia.org/wiki/SOLID> (cons. 14-05-2023).
- [37] Refactoring Guru. *Singleton*. URL: <https://refactoring.guru/design-patterns/singleton> (cons. 30-05-2023).
- [38] Snesh Prajapati. *Factory Patterns - Simple Factory Pattern*. URL: <https://www.codeproject.com/Articles/1131770/Factory-Patterns-Simple-Factor-Pattern> (cons. 30-05-2023).
- [39] NodeJS. *NodeJS*. URL: <https://nodejs.org/en> (cons. 30-05-2023).
- [40] MDN web docs. *Javascript*. URL: <https://developer.mozilla.org/es/docs/Web/JavaScript> (cons. 30-05-2023).

- [41] Typescript. *Typescript*. URL: <https://www.typescriptlang.org/> (cons. 30-05-2023).
- [42] Yarn. *Yarn*. URL: <https://yarnpkg.com/> (cons. 30-05-2023).
- [43] Wikipedia. *Docker(software)*. URL: [https://es.wikipedia.org/wiki/Docker\\_\(software\)](https://es.wikipedia.org/wiki/Docker_(software)) (cons. 14-05-2023).
- [44] React. *React*. URL: <https://react.dev/> (cons. 30-05-2023).
- [45] Peersyst. *React Components docs*. URL: <https://peersyst.github.io/react-components-docs/> (cons. 30-05-2023).
- [46] React Router. *React Router*. URL: <https://reactrouter.com/en/main> (cons. 30-05-2023).
- [47] React Query. *React Query*. URL: <https://tanstack.com/query/v3/> (cons. 30-05-2023).
- [48] styled components. *styled components*. URL: <https://styled-components.com/> (cons. 30-05-2023).
- [49] Zustand docs. *Introduction*. URL: <https://docs.pmnd.rs/zustand/getting-started/introduction> (cons. 30-05-2023).
- [50] i18next documentation. *i18next documentation*. URL: <https://www.i18next.com/> (cons. 30-05-2023).
- [51] NestJS. *NestJS*. URL: <https://nestjs.com/> (cons. 30-05-2023).
- [52] TypeORM. *TypeORM docs*. URL: <https://typeorm.io/> (cons. 30-05-2023).
- [53] typestack. *class-validator repository*. URL: <https://github.com/typestack/class-validator> (cons. 30-05-2023).
- [54] PassportJS. *PassportJS*. URL: <https://www.passportjs.org/> (cons. 30-05-2023).
- [55] Nat Sakimura Michael B. Jones John Bradley. *RFC 7519*. URL: <https://datatracker.ietf.org/doc/html/rfc7519> (cons. 30-05-2023).
- [56] Wikipedia. *HMAC*. URL: <https://es.wikipedia.org/wiki/HMAC> (cons. 30-05-2023).
- [57] React Native. *React Native. Learn once, write anywhere*. URL: <https://reactnative.dev/> (cons. 30-05-2023).
- [58] Expo. *Expo*. URL: <https://expo.dev/> (cons. 30-05-2023).
- [59] React Navigation. *React Navigation*. URL: <https://reactnavigation.org/> (cons. 30-05-2023).
- [60] Recoil. *Recoil*. URL: <https://recoiljs.org/> (cons. 30-05-2023).
- [61] ckb-js. *Lumos - A full featured dapp framework for Nervos CKB - GitHub*. URL: <https://github.com/ckb-js/lumos> (cons. 30-05-2023).
- [62] Wikipedia. *Data Transfer Object*. URL: [https://en.wikipedia.org/wiki/Data\\_transfer\\_object](https://en.wikipedia.org/wiki/Data_transfer_object) (cons. 30-05-2023).
- [63] Wikipedia. *Dependency Injection*. URL: [https://en.wikipedia.org/wiki/Dependency\\_injection](https://en.wikipedia.org/wiki/Dependency_injection) (cons. 30-05-2023).
- [64] Microsoft. *Design the infrastructure persistence layer*. URL: <https://learn.microsoft.com/en-us/dotnet/architecture/microservices/microservice-ddd-cqrs-patterns/infrastructure-persistence-layer-design> (cons. 30-05-2023).
- [65] Wikipedia. *Data mapper pattern*. URL: [https://en.wikipedia.org/wiki/Data\\_mapper\\_pattern](https://en.wikipedia.org/wiki/Data_mapper_pattern) (cons. 30-05-2023).

- [66] Wikipedia. *Unit testing*. URL: [https://en.wikipedia.org/wiki/Unit\\_testing](https://en.wikipedia.org/wiki/Unit_testing) (cons. 30-05-2023).
- [67] Wikipedia. *Mock object*. URL: [https://en.wikipedia.org/wiki/Mock\\_object](https://en.wikipedia.org/wiki/Mock_object) (cons. 30-05-2023).
- [68] GitBook. *GitBook - Where technical teams document*. URL: <https://www.gitbook.com/> (cons. 30-05-2023).
- [69] Wikipedia. *Seguridad de la capa de transporte*. URL: [https://es.wikipedia.org/wiki/Seguridad\\_de\\_la\\_capa\\_de\\_transporte](https://es.wikipedia.org/wiki/Seguridad_de_la_capa_de_transporte) (cons. 14-03-2023).
- [70] Wikipedia. *Llicència X11*. URL: [https://ca.wikipedia.org/wiki/Llic%C3%A8ncia\\_X11](https://ca.wikipedia.org/wiki/Llic%C3%A8ncia_X11) (cons. 14-03-2023).
- [71] Generalitat de Catalunya. *Factor d'emissió de l'energia elèctrica: el mix elèctric. Canvi climàtic*. URL: [https://canviclimate.gencat.cat/es/actua/factors\\_demissio\\_associats\\_a\\_lenergia/](https://canviclimate.gencat.cat/es/actua/factors_demissio_associats_a_lenergia/) (cons. 30-05-2023).
- [72] Teads. *Carbon footprint estimator for AWS instances*. URL: [https://engineering.teads.com/sustainability/carbon-footprint-estimator-for-aws-instances/?estimation=true&instance\\_id=2615&region\\_id=2248&compute\\_hours=30#calculator](https://engineering.teads.com/sustainability/carbon-footprint-estimator-for-aws-instances/?estimation=true&instance_id=2615&region_id=2248&compute_hours=30#calculator) (cons. 06-02-2023).
- [73] Amazon. *OGADA 13W Bombillas LED*. URL: [https://www.amazon.es/OGADA-Bombillas-casquillo-Reemplazar-Incandescente/dp/B0B5HJB41R/ref=sr\\_1\\_5?keywords=bombillas+led&qid=1686593820&sr=8-5](https://www.amazon.es/OGADA-Bombillas-casquillo-Reemplazar-Incandescente/dp/B0B5HJB41R/ref=sr_1_5?keywords=bombillas+led&qid=1686593820&sr=8-5) (cons. 06-02-2023).
- [74] Teads. *Carbon footprint estimator for AWS instances*. URL: [https://engineering.teads.com/sustainability/carbon-footprint-estimator-for-aws-instances/?estimation=true&instance\\_id=2623&region\\_id=2248&compute\\_hours=1#calculator](https://engineering.teads.com/sustainability/carbon-footprint-estimator-for-aws-instances/?estimation=true&instance_id=2623&region_id=2248&compute_hours=1#calculator) (cons. 06-02-2023).
- [75] University of Cambridge. *Cambridge Bitcoin Electricity Consumption Index (CBECI)*. URL: <https://ccaf.io/cbnsi/cbeci> (cons. 06-02-2023).
- [76] Nervos. *CKB Explorer*. URL: <https://explorer.nervos.org/charts/transaction-count> (cons. 06-02-2023).
- [77] Tesla. *European Union energy label*. URL: [https://www.tesla.com/en\\_eu/support/european-union-energy-label](https://www.tesla.com/en_eu/support/european-union-energy-label) (cons. 06-02-2023).

# Apèndix A

## Especificació

### A.1 Diagrama de casos d'ús



Figura 58: Diagrama de casos d'ús.

Font: elaboració pròpia.

# Apèndix B

## Disseny

### B.1 Rutes de CKBull Signer API

#### B.1.1 Rutes per peticions d'inici de sessió (sign-in-requests)

<b>Descripció</b>	Obtindre peticions d'inici de sessió segons els atributs <i>status</i> , <i>network</i> i <i>address</i> .
<b>Ruta</b>	/api/sign-in-requests
<b>Mètode</b>	GET
<b>Paràmetres</b>	status: SignInRequestStatus, network: Network, address: string
<b>Resposta</b>	200: [{ id: 0, name: string, description: string, email: string, projectUrl: string, supportUrl: string, apiKey: string, termsUrl: string, privacyPolicyUrl: string, image: string, createdAt:date, updatedAt:date }]

Taula 74: Petició per obtindre peticions d'inici de sessió.

<b>Descripció</b>	Crear petició d'inici de sessió com a dApp mitjançant autenticació API.
<b>Ruta</b>	/api/sign-in-requests
<b>Mètode</b>	POST
<b>Capceleres</b>	x-timestamp: number, x-signature: string, x-api-key: string
<b>Resposta</b>	200: { id: 0, signInToken: string, status: pending, createdAt: date, expiresAt: date, sessionExpiresAt: date, metadata: { address: string, network: mainnet }, dapp: DAppDto } 403: Not authorized,

Taula 75: Petició per crear una petició d'inici de sessió.

<b>Descripció</b>	Obtindre peticions d'inici de sessió mitjançant l' <i>apiKey</i> de la dApp.
<b>Ruta</b>	/api/sign-in-requests/dapp/{apiKey}
<b>Mètode</b>	GET
<b>Paràmetres</b>	apiKey: string
<b>Resposta</b>	200: [{ id: 0, signInToken: string, status: pending, createdAt: date, expiresAt: date, sessionExpiresAt: date, metadata: { address: string, network: mainnet }, dapp: DAppDto }]

Taula 76: Petició per obtindre les peticions d'inici de sessió creades per una dApp.

<b>Descripció</b>	Obtindre petició d'inici de sessió mitjançant el <i>signInToken</i> .
<b>Ruta</b>	/api/sign-in-requests/{token}
<b>Mètode</b>	GET
<b>Paràmetres</b>	token: string
<b>Resposta</b>	200: { id: 0, signInToken: string, status: pending, createdAt: date, expiresAt: date, sessionExpiresAt: date, metadata: { address: string, network: mainnet }, dapp: DAppDto } 404: Not Found

Taula 77: Petició per obtenir informació d'una petició d'inici de sessió.

<b>Descripció</b>	<i>Endpoint</i> pensat per fer <i>polling</i> de l'estat de la petició d'inici de sessió.
<b>Ruta</b>	/api/sign-in-requests/{token}/status
<b>Mètode</b>	GET
<b>Paràmetres</b>	token: string
<b>Resposta</b>	200: { signInToken: string, status: pending, } 404: Not Found

Taula 78: Petició per fer *polling* d'una petició d'inici de sessió.

<b>Descripció</b>	<i>Endpoint</i> per acceptar una petició per part d'un usuari de CKBull Wallet.
<b>Ruta</b>	/api/sign-in-requests/{token}/sign
<b>Mètode</b>	POST
<b>Paràmetres</b>	token: string
<b>Cos</b>	{ metadata: { address: string, network: Network }, }
<b>Resposta</b>	200: { id: 0, signInToken: string, status: pending, createdAt: date, expiresAt: date, sessionExpiresAt: date, metadata: { address: string, network: mainnet }, dapp: DAppDto } 400: Bad request 404: Not Found

Taula 79: Petició d'acceptació d'una petició d'inici de sessió.

<b>Descripció</b>	<i>Endpoint</i> per declinar una petició per part d'un usuari de CKBull Wallet.
<b>Ruta</b>	/api/sign-in-requests/{token}/decline
<b>Mètode</b>	POST
<b>Paràmetres</b>	token: string
<b>Resposta</b>	200: { id: 0, signInToken: string, status: pending, createdAt: date, expiresAt: date, sessionExpiresAt: date, metadata: { address: string, network: mainnet }, dapp: DAppDto } 404: Not Found

Taula 80: Petició de declinació d'una petició de d'inici de sessió.

<b>Descripció</b>	<i>Endpoint</i> per desconectar i invalidar les peticions d'inici de sessió signades per part d'un usuari de CKBull Wallet.
<b>Ruta</b>	/api/sign-in-requests/{token}/disconnect
<b>Mètode</b>	POST
<b>Paràmetres</b>	token: string
<b>Resposta</b>	200: void 404: Not Found

Taula 81: Petició de desconnexió de dApps.

### B.1.2 Rutes per peticions d'autenticació

<b>Descripció</b>	<i>Endpoint</i> per a iniciar sessió com a desenvolupador de dApps a CKBull Developer Console.
<b>Ruta</b>	/api/auth/login
<b>Mètode</b>	POST
<b>Cos</b>	{ email: string, password: string }
<b>Resposta</b>	200: { access_token: string } 400: Bad Request (invalid_credentials)

Taula 82: Petició per iniciar sessió com desenvolupador.

<b>Descripció</b>	<i>Endpoint</i> que redirigeix cap a l'inici de sessió de Google.
<b>Ruta</b>	/api/auth/google
<b>Mètode</b>	GET
<b>Resposta</b>	200: void

Taula 83: Petició per iniciar sessió amb Google com desenvolupador.

<b>Descripció</b>	<i>Endpoint</i> per redirigir al client després d'haver iniciat sessió amb Google.
<b>Ruta</b>	/api/auth/google/callback
<b>Mètode</b>	GET
<b>Resposta</b>	200: void

Taula 84: Petició per redirigir després d'iniciar sessió

<b>Descripció</b>	<i>Endpoint</i> per validar un <i>token</i> prèviament enviat per correu electrònic al desenvolupador.
<b>Ruta</b>	/api/auth/verify-email
<b>Mètode</b>	POST
<b>Cos</b>	{ token: string }
<b>Resposta</b>	200: { access_token: string } 404: Not found (token_not_found) 409: Conflict (token_already_used)

Taula 85: Petició per validar direcció de correu com desenvolupador.

<b>Descripció</b>	<i>Endpoint</i> per generar un <i>token</i> per restablir la contrasenya del desenvolupador.
<b>Ruta</b>	/api/auth/recover-password
<b>Mètode</b>	POST
<b>Cos</b>	{ email: string }
<b>Resposta</b>	200: void 404: Not found (user_not_found)

Taula 86: Petició per recuperar la contrasenya com desenvolupador.

<b>Descripció</b>	<i>Endpoint</i> per restablir la contrasenya d'un desenvolupador amb el <i>token</i> enviat per correu electrònic.
<b>Ruta</b>	/api/auth/reset-password
<b>Mètode</b>	POST
<b>Cos</b>	{ password: string, token: string }
<b>Resposta</b>	200: void 404: Not found (token_not_found) 409: Not found (token_expired)

Taula 87: Petició per a restablir la contrasenya com desenvolupador.

### B.1.3 Rutes per peticions d'usuari

<b>Descripció</b>	<i>Endpoint</i> per obtindre informació de l'usuari que sol·licita la informació.
<b>Ruta</b>	/api/user
<b>Mètode</b>	GET
<b>Autenticació</b>	Bearer
<b>Resposta</b>	200: { id: 0, name: string, email: string, type: string, googleAuth: true } 401: Unauthorized 403: Forbidden

Taula 88: Petició per a obtenir informació d'un usuari.

<b>Descripció</b>	<i>Endpoint</i> per crear un nou desenvolupador al sistema.
<b>Ruta</b>	/api/user
<b>Mètode</b>	POST
<b>Resposta</b>	200: { name: string, email: string, password: string, } 409: Conflict(email_already_expired)

Taula 89: Petició per a enregistrar un usuari.

<b>Descripció</b>	<i>Endpoint</i> per a editar la informació de l'usuari que sol·licita la petició.
<b>Ruta</b>	/api/user
<b>Mètode</b>	PATCH
<b>Autenticació</b>	Bearer
<b>Cos</b>	{ name: string, email: string, password: string, }
<b>Resposta</b>	void 401: Unauthorized 403: Forbidden

Taula 90: Petició per a editar un usuari.

<b>Descripció</b>	<i>Endpoint</i> per esborrar la instància de l'usuari que sol·licita la petició.
<b>Ruta</b>	/api/user
<b>Mètode</b>	DELETE
<b>Autenticació</b>	Bearer
<b>Resposta</b>	void 401: Unauthorized 403: Forbidden

Taula 91: Petició per eliminar un usuari.

#### B.1.4 Rutes per peticions de fitxers

<b>Descripció</b>	<i>Endpoint</i> per a pujar imatges al servidor FS2.
<b>Ruta</b>	/api/file/image
<b>Mètode</b>	POST
<b>Autenticació</b>	Bearer
<b>Cos</b>	imatge
<b>Resposta</b>	200: string 401: Unauthorized 403: Forbidden

Taula 92: Petició per a pujar una imatge.

### B.1.5 Rutes per peticions de dApps

<b>Descripció</b>	<i>Endpoint</i> per obtindre totes les dApps generades per el sol·licitant.
<b>Ruta</b>	/api/dapps
<b>Mètode</b>	GET
<b>Autenticació</b>	Bearer
<b>Resposta</b>	[{ id: 0, name: string, description: string, email: string, projectUrl: string, supportUrl: string, apiKey: string, termsUrl: string, privacyPolicyUrl: string, image: string, createdAt: date, updatedAt: date }] 401: Unauthorized 403: Forbidden

Taula 93: Petició per a obtindre totes les dApps d'un usuari.

<b>Descripció</b>	<i>Endpoint</i> per obtindre la informació d'una dApp mitjançant l'atribut <i>id</i> .
<b>Ruta</b>	/api/dapps/{id}
<b>Mètode</b>	GET
<b>Autenticació</b>	Bearer
<b>Resposta</b>	<pre>{   id: 0,   name: string,   description: string,   email: string,   projectUrl: string,   supportUrl: string,   apiKey: string,   termsUrl: string,   privacyPolicyUrl: string,   image: string,   createdAt: date,   updatedAt: date }</pre> <p>401: Unauthorized      403: Forbidden      404: Not found</p>

Taula 94: Petició per a obtindre una dApp.

<b>Descripció</b>	<i>Endpoint</i> per enregistrar una nova dApp.
<b>Ruta</b>	/api/dapps
<b>Mètode</b>	POST
<b>Autenticació</b>	Bearer
<b>Cos</b>	<pre>{   name: string,   description: string,   email: string,   projectUrl: string,   supportUrl: string,   apiKey: string,   termsUrl: string,   privacyPolicyUrl: string,   image: string, }</pre>
<b>Resposta</b>	<pre>201: {   id: 0,   name: string,   description: string,   email: string,   projectUrl: string,   supportUrl: string,   apiKey: string,   termsUrl: string,   privacyPolicyUrl: string,   image: string,   createdAt: date,   updatedAt: date }</pre> <p>401: Unauthorized    403: Forbidden    409: Conflict</p>

Taula 95: Petició per a enregistrar una dApp.

<b>Descripció</b>	<i>Endpoint</i> per editar la informació de la dApp amb identificador <i>id</i> .
<b>Ruta</b>	/api/dapps/{id}
<b>Mètode</b>	PATCH
<b>Autenticació</b>	Bearer
<b>Cos</b>	<pre>{   name: string,   description: string,   email: string,   projectUrl: string,   supportUrl: string,   apiKey: string,   termsUrl: string,   privacyPolicyUrl: string,   image: string, }</pre>
<b>Resposta</b>	200: void 401: Unauthorized 403: Forbidden 409: Conflict

Taula 96: Petició per a enregistrar una dApp.

<b>Descripció</b>	<i>Endpoint</i> per eliminar la dApp amb l'identificador <i>id</i> .
<b>Ruta</b>	/api/dapps/{id}
<b>Mètode</b>	DELETE
<b>Autenticació</b>	Bearer
<b>Paràmetres</b>	id: number
<b>Resposta</b>	200: void 401: Unauthorized 403: Forbidden 404: Not found

Taula 97: Petició per eliminar una dApp.

<b>Descripció</b>	<i>Endpoint</i> per regenerar l' <i>API secret</i> de la dApp amb identificador <i>id</i> .
<b>Ruta</b>	/api/dapps/{id}/credentials
<b>Mètode</b>	GET
<b>Autenticació</b>	Bearer
<b>Paràmetres</b>	id: number
<b>Resposta</b>	200: apiSecret: string 401: Unauthorized 403: Forbidden 404: Not found

Taula 98: Petició per eliminar una dApp.

### B.1.6 Rutes per peticions de transacció

<b>Descripció</b>	<i>Endpoint</i> per obtindre totes les peticions de transacció amb els atributs <i>status</i> , <i>network</i> i <i>address</i> .
<b>Ruta</b>	/api/transaction-requests
<b>Mètode</b>	GET
<b>Paràmetres</b>	status: TransactionRequestStatus, network: Network, address: string
<b>Resposta</b>	200: [{ id: number, transactionToken: string, status: TransactionRequestStatus, transaction: { id: number, transactionHash: string, transaction: { cellProvider: null, cellDeps: [], headerDeps: [], inputs: [], outputs: [], witnesses: [], signingEntries: [], inputSinces: object } }, createdAt: date, expiresAt: date, signInRequest: SignInRequestDto }]

Taula 99: Petició d'obtenció de peticions de transaccions.

<b>Descripció</b>	<i>Endpoint</i> per crear una petició de transacció mitjançant l'autenticació API, un <i>signInToken</i> i una transacció.
<b>Ruta</b>	/api/transaction-requests
<b>Mètode</b>	POST
Capceleres	x-timestamp: number, x-signature: string, x-api-key: string
<b>Cos</b>	{ signInToken: string, transaction: { cellProvider: null, cellDeps: [], headerDeps: [], inputs: [], outputs: [], witnesses: [], fixedEntries: [], signingEntries: [], inputSinces: object } }
<b>Resposta</b>	200: [         {           id: number,           transactionToken: string,           status: TransactionRequestStatus,           transaction: {             id: number,             transactionHash: string,             transaction: {               cellProvider: null,               cellDeps: [],               headerDeps: [],               inputs: [],               outputs: [],               witnesses: [],               fixedEntries: [],               signingEntries: [],               inputSinces: object             }           },           createdAt: date,           expiresAt: date,           signInRequest: SignInRequestDto         }       ],       401: Unauthorized       403: Forbidden

Taula 100: Petició de creació de peticions de transaccions.

<b>Descripció</b>	<i>Endpoint</i> per obtindre totes peticions de transacció mitjançant <i>apiKey</i> .
<b>Ruta</b>	/api/transaction-requests/dapp/{apiKey}
<b>Mètode</b>	GET
<b>Paràmetres</b>	apiKey: string
<b>Resposta</b>	<pre> 200: [   {     id: number,     transactionToken: string,     status: TransactionRequestStatus,     transaction: {       id: number,       transactionHash: string,       transaction: {         cellProvider: null,         cellDeps: [],         headerDeps: [],         inputs: [],         outputs: [],         witnesses: [],         fixedEntries: [],         signingEntries: [],         inputSinces: object       },       createdAt: date,       expiresAt: date,       signInRequest: SignInRequestDto     }   ] 404: Not found </pre>

Taula 101: Petició d'obtenció de peticions de transacció per dApp.

<b>Descripció</b>	<i>Endpoint</i> per obtindre una petició de transacció mitjançant el <i>transactionToken</i> .
<b>Ruta</b>	/api/transaction-requests/{transactionToken}
<b>Mètode</b>	GET
<b>Paràmetres</b>	transactionToken: string
<b>Resposta</b>	<p>200: {      id: number,      transactionToken: string,      status: TransactionRequestStatus,      transaction: {      id: number,      transactionHash: string,      transaction: {      cellProvider: null,      cellDeps: [],      headerDeps: [],      inputs: [],      outputs: [],      witnesses: [],      fixedEntries: [],      signingEntries: [],      inputSinces: object      },      createdAt: date,      expiresAt: date,      signInRequest: SignInRequestDto      }      }</p> <p>404: Not found</p>

Taula 102: Petició d'obtenció de petició de transacció per transactionToken.

<b>Descripció</b>	<i>Endpoint</i> pensat per fer <i>polling</i> de l'estat de la petició de transacció.
<b>Ruta</b>	/api/transaction-requests/{transactionToken}/status
<b>Mètode</b>	GET
<b>Paràmetres</b>	transactionToken: string
<b>Resposta</b>	<p>200: {      transactionToken: string,      status: TransactionRequestStatus,      }      }</p> <p>404: Not found</p>

Taula 103: Petició d'obtenció de l'estat petició de transacció per transactionToken.

<b>Descripció</b>	<i>Endpoint</i> per generar una transacció de <i>tokens</i> natius mitjançant una <i>address</i> i la quantitat a enviar.
<b>Ruta</b>	/api/transaction-requests/generate-native-token-transaction
<b>Mètode</b>	POST
<b>Cos</b>	amount: string, to: string
<b>Resposta</b>	201: { cellProvider: null, cellDeps: [], headerDeps: [], inputs: [], outputs: [], witnesses: [], fixedEntries: [], signingEntries: [], inputSincs: object }

Taula 104: Petició de generació d'una transacció de token natiu.

<b>Descripció</b>	<i>Endpoint</i> per generar una transacció de <i>NFTs</i> mitjançant una <i>address</i> i el <i>NFT</i> a enviar.
<b>Ruta</b>	/api/transaction-requests/generate-nft-transaction
<b>Mètode</b>	POST
<b>Cos</b>	nft: string, to: string
<b>Resposta</b>	201: { cellProvider: null, cellDeps: [], headerDeps: [], inputs: [], outputs: [], witnesses: [], fixedEntries: [], signingEntries: [], inputSincs: object }

Taula 105: Petició de generació d'una transacció d'un nft.

<b>Descripció</b>	<i>Endpoint</i> per a enviar una petició de transacció signada per part d'un usuari de CKBull Wallet.
<b>Ruta</b>	/api/transaction-requests/{transactionToken}/sign
<b>Mètode</b>	POST
<b>Cos</b>	<pre>{   signInToken: string,   transaction: {     id: number,     transactionHash: string,     transaction: {       cellProvider: null,       cellDeps: [],       headerDeps: [],       inputs: [],       outputs: [],       witnesses: [],       fixedEntries: [],       signingEntries: [],       inputSinces: object     }   } }</pre>
<b>Resposta</b>	<pre>201: {   id: number,   transactionToken: string,   status: TransactionRequestStatus,   transaction: {     id: number,     transactionHash: string,     transaction: {       cellProvider: null,       cellDeps: [],       headerDeps: [],       inputs: [],       outputs: [],       witnesses: [],       fixedEntries: [],       signingEntries: [],       inputSinces: object     },     createdAt: date,     expiresAt: date,     signInRequest: SignInRequestDto   } }</pre>

Taula 106: Petició de generació d'una transacció d'un nft.

<b>Descripció</b>	<i>Endpoint</i> per a rebutjar una petició de transacció per part d'un usuari de CKBull Wallet.
<b>Ruta</b>	/api/transaction-requests/{transactionToken}/decline
<b>Mètode</b>	POST
<b>Cos</b>	<pre>{   signInToken: string,   transaction: {     id: number,     transactionHash: string,     transaction: {       cellProvider: null,       cellDeps: [],       headerDeps: [],       inputs: [],       outputs: [],       witnesses: [],       fixedEntries: [],       signingEntries: [],       inputSinces: object     }   } }</pre>
<b>Resposta</b>	<pre>201: {   id: number,   transactionToken: string,   status: TransactionRequestStatus,   transaction: {     id: number,     transactionHash: string,     transaction: {       cellProvider: null,       cellDeps: [],       headerDeps: [],       inputs: [],       outputs: [],       witnesses: [],       fixedEntries: [],       signingEntries: [],       inputSinces: object     },     createdAt: date,     expiresAt: date,     signInRequest: SignInRequestDto }</pre>

Taula 107: Petició de generació d'una transacció d'un nft.

### B.1.7 Rutes per peticions de *playground*

<b>Descripció</b>	<i>Endpoint</i> per desconectar i invalidar les peticions d'inici de sessió signades per part d'un usuari de CKBull Wallet.
<b>Ruta</b>	/api/playground/verify
<b>Mètode</b>	POST
<b>Capçaleres</b>	x-timestamp: number, x-signature: string, x-api-key: string
<b>Resposta</b>	200: verify: boolean 404: Not Found

Taula 108: Petició de desconnexió de dApps.