



Universitat de Girona

Escola Politècnica Superior

Vote-vote, un sistema de votació electrònica
basat en Blockchain que incentiva la participació.

Tutor

Andrés El-Fakdi Sencianes

Document

Memòria

Alumne

Bogdan Cata

Índex

1	Introducció	4
2	Estudi de viabilitat	5
2.1	Viabilitat econòmica	5
2.2	Viabilitat tècnica	5
2.3	Viabilitat temporal	6
2.4	Viabilitat legal	6
2.5	Conclusions sobre la viabilitat	6
3	Metodologia	6
3.1	Metodologia àgil	7
4	Planificació	8
4.1	Estructura de descomposició de treball	9
4.2	Paquets de treball	9
4.2.1	Pla d'activitats	13
4.2.2	Diagrama d'activitats	14
4.2.3	Camí crític	15
4.2.4	Diagrama de Gantt	16
5	Marc de treball i conceptes previs	16
5.1	Conceptes relacionats amb votacions	16
5.1.1	Votació	17
5.1.2	Vot	17
5.1.3	Vot incentivat	17
5.1.4	Configuració	17
5.2	Viabilitat del punt de vista legal	18
5.3	Conceptes tècnics	18
5.3.1	Estructura client-servidor	18
5.3.2	Fitxers JSON	19
5.3.3	Estructura REST	20
5.4	Anàlisi	21
6	Requisits del sistema	21
6.1	Requisits funcionals	21
6.2	Requisits no funcionals	22
6.3	Requisits per l'entorn de desenvolupament	22
6.4	Requisits per a utilitzar Vote-Vote	23
7	Estudis i decisions	23
7.1	Hardware	23
7.2	Software	24

7.2.1	Documentació	24
7.3	Tecnologies utilitzades	24
7.3.1	React-Native	24
7.3.2	Node.JS	25
7.3.3	Express.JS	25
7.3.4	MongoDB i Solidity	25
7.4	Eines	25
7.4.1	Especificacions servidor on està allotjat	25
8	Anàlisi i disseny del sistema	26
8.1	Descripció	26
8.2	Anàlisi	26
8.2.1	Creador de la votació	26
8.2.2	Votant	27
8.2.3	Administrador	27
8.3	Diagrames i fitxes de casos d'ús	27
8.3.1	Creador de la votació	29
8.3.2	Votant	32
8.3.3	Administrador	36
8.4	Models de la base de dades	37
8.4.1	Answers	38
8.4.2	EmailQueues	39
8.4.3	Results	40
8.4.4	Users	41
8.4.5	Votations	42
8.4.6	Votes	43
8.4.7	Configurations	44
8.5	Model de dades Smart Contract	44
8.6	Disseny de les interfícies d'usuari	45
8.6.1	Home	46
8.6.2	Crear votació	47
8.7	Pantalla post-creació votació	49
8.7.1	Plantilla correu votació creada	50
8.7.2	Votar	52
8.7.3	Plantilla correu animar a votar	54
8.7.4	Plantilla correu certificat blockchain	55
8.7.5	Resultats públics	55
8.7.6	Resultats creador votació	57
8.7.7	Login	58
8.7.8	Pantalla editar configuració	59

9 Implementació i proves	59
9.1 Primers passos	60
9.2 Pm2	60
9.3 MongoDB	61
9.4 Nominalia	61
9.5 Nginx	64
9.6 Comprovació del funcionament	65
10 Implantació i resultats	66
10.1 Resultats obtinguts	66
11 Conclusions	66
11.1 Aprenentatge	66
11.2 Conclusions personals	67
12 Treball futur	67
13 Bibliografia	67
14 Annexos	68
14.1 Pressupost	68
14.1.1 Mètode d'estimació per paquets de treball	68
14.1.2 Estimació del temps per cada paquet de treball	68
14.1.3 Estimació del cost monetari del projecte	70
15 Manual d'usuari	70

1 Introducció

Els sistemes i conceptes relacionats amb procediments de votació electrònica han adquirit molta importància en els darrers anys i ara, amb la pandèmia de COVID-19 que estem patint, encara més. En aquest camp s'han proposat diverses solucions electròniques, les més recents basades en tecnologia Blockchain. Les característiques i propietats més destacades de la tecnologia Blockchain són: transparència, immutabilitat i traçabilitat de les dades, unes avantatges que són molt interessants en el camp que ocupa aquest projecte, els processos electorals i els garanties que ofereixen en format electrònic. A més, un dels problemes recurrents en tot procés electoral és la participació o, més concretament, els mecanismes d'incentivació de la participació. Aquest projecte busca oferir solucions en aquest sentit.

En aquest projecte final de grau presentem l'aplicació Vote-Vote, una plataforma digital per dur a terme processos de votació electrònica de forma ràpida i fiable, basada en Blockchain, que utilitza mecanismes de votació originals per tal d'incentivar la participació. Vote-Vote pretén trencar amb el principi de votació "una persona un vot" i introdueix un mecanisme més atractiu que permet emetre pre-vots per companys, amics i familiars que també tenen el dret a vot en un determinat procés de votació. El vots emesos en segona ronda han de ser ratificats per els destinataris que poden confirmar-lo, anular-lo o canviar-ne el sentit.

El projecte té com a objectiu principal desenvolupar el front-end (interfície web d'usuari), el back-end (part que realitza tota la lògica de l'aplicació) i l' Smart Contract (contracte intel·ligent preparat per ésser executat a una xarxa Blockchain i que garanteix la immutabilitat del procés d'acord a les condicions pactades al contracte i que assegurin la traçabilitat, seguretat i transparència de tot el procés).

Un segon objectiu del projecte pretén desenvolupar un mecanisme d'incentivació de vot associat a l'aplicació Vote-Vote, que permet emetre pre-vots per altres persones del cens i que actuen com a mecanisme dinamitzador del procés electoral.

Finalment, es presentaran un estudi dels resultats obtinguts mitjançant una bateria de proves experimentals amb usuaris reals que ens proveiran de les dades necessàries per a poder estudiar els resultats estadístics dels processos de votació realitzats i valorar-ne els resultats.

Aquest projecte ha estat proposat pel grup de recerca TECNIO Centre EASY de la Universitat de Girona, el mateix lloc on he realitzat la meua estada en entorn laboral i on segueixo formant part de forma activa de l'equip de desenvolupament software. El grup de recerca m'ha facilitat les eines per tal de poder desenvolupar tot el projecte: la màquina on s'ha desenvolupat i el servidor on està actualment allotjat el projecte per tal que sigui d'ús públic.

2 Estudi de viabilitat

En aquest segon apartat s'exposa un estudi de viabilitat del projecte. Primer farem una definició sobre que és i per a què s'utilitza. Un **estudi o anàlisi de viabilitat** consisteix en un anàlisi que prediu l'èxit o el fracàs d'un projecte a partir d'uns determinats paràmetres que el descriuen, com poden ésser la rendibilitat, les necessitats de mercat, la factibilitat política o la legislació, entre d'altres. A partir d'aquestes característiques hem de realitzar un estudi on es preveu si està ben emprat l'esforç que costa realitzar aquest projecte, o en resum, si realment val la pena dur a terme aquest projecte.

En la planificació de projectes de caire informàtic, l'estudi de viabilitat es revisa sovint al llarg de tota la planificació a mida que aquesta avança. En una etapa més madura del projecte, els resultats obtinguts acostumen a ser més semblants a la realitat. En fase inicial, l'anàlisi de viabilitat ens dona una informació molt allunyada de la realitat, donat que les tecnologia emprada és molt recent i està sotmesa a canvis importants que en modifiquen les característiques d'una forma molt important.

Per a portar a terme aquest estudi sobre la viabilitat del nostre projecte enumerarem els diferents aspectes a tenir en compte, que són els següents:

1. **Viabilitat econòmica**
2. **Viabilitat tècnica**
3. **Viabilitat temporal**
4. **Viabilitat legal**

A continuació profundirem en els apartats esmentats per arribar a una conclusió.

2.1 Viabilitat econòmica

En aquest apartat es valorarà si el projecte és possible des del punt de vista econòmic. Aquest projecte és bastant rendible, ja que, dels únics costos que suposa és comprar el domini i el hosting, en aquest cas no comptarem les hores de dedicació emprades al seu desenvolupament donat que, s'ha considerat que són les apropiades en el desenvolupament d'un *Projecte de Fi de Grau* i entren dins l'activitat acadèmica d'aquest.

2.2 Viabilitat tècnica

En aquest apartat es valorarà si el projecte és possible des del punt de vista d'habilitats tècniques. El projecte està plantejat per ser desenvolupat en React-Native per la part de front-end. Aquest llenguatge ofereix una alta compatibilitat entre les 3 grans

plataformes (Web, iOS i Android), a part d'això, hi estic bastant familiaritzat, donat que en el meu entorn laboral és un dels que més utilitzem per a desenvolupament. Per la part de back-end s'ha optat per a desenvolupar-ho en Node.JS, un framework de Javascript que és l'adequat per aquesta tasca i que també fem servir de forma comuna al grup.

En tercer lloc per l'emmagatzematge de dades hem triat MongoDB. Hem escollit aquesta base de dades perquè permet emmagatzemar conjunts de dades no estructurades, és més flexible davant absència de dades i és molt escalable. Finalment, per la part de Blockchain el contracte intel·ligent (Smart Contract) estarà desenvolupat en Solidity i estarà publicat a la xarxa pública d'Alastria. Aquesta xarxa basada en el protocol Ethereum és ara mateix gratuïta i no hem de pagar costos d'execució com si hauríem de fer en una xarxa Blockchain pública. Altres opcions que s'han estudiat han estat Rinkeby o Ropsten.

El mitjans tècnics que s'utilitzaran són perfectament coneguts per l'autor del projecte i es considera el projecte perfectament viable des del punt de vista tècnic.

2.3 Viabilitat temporal

En aquest apartat es valorarà si el projecte és possible des del punt de vista temporal. El projecte s'ha desenvolupat d'acord als terminis temporals fixats per la Universitat al inici respecte els temps d'entrega, per tant, podem concloure que des del punt de vista temporal és viable.

2.4 Viabilitat legal

En aquest apartat es valorarà si el projecte és possible des del punt de vista legal. El projecte serà desenvolupat utilitzant software lliure. La implementació del projecte es farà des de 0, per tant, del punt de vista legal també el considero viable.

2.5 Conclusions sobre la viabilitat

Les valoracions descrites en els apartats anteriors ens indiquen que aquest projecte **és viable**.

3 Metodologia

A continuació exposarem la metodologia que utilitzarem per a desenvolupar aquest projecte. Abans però, detallarem què és la metodologia d'un projecte. La metodologia d'un projecte fa referència al conjunt de procediments per a la planificació i gestió

de tots els aspectes del projecte, des de la gestió de recursos fins a la coordinació de l'equip de treball o la relació amb tots els interessats en els resultats del projecte.

3.1 Metodologia àgil

AGILE i **SCRUM** són els dos sistemes de planificació i organització del treball més utilitzats en disseny software. La metodologia **AGILE** enfoca els projectes de desenvolupament de software de manera que els requisits i solucions evolucionen segons les necessitats del projecte. El treball es realitza mitjançant la col·laboració d'equips auto-organitzats i multidisciplinaris. La metodologia **SCRUM** destaca per treball en equip, on tots els membres estan involucrats des de bon inici fins a la fi del projecte. Els diferents rols dins de l'equip estan molt definits i cada membre de l'equip té definides les seves tasques de manera que la resolució de problemes sigui el més efectiva possible.

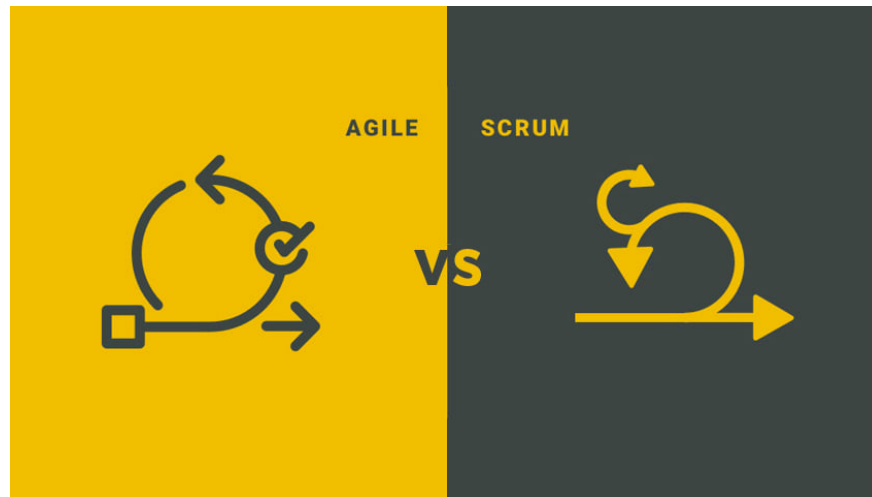


Figura 1: Representació gràfica de les dues metodologies

Per a l'elaboració d'aquest projecte s'ha escollit treballar amb metodologia **SCRUM**. Tot i que el projecte es realitzarà de manera individual, aquesta forma de treballar és comuna al TECNIO Centre EASY i facilita la progressió del projecte dins el flux de projectes vius que es desenvolupen al grup. La metodologia SCRUM consta d'un seguit de iteracions en les quals dividim el projecte anomenades **Sprints**.

En SCRUM és molt comú utilitzar un gestor de tasques on consten les tasques que s'han de realitzar dins d'un sprint i el seu estat, si s'està fent, qui, etc... Vet aquí una captura d'exemple.

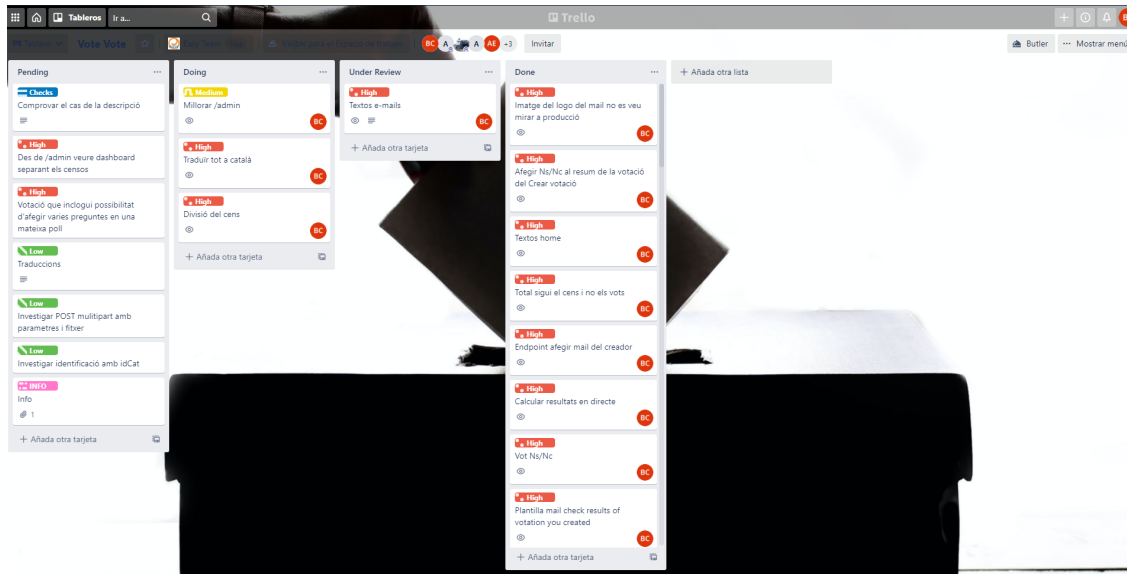
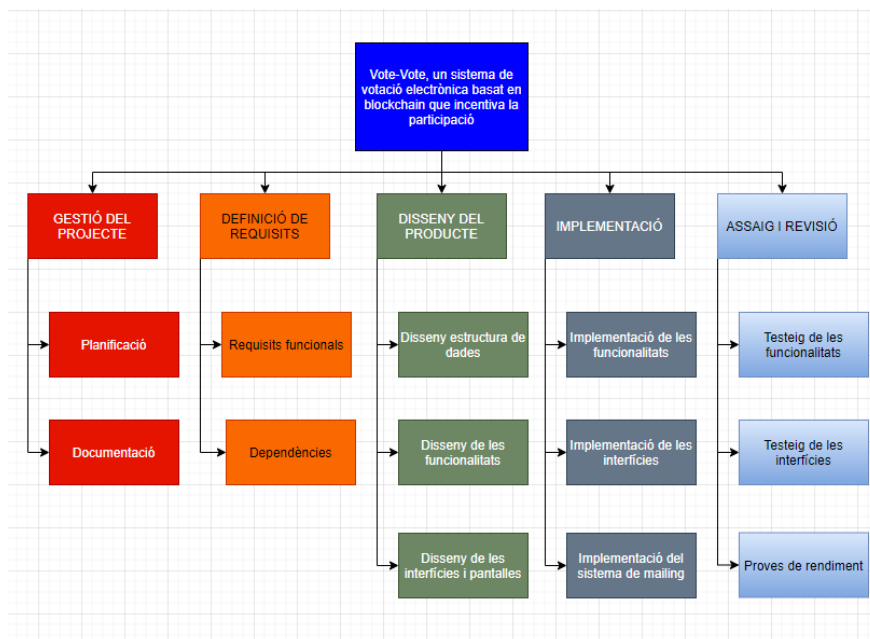


Figura 2: Captura de pantalla del tauler de Trello creat per tenir un control de tasques

4 Planificació

En aquest apartat es mostrarà la descomposició jeràrquica de la feina a ser executada per tal d'assolir els objectius d'aquest projecte. Cada nivell descendent de la Estructura de descomposició de treball (EDT) representa una definició amb un detall incrementat de la feina a realitzar.

4.1 Estructura de descomposició de treball



4.2 Paquets de treball

En aquest apartat s'exposaran els paquets de treball mostrats anteriorment de forma precisa.

Nom del paquet	Planificació
Descripció	Avaluació del conjunt de tasques que s'han de realitzar i aproximar un temps per aquestes.
Tasques a realitzar	<ul style="list-style-type: none"> - Enumerar les tasques a desenvolupar - <u>Avaluar temporalment les tasques</u>
Lliuraments	
Temps	7 dies

Nom del paquet	Documentació del projecte
Descripció	Proveir informació relativa al desenvolupament de l'aplicació
Tasques a realitzar	<ul style="list-style-type: none"> - Proveir informació sobre les diferents tasques realitzades al llarg del procés de desenvolupament de l'aplicació
Lliuraments	
Temps	14 dies

Nom del paquet	Requisits funcionals
Descripció	Enumeració del conjunt de requisits de l'aplicació
Tasques a realitzar	<ul style="list-style-type: none"> - Enumerar tots els requisits que volem que es realitzin durant el desenvolupament del nostre projecte
Lliuraments	
Temps	7 dies

Nom del paquet	Dependències
Descripció	Enumerar el conjunt de dependències funcionals de l'aplicació.
Tasques a realitzar	<ul style="list-style-type: none"> - Enumerar totes les dependències directes i indirectes que tindrà la nostra aplicació
Lliuraments	
Temps	7 dies

Nom del paquet	Disseny de l'estructura de dades
Descripció	Estudi i tria de les estructures de dades que <u>s'utilitzaran</u>
Tasques a realitzar	<ul style="list-style-type: none"> - Trobar el millor SGBD i dissenyar la Base de dades - Disseny del Smart Contract
Lliuraments	
Temps	14 dies

Nom del paquet	Disseny de les funcionalitats
Descripció	Enumeració del conjunt de funcionalitats a assolir
Tasques a realitzar	<ul style="list-style-type: none"> - Enumerar el conjunt de funcionalitats que ha d'assolir la nostra aplicació
Lliuraments	
Temps	7 dies

Nom del paquet	Disseny de les interfícies i pantalles
Descripció	Dissenyar el conjunt d' <u>interfícies</u> i pantalles que tindrà la nostra aplicació
Tasques a realitzar	<ul style="list-style-type: none"> - Dissenyar les interfícies de l'aplicació <u>necessàries</u> per tal de complir els requisits funcionals plantejats.
Lliuraments	
Temps	7 dies

Nom del paquet	Implementació de les funcionalitats
Descripció	Implementació del conjunt de funcionalitats de l'aplicació
Tasques a realitzar	<ul style="list-style-type: none"> - Implementar les diferents funcionalitats de l'aplicació
Lliuraments	
Temps	45 dies

Nom del paquet	Implementació de les <u>interfícies</u>
Descripció	Implementació del conjunt d' <u>interfícies</u> de l'aplicació
Tasques a realitzar	<ul style="list-style-type: none"> - Implementar les diferents <u>interfícies</u> de l'aplicació
Lliuraments	
Temps	45 dies

Nom del paquet	Implementación de mailing
Descripció	Implementació d'un sistema de mailing per a enviar e-mails des de l'aplicació
Tasques a realitzar	- Implementar un sistema d'enviament de correus electrònics.
Lliuraments	
Temps	20 dies

Nom del paquet	<u>Testeig de les funcionalitats</u>
Descripció	Comprovació del funcionament esperat de les funcionalitats implementades
Tasques a realitzar	- Testejar totes les funcionalitats de l'aplicació conforme l'aplicació és comporta com desitgem
Lliuraments	
Temps	14 dies

Nom del paquet	<u>Testeig de les interfícies</u>
Descripció	Comprovació de l'adaptació de les <u>interfícies</u> en diferents mides de pantalla
Tasques a realitzar	- <u>Comprovar</u> que l'aplicació digui el màxim responsive possible
Lliuraments	
Temps	14 dies

Nom del paquet	Proves de rendiment
Descripció	Avaluar com es comporta l'aplicació quan hi ha molt volum d'usuaris
Tasques a realitzar	- Realitzar proves per a comprovar com es comporta l'aplicació quan és utilitzada a gran escala
Lliuraments	
Temps	14 dies

4.2.1 Pla d'activitats

El pla d'activitats es la part del projecte on es recullen el conjunt de tasques mínimes necessàries per a la consecució del desenvolupament de l'aplicació. En la planificació d'activitats s'elabora un cronograma d'activitats. En ell, posarem la data i durada de la mateixa, així com les sub-tasques per a la seva execució. Per començar, farem un llistat del conjunt d'activitats. Per cada activitat es mostrara el codi de l'activitat, el nom de l'activitat, la durada i les sub-tasques si escau. Seguint la metodologia de l'apartat anterior, la unitat de mesura utilitzada es el dia. Per un dia entendrem 8 hores de treball.

1. Activitat 1

- (a) **Codi:** A01
- (b) **Nom:** Crear client i servidor bàsics
- (c) **Durada:** 40 dies

2. Activitat 2

- (a) **Codi:** A02
- (b) **Nom:** Client
- (c) **Durada:** 25 dies
- (d) **Sub-tasques:**
 - i. Pantalla home
 - ii. Pantalla crear votació
 - iii. Pantalla votar
 - iv. Plantilla e-mails
 - v. Pantalla resultats
 - vi. Pantalla login admin
 - vii. Pantalla configuració admin

3. Activitat 3

- (a) **Codi:** A03
- (b) **Nom:** Servidor
- (c) **Durada:** 25 dies
- (d) **Sub-tasques:**
 - i. Crear votació
 - ii. Votar
 - iii. Enviament de e-mails

- iv. Càlcul de resultats
- v. Login com a admin
- vi. Carregar configuració votacions admin

4. Activitat 4

- (a) **Codi:** A04
- (b) **Nom:** Smart Contract
- (c) **Durada:** 20 dies
- (d) **Sub-tasques:**
 - i. Idear Smart Contract
 - ii. Desenvolupar-lo
 - iii. Publicació a Alastria

5. Activitat 5

- (a) **Codi:** A05
- (b) **Nom:** Publicació a l'entorn de producció
- (c) **Durada:** 2 dies
- (d) **Sub-tasques:**
 - i. Configurar domini
 - ii. Configurar nginx a la màquina

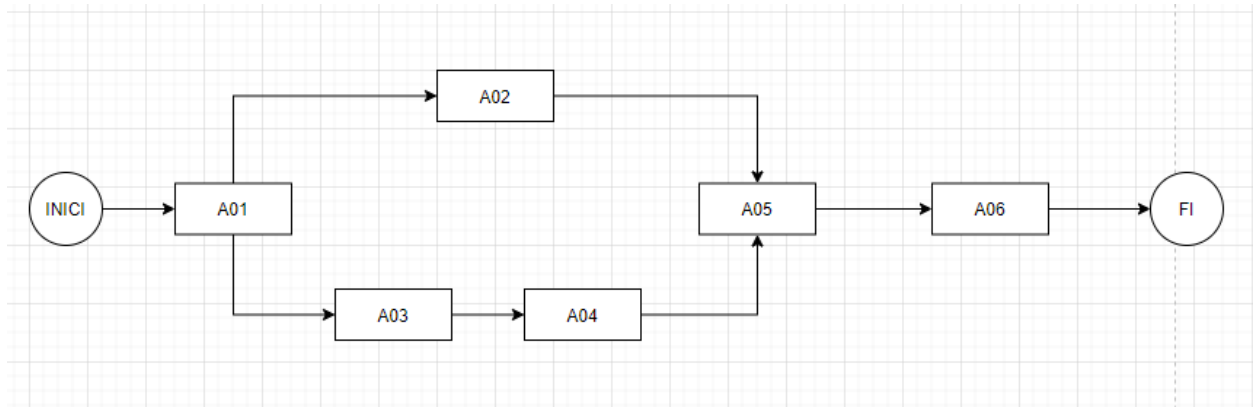
6. Activitat 6

- (a) **Codi:** A06
- (b) **Nom:** Proves
- (c) **Durada:** 14 dies

4.2.2 Diagrama d'activitats

El diagrama d'activitats és la representació gràfica del procés de desenvolupament del projecte. En aquest diagrama es representa el flux de treball pas a pas.

Un cop mostrat el conjunt d'activitats necessàries per tal de desenvolupar l'aplicació, les unirem i planificarem l'ordre en el que s'han de realitzar i quines dependències existeixen entre si. Per dur a terme la planificació començarem amb el diagrama d'activitats.



4.2.3 Camí crític

El mètode de la ruta o del camí crític és un algorisme utilitzat per el càlcul de temps i terminis amb el qual realitzarem l'estimació del temps de desenvolupament del projecte.

Per tal de realitzar aquest algorisme, primer, necessitarem una taula on es mostrin les diferents activitats amb els seus predecessors i els costos d'aquestes.

Activitat	Predecessors	Durada
A01	-	40 dies
A02	A01	25 dies
A03	A01	25 dies
A04	A03	20 dies
A05	A02, A04	2 dies
A06	A06	14 dies

Un cop sabem els predecessors podem calcular el camí crític.

1. $E_1 = 0$
2. $E_2 = E_1 + t_{1,2} [t_{1,2} = A_{01} = 10] = 0 + 40 = 10$
3. $E_3 = E_2 + t_{2,3} [t_{2,3} = A_{02} = 6] = 40 + 25 = 65$
4. $E_4 = E_2 + t_{2,4} [t_{2,4} = A_{03} = 5] = 40 + 25 = 65$
5. $E_5 = E_4 + t_{4,5} [t_{4,5} = A_{04} = 3] = 65 + 20 = 85$
6. $E_6 = \text{Max}(E_i + t_{i,6}) [i=3,5]$
7. $E_6 = 65$

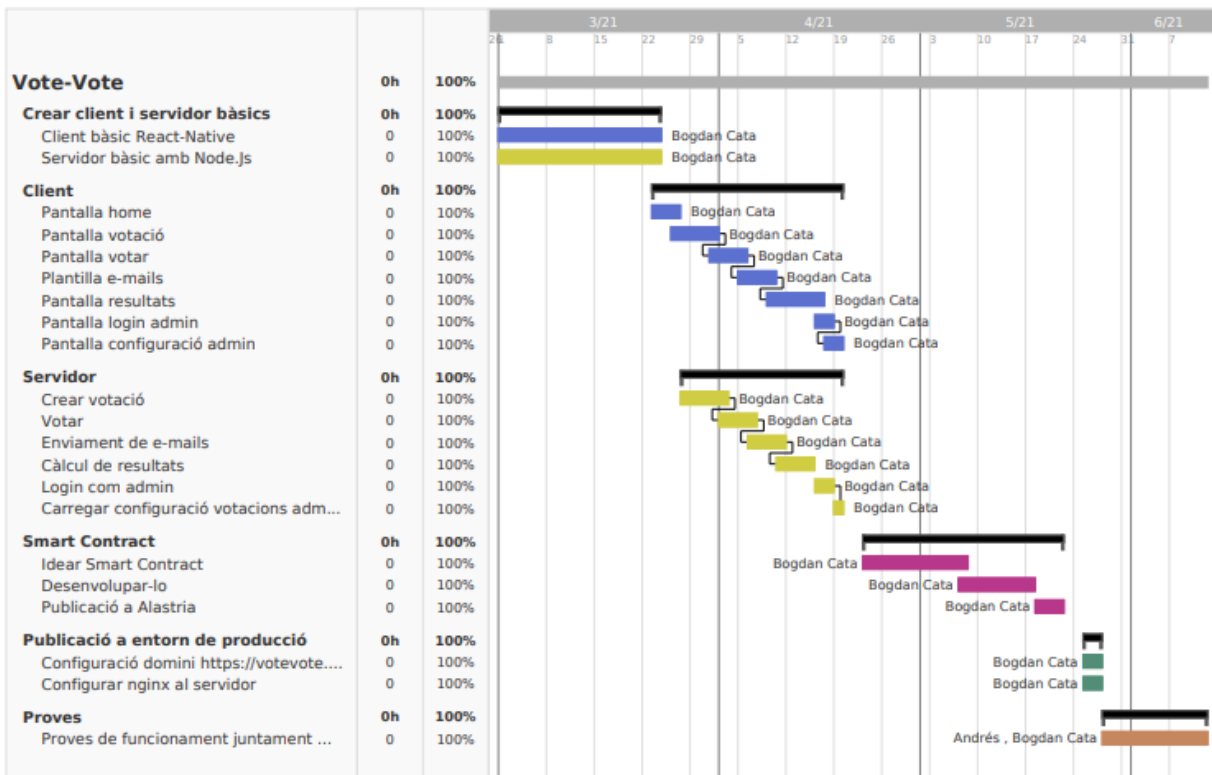
$$8. E7 = E6 + t_{6,7} [t_{6,7} = A_{05} = 1] = 65 + 2 = 67$$

$$9. E8 = E7 + t_{7,8} [t_{7,8} = A_{06} = 1] = 67 + 14 = 81$$

Un cop finalitzat el càlcul, el cost final obtingut és de 81 dies. A més, podem dir que el camí crític en aquest diagrama d'activitats és: **A01-A03-A04-A05-A06**

4.2.4 Diagrama de Gantt

El diagrama de Gantt es una eina gràfica que té l'objectiu d'exposar el temps de dedicació previst en les diferents tasques o activitats al llarg d'un temps total determinat. Tot i això, el diagrama de Gantt no indica les relacions existents entre activitats. Per acabar la secció de planificació i gestió del temps del projecte, mostrarem el diagrama de Gantt pertanyent a la nostra aplicació.



5 Marc de treball i conceptes previs

5.1 Conceptes relacionats amb votacions

En aquest apartat es resumiran els conceptes més bàsics relacionats amb el funcionament final de l'aplicació, donat que, durant la memòria faig menció d'elles i trobo oportú descriure-les.

5.1.1 Votació

Anomenarem votació la pregunta o el conjunt d'aquestes que un usuari crea des de la nostra plataforma amb els paràmetres següents:

El conjunt de participants que participaran a aquesta votació.

La pregunta o el conjunt d'aquestes amb les seves corresponents respostes.

La data i hora d'inici a la qual es podran contestar la o les preguntes i la data i hora de finalització a partir de la qual ja no serà permès participar a aquesta.

El tipus de la votació, si és una votació de tipus normal, aquesta essent una persona un vot com qualsevol votació coneguda fins ara o bé sigui de tipus incentivat el qual nosaltres emetent el nostre vot podem animar a un participant del cens a votar triant les opcions que creiem que ell respondrà a les preguntes per tal de facilitar-li el màxim i per tal d'augmentar la participació.

La opció de recordatori, es un flag que habilitem en el cas que no vulguem que els participants que incentiven a altres triïn les opcions per ells, atès que, d'aquesta manera descartem la possible influència que hi pot haver d'un respecte l'altre.

Finalment l'enllaç de seguiment, aquest flag l'activem en el cas que vulguem que els participants de les votacions sàpiguen els resultats de la votació la qual han participat.

5.1.2 Vot

És la resposta o el conjunt d'aquestes que un usuari pertanyent al cens emet amb finalitat d'esser comptabilitzades a una votació a la qual ha estat convocat.

5.1.3 Vot incentivat

Es el mateix que un vot però amb la diferència que el tipus de la votació ha d'èsser de tipus incentivat i que un usuari del cens prèviament hagi votat i vulgui que nosaltres ho fem també amb les opcions ja seleccionades o en blanc amb l'opció de recordatori.

5.1.4 Configuració

S'ha optat per fer un apartat de configuració des de la part d'administrador per tal de fer experiments.

Aquesta configuració s'aplica a les votacions de tipus incentivat i el que es fa és que de la part d'administrador nosaltres entrem dos percentatges, el percentatge del cens pel qual volem que la votació sigui realment de tipus incentivat i el percentatge del cens per la qual, encara que el creador hagi seleccionat que la votació sigui de tipus incentivat serà de tipus normal.

Per tant, quan es crea una votació de tipus incentivat, agafem la configuració més recent que tenim i el que fem és dividir aleatòriament el cens entre aquests percentatges.

Posarem un exemple per a què sigui més entenedor.

Un usuari crea una votació de tipus incentivat i l'última configuració que nosaltres hem entrat és de 50% cens incentivat i 50% cens normal, per tant, si el cens d'aquesta votació consta de 10 persones, aleatòriament es faran dos subconjunts de 5 i 5 en els que pels primers 5 la votació serà de tipus incentivat i es podran animar entre ells i els altres 5 serà una votació normal de una persona un vot sense possibilitat d'animar a ningú.

Una gran pregunta seria perquè hem fet això. Hem optat per implementar aquesta funcionalitat per a veure realment si, amb el mètode d'animar els altres a votar augmenta la participació de la gent, ja que, així tenim les mateixes condicions, o sigui, és la mateixa votació, atès que, es podria fer una votació de tipus incentivat i una votació de tipus normal i comparar-ho però llavors serien preguntes diferents i no seria tant precís, tot i així, tampoc és el 100% precís el mètode aplicat perquè esta clar que, participar a una votació acaba essent elecció d'un mateix i si no es vol per molt que s'animi no s'acabarà fent.

5.2 Viabilitat del punt de vista legal

Un dels meus objectius a llarg plaç fent aquest projecte és que s'arribi a utilitzar per votacions d'àmbit més oficial com poden arribar a ser per exemple, unes eleccions de poble o si més no unes eleccions generals, el problema que actualment hi veig és que el Smart Contract on hi consten totes les transaccions de l'aplicació ha d'estar publicat a una xarxa de blockchain, en el meu cas aquesta és *Alastria*, donat que, és gratuïta, però és clar, aquesta xarxa és espanyola, per tant, si es volgués utilitzar per unes eleccions d'algun altre país, podria esser que aquest no estigui gaire d'acord amb que el contracte estigui publicat en una xarxa espanyola.

Per tant, la intenció futura d'aquest projecte seria trobar alguna alternativa a aquesta que tenim actualment per tal que es pugui complir l'objectiu i que en un futur, moltes votacions de tipus oficials utilitzin *Vote-Vote*.

5.3 Conceptes tècnics

Aquest treball s'ha enfocat d'una manera molt tècnica i per tant cal explicar els següents conceptes per entendre algunes decisions preses, els problemes que han portat i la complexitat no aparent dels objectius que s'han aconseguit.

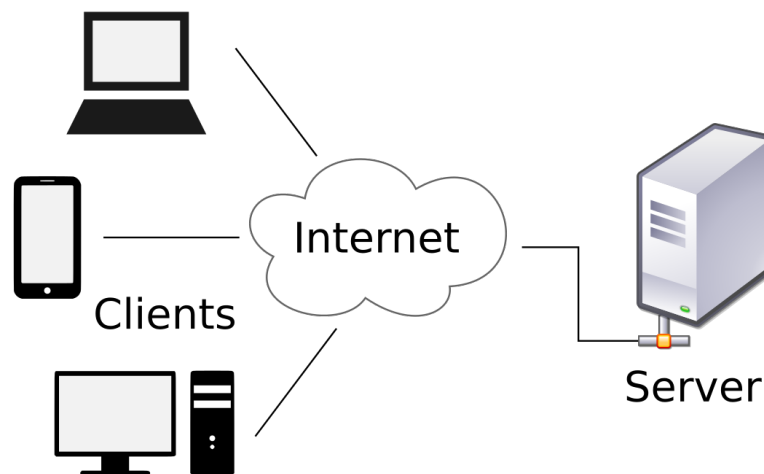
5.3.1 Estructura client-servidor

És un model que descriu un sistema amb un funcionament distribuït fent servir un protocol de comunicació, normalment a la web. El componen dos elements:

- **Client:** és l'agent que demana i envia recursos al servidor.
- **Servidor:** és l'agent que rep, processa i envia totes les peticions dels clients.

Per comunicar els dos components o agents cal fer servir un estàndard que tècnicament s'anomena protocol. Els que s'han utilitzat són els següents:

1. **HTTPS (HyperText Transfer Protocol Secure)**: és l'estàndard per intercanviar dades a la xarxa World Wide Web. És el protocol que fa servir l'aplicació per comunicar el client amb el servidor.
2. **SMTP (Simple Mail Transfer Protocol)**: és el protocol per comunicar clients i servidors de correus electrònics.
3. **SSH (Secure Shell)**: serveix per controlar remotament el servidor fent servir comandes. Aquest protocol es fa servir al projecte per controlar la màquina en la qual s'han desplegat el servidor i el client.



5.3.2 Fitxers JSON

Els fitxers amb extensió *JSON* són uns fitxers de text pla que tenen un format determinat i és el format estàndard que s'utilitza per a la comunicació entre un client i un servidor.

Un exemple de fitxer *JSON* seria el següent:

```
1 {  
2   "number": 1,  
3   "array": ["array", "de", "exemple"],  
4   "bool": true,  
5   "object": {  
6     "field": "camp de l'objecte d'exemple"
```



```
7 }  
8 }
```

En l'exemple mostrat podem veure un fitxer *JSON* que conté un objecte que té com a paràmetres un de cada tipus, en primer lloc tenim *number* que és la manera en la que es passa un paràmetre numèric, en segon lloc tenim *array* la qual mostra com es passa una llista que en aquest cas conté tres elements de tipus *string*, en tercer lloc tenim *bool* que mostra com es passa un paràmetre de tipus booleà i per últim tenim *object* que mostra com es passa un objecte el qual conté un sol camp, en aquest cas *field* que té com a valor de tipus *string*

5.3.3 Estructura REST

És un concepte de disseny que defineix com estructurar un servei web en el model client-servidor. Per aquest treball s'ha fet servir bàsicament per definir la interfície que comunicarà les dades entre el client i el servidor (API).

Un dels dos conceptes més importants a destacar són els diferents tipus de crides que utilitza REST per comunicar les dades utilitzant el protocol HTTPS:

1. **GET**: per indicar que es vol obtenir un recurs.
2. **POST**: per indicar que es vol crear un recurs.
3. **PUT**: per indicar que es vol substituir un recurs (o crear-lo en cas de que no existeixi).
4. **DELETE**: per indicar que es vol eliminar un recurs.

I l'altre concepte és com s'indica a quin recurs afecta la crida i com s'envien les dades en cas de necessitar-les. Per facilitar l'enteniment posarem un exemple.

Per una banda tenim el recurs afectat (concretament *votation*) que generalment s'indica a la URL. Per exemple amb la següent URL:

<https://votevote.centreeasy.com/votation?id=1>

Ens estem referint a la votació passant un paràmetre de clau *id* i valor 1. D'aquesta manera i depenent del tipus de crida, estarem, per exemple, eliminant o obtenint la informació de la votació en qüestió.

D'altra banda, la manera d'enviar informació al servidor es fa utilitzant el cos (*body*) del missatge HTTPS. Bé pot ser text pla, com fitxers o altres tipus de dades.

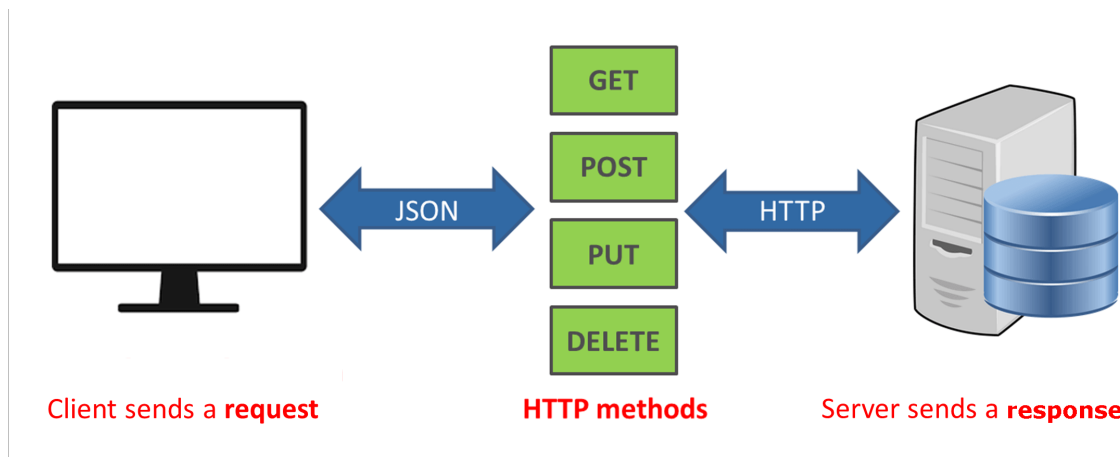


Figura 3: Representació gràfica d'una interacció entre client i servidor amb una API REST

5.4 Anàlisi

Fer una anàlisi si amb el mètode de incentivació proposat augmenta la participació perquè la intenció de la incentivació és que a la gent que li fa mandra anar a votar ho tingui de la manera més accessible possible.

6 Requisits del sistema

6.1 Requisits funcionals

1. L'usuari podrà crear una votació normal o una votació de tipus incentivada on un usuari votant podrà animar a votar altres usuaris pertanyents al cens.
2. Un usuari pertanyent al cens rebrà un correu de notificació conforme hi ha una nova votació en la que pot votar
3. Un usuari pertanyent al cens podrà votar a la votació que ha estat invitat i en el cas que la votació sigui de tipus incentivat animar a algú a votar (proposant una resposta a la votació on simplement ell ha de confirmar el vot) o bé fer-li un recordatori enviant-li una notificació sempre i quan aquest pertanyi al cens.
4. En el cas que el tipus de votació sigui amb vot incentivat, l'usuari del qual han incentivat o recordat per participar a la votació serà notificat per tal d'acceptar la votació proposada o bé realitzar el seu propi vot.
5. En el cas que l'usuari creador de la votació ho desitgi, els usuaris pertanyents al cens rebran un correu amb l'enllaç dels resultats per tal de consultar els resultats de la votació la qual han participat.

6. L'usuari creador de la votació podrà consultar els resultats en directe mitjançant un enllaç, en el cas que aquest ho desitgi, podrà introduir el seu correu electrònic i aquest enllaç li serà enviat a la seva safata d'entrada per a tenir-lo present per a futures consultes.
7. L'usuari votant podrà veure les transaccions de la seva votació realitzada a [l'explorador de Blockchain](#)
8. Un usuari amb privilegis d'administrador podrà configurar les votacions amb la finalitat de fer experiments i simular una votació incentivada en certs percentatges (que en una votació de tipus incentivat no siguin tots els vots de tipus incentivat sinó només un percentatge d'aquests, o sigui, per exemple si tenim una configuració de 50% normals i 50% incentivats si tenim un cens de 10 persones per a 5 persones de escollides de manera aleatòria la votació serà una votació normal i per els 5 restants serà una votació incentivada on només es podrà incentivar a una d'aquestes 5 persones que pertanyen al mateix "subgrup").
9. Qualsevol usuari que disposi de connexió a internet podrà accedir a l'enllaç <https://votevote.centreeasy.com/> i executar tota l'aplicació.

6.2 Requisits no funcionals

1. Disposar d'una aplicació amb una interfície atractiva i perfectament funcional.
2. Disposar d'una aplicació segura.
3. Disposar d'una alta compatibilitat entre sistemes operatius/dispositius.
4. Disposar d'un disseny web adaptable per a tots els dispositius (Responsive web design)
5. Disposar d'una alta traçabilitat i transparència de les dades utilitzant tecnologia Blockchain.

6.3 Requisits per l'entorn de desenvolupament

Com posteriorment es mencionarà, per a desenvolupar aquest projecte es necessita un conjunt de software específic per tal de poder córrer el projecte i també requereix certa memòria RAM motiu per el qual, durant el desenvolupament la nostra màquina executarà el client i el servidor alhora. El software que és indispensable pel desenvolupament és el següent:

1. [IntelliJ IDEA 2021.1.1](#)
2. [MongoDB](#)

3. [Node.js](#)
4. [Ganache-Cli](#)
5. [RabbitMQ](#)
6. [Robo 3T](#) (no indispensable però recomanable per a veure gràficament la base de dades)
7. [Google Chrome](#)

6.4 Requisits per a utilitzar Vote-Vote

Per a utilitzar Vote-Vote un usuari ha de complir els següents requisits:

1. Tenir connexió a la xarxa d'internet
2. Utilitzar algun dels navegadors de la llista (s'ha provat amb aquests, per tant els altres els considerarem no compatibles ja que, en desconexem el seu comportament)
 - (a) **Firefox i derivats (developer edition per exemple)**
 - (b) **Chrome**
 - (c) **Edge**
 - (d) **Safari**
3. Disposar d'una adreça de correu electrònic (s'ha provat amb mailinator, nominalia i gmail, es desconex el comportament amb altres adreces)

7 Estudis i decisions

El desenvolupament íntegre d'aquest treball de fi de grau es va dur a terme en un ordinador de sobretaula amb les següents especificacions tècniques:

7.1 Hardware

- **CPU:** Intel(R) Core(TM) i7-6700 CPU 3.40GHz 3.41GHz
- **Memòria RAM:** 20.0 GB (19.0 GB usable)
- **Disc dur:** 1.81 TB de capacitat HDD
- **Altres:** Dues pantalles una de resolució 1920x1080 i l'altre de 1680x1050

7.2 Software

- **Sistema operatiu:** Windows 10 Home
- **IDE:** [Intelij IDEA 2021.1.1](#)
- **MongoDB:** S'ha de tenir instal·lat el servidor de [MongoDB](#) per a poder realitzar transaccions a la base de dades.
- **Node.js:** S'ha de tenir instal·lat [Node.js](#) per tal de poder correr el servidor.
- **Ganache-Cli:** S'ha de tenir instal·lat [Ganache-Cli](#) per tal de poder realitzar transaccions a una xarxa de *Blockchain* emulada per tal de fer proves.
- **RabbitMQ:** S'ha de tenir instal·lat el [RabbitMQ](#) per a la gestió de cues de les transaccions al *Blockchain*
- **Robo3T:** És recomanable instal·lar el [Robo 3T](#) per a gestionar més fàcilment la Base de Dades
- **Navegador web:** Com a principal s'ha utilitzat [Chrome](#) i per mòbil s'ha utilitzat [Safari](#) però també s'han realitzat proves en altres navegadors com [Firefox developer edition](#)

7.2.1 Documentació

- **Overleaf:** Editor de text [LaTeX](#)
- **Draw.io:** Recurs web per crear [figures](#)

7.3 Tecnologies utilitzades

L'objectiu principal del desenvolupament d'aquest projecte és que sigui compatible per el màxim de dispositius possibles i per això s'ha fet un petit estudi i una tria dels diversos llenguatges i frameworks disponibles per a desenvolupar un projecte amb aquests requisits.

7.3.1 React-Native

Per la part de front-end s'ha optat per utilitzar *React-Native* per bàsicament dues raons, en primer lloc per la alta compatibilitat entre plataformes que ens dona aquest llenguatge, atès que, amb el mateix codi o bé fent alguns petits retocs podem compilar el codi en les 3 grans plataformes (Android, Web i iOS) i en segon lloc per la certa experiència que tinc en aquest llenguatge, donat que, algunes altres opcions per a desenvolupar-la han estat *Flutter* (compatibilitat entre les 3 plataformes però fase beta per Web) o *Swift* (compatibilitat només per iOS) però han estat descartades per la baixa experiència en aquests llenguatges i per la por de compatibilitat entre plataformes.

7.3.2 Node.JS

Per la part de back-end s'ha optat per utilitzar *Node.JS* per bàsicament les raons anteriors, per l'experiència i també perquè és dels frameworks de Javascript més utilitzats i més estables que hi ha i amb facilitat d'instal·lar moltíssimes llibreries que puguin aportar diverses funcionalitats.

7.3.3 Express.JS

Express.js és un framework que serveix per construir un servidor en *Node.js*. Es va escollir bàsicament per que és el que més s'utilitza i per tant pel que més documentació hi ha.

7.3.4 MongoDB i Solidity

Per la part d'emmagatzematge de dades s'utilitzarà una Base de dades no relacional de *MongoDB* i aquesta serà reforçada amb *Blockchain* on els *Smart Contracts* estaran programats amb *Solidity*

7.4 Eines

Per a desenvolupar aquest projecte necessitarem diverses eines de software. En primer lloc per a "picar"el codi necessitarem un **IDE** que particularment he optat per utilitzar *IntelliJ IDEA* de JetBrains per la costum que tinc d'utilitzar-lo i perquè essent estudiant de la UdG ens atorga el privilegi de tenir una llicència gratuïta. Per a tenir un control de versions d'aquest projecte tindrem un *Repository* a la plataforma *Bitbucket* on hi anirem pujant les diferents versions del projecte utilitzant *Git*. Llavors per a allotjar l'aplicació utilitzarem un servidor que té com a sistema operatiu *Ubuntu 18* i que prèviament hem fet tota la configuració. El domini de l'aplicació serà <https://votevote.centreeasy.com/>, ja que, utilitzarem un subdomini de l'empresa on estic treballant (**Centre TECNIO Easy**) i ens ha proporcionat l'ajuda de deixar-nos allotjar el projecte en un dels seus servidors.

7.4.1 Especificacions servidor on està allotjat

És un hosting proporcionat per [AWS \(Amazon Web Services\)](#) que té com a sistema operatiu un *Ubuntu 18* i té de memòria RAM 7.7 GB.

```
easy@idl5294:~$ free -h
              total        used        free      shared  buff/cache   available
Mem:           7.7G         4.5G         216M         143M           3.0G           2.8G
Swap:          0B           0B           0B
```

Figura 4: Captura de la comanda `free -h` per a comprovar la RAM del servidor

```

easy@id15294:~$ pm2 ls

```

App name	id	version	mode	pid	status	restart	uptime	cpu	mem	user	watching
back-to-life	5	1.0.0	fork	1824	online	0	29D	0.2%	59.7 MB	easy	disabled
idart	1	0.4.0	fork	203280	online	27	6D	0.6%	180.9 MB	easy	disabled
obrir	6	1.0.0	fork	165002	online	30	10h	0.2%	152.9 MB	easy	disabled
venushina	2	0.0.0	fork	1815	online	0	29D	0.2%	38.7 MB	easy	disabled
venushina-app	3	2.0.5	fork	16165	online	26	29D	0.2%	97.2 MB	easy	disabled
vote-vote-react	7	0.1.0	fork	43889	online	1	5D	0.3%	88.1 MB	easy	disabled
wedigital	4	0.1.0	fork	49945	online	0	27D	0.2%	273.6 MB	easy	disabled

```

Use `pm2 show <id|name>` to get more details about an app

```

Figura 5: Captura de la comanda pm2 ls per a comprovar que el nostre projecte està allotjat en aquest servidor (és el que té nom vote-vote-react i id 7)

8 Anàlisi i disseny del sistema

En aquest apartat del projecte es descriurà l'aplicació en detall, el seu disseny, els elements que la componen, etc...

8.1 Descripció

Dins l'aplicació disposarem de tres rols segons el tipus d'usuari que hi intervingui:

- **Creador de la votació:** és l'usuari que crea la votació introduint els paràmetres bàsics per a la creació d'aquesta, paràmetres que veurem més endavant en aquest mateix apartat.
- **Votant:** és l'usuari que ha estat nomenat dins el cens de la votació el qual ha de contestar la votació proposada i, segons els paràmetres de la votació, incentivar a un altre usuari pertanyent al cens si s'escau.
- **Administrador:** és l'usuari que decideix una configuració per les votacions de tipus incentivat.

8.2 Anàlisi

8.2.1 Creador de la votació

Aquest tipus d'usuari no requereix autenticació i ha de poder realitzar els següents casos d'ús:

1. Crear una votació
2. Consultar els resultats de la votació creada per ell

3. Ésser notificat amb l'enllaç dels resultats si ho desitja amb la finalitat de no perdre'l i tenir-lo a la safata d'entrada del correu electrònic

8.2.2 Votant

Aquest tipus d'usuari no requereix autenticació i ha de poder realitzar els següents casos d'ús:

1. Contestar la o les diverses pregunta o preguntes de la votació que haurà posat el creador de la votació a l'hora de crear-la
2. En el cas que el creador de la votació hagi triat que la votació sigui de tipus incentivat podrà animar a votar a algun dels usuaris pertanyents al cens.
3. En el cas que el creador de la votació hagi triat que la votació adjunti enllaç de seguiment, aquest usuari podrà consultar els resultats de la votació a la qual ha participat.
4. Un cop l'usuari hagi votat rebrà un enllaç amb la transacció del vot per a que pugui verificar que s'ha realitzat correctament dins el Blockchain

8.2.3 Administrador

Aquest tipus d'usuari requereix autenticació i ha de poder realitzar els següents casos d'ús:

1. Iniciar sessió
2. Editar la configuració que s'aplicarà a les votacions de tipus incentivat

8.3 Diagrames i fitxes de casos d'ús

En aquest apartat s'exposaran els diagrames de casos d'ús representant els requisits i actors mencionats anteriorment. En primer lloc mostrem el diagrama de casos d'ús general i llavors anirem aprofundint en cada actor.

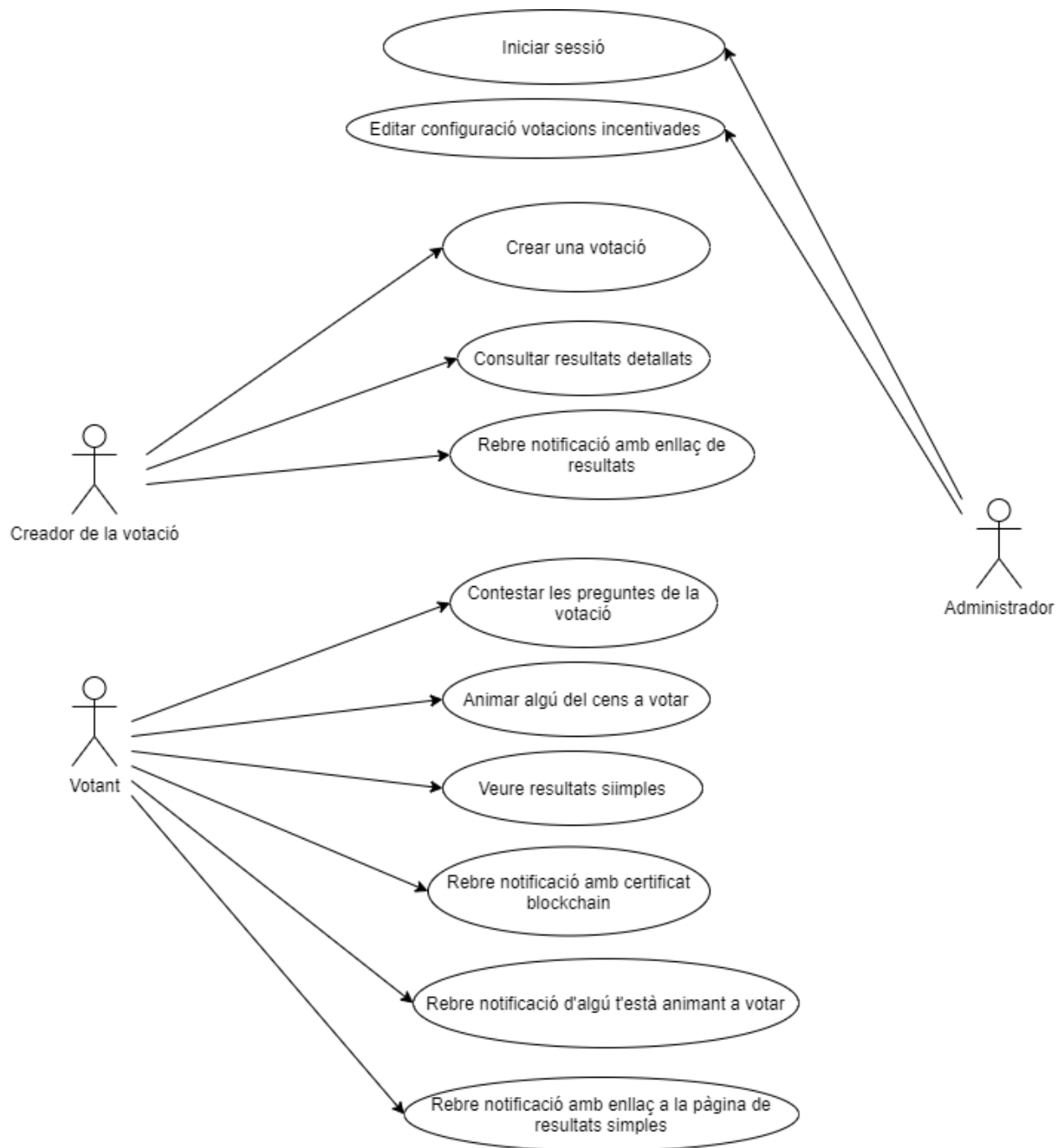


Figura 6: Diagrama de casos d'ús general de l'aplicació

Un cop vist el diagrama general de l'aplicació en els següents apartats anirem aprofundint per cada actor i exposarem les seves funcionalitats amb la seva fitxa de cas d'ús corresponent.

8.3.1 Creador de la votació

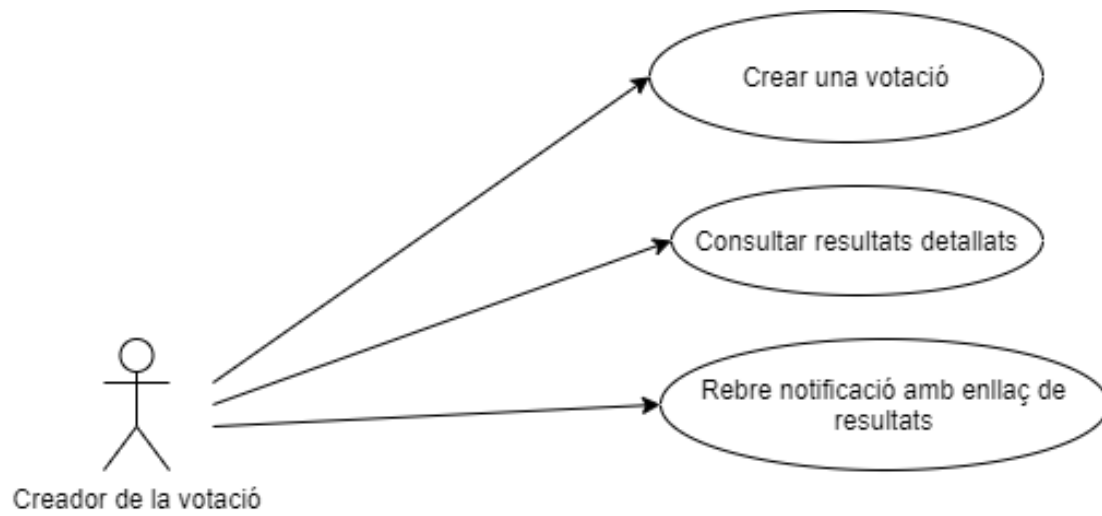


Figura 7: Diagrama de casos enfocant en l'actor creador de la votació

Nom	Crear una votació
Autor	Bogdan Cata
Descripció	L'usuari crea una votació amb els paràmetres que desitja
Actors	Creador de la votació
Precondicions	--
Postcondicions	L'usuari ha creat una votació
Escenari principal	<ol style="list-style-type: none"> 1.- L'usuari selecciona el botó amb la icona de "Crear Votació" que hi ha al centre de la pàgina principal. 2.- Selecciona els participants de la votació entrant els e-mails escrivint o bé pujant un fitxer amb el llistat de e-mails. S'han d'introduir un mínim de dos participants per a passar al pas següent. 3.- Introdueix una o varies preguntes a ser contestades amb les seves opcions de resposta. 4.- Introdueix una data i hora d'inici de la votació i una data i hora de finalització. 5.- Es seleccionarà si vol que la votació sigui de tipus normal o de tipus incentivada, si es vol adjuntar enllaç de resultats perquè els participants puguin veure els resultats de la votació i es seleccionarà l'opció de recordatori (s'aplica en el cas que la votació sigui incentivada, per tal de no "mullar-nos" per algun altre usuari i simplement enviar-li una notificació per a que voti)

Figura 8: Fitxa del cas d'ús crear votació feta per l'actor creador de la votació

Nom	Consultar resultats detallats
Autor	Bogdan Cata
Descripció	Es visualitzen els resultats de manera detallada d'una votació creada per nosaltres.
Actors	Creador de la votació
Precondicions	Haver creat una votació prèviament
Postcondicions	Veure els resultats
Escenari principal	<p>1.- Un cop la votació s'hagi creat satisfactòriament l'usuari tindrà un botó al mig de la pantalla de "Comprovar resultats" el qual el portarà a la pàgina de resultats detallats.</p> <p>2.- Podrà introduir el seu correu electrònic i aquest enllaç li serà enviat a la seva safata d'entrada per tal de no perdre'l.</p> <p>3.- Quan accedeixi a la pàgina es visualitzarà una taula amb els diversos paràmetres de resultats i un gràfic de sectors que representen gràficament aquests resultats.</p>

Figura 9: Fitxa del cas d'ús consultar resultats detallats feta per l'actor creador de la votació

Nom	Rebre notificació amb enllaç de resultats
Autor	Bogdan Cata
Descripció	Es rebrà una notificació de correu electrònic a l'adreça entrada amb l'enllaç dels resultats de la votació
Actors	Creador de la votació
Precondicions	Haver creat una votació i haver introduït una adreça de correu electrònic
Postcondicions	Rebre un correu electrònic amb l'enllaç per accedir als resultats detallats
Escenari principal	<p>1.- Consultar la safata d'entrada de l'adreça de correu electrònic introduïda.</p> <p>2.- Premer el botó del missatge per a accedir a la pàgina de resultats detallats</p>

Figura 10: Fitxa del cas d'ús rebre notificació amb enllaç de resultats feta per l'actor creador de la votació

8.3.2 Votant

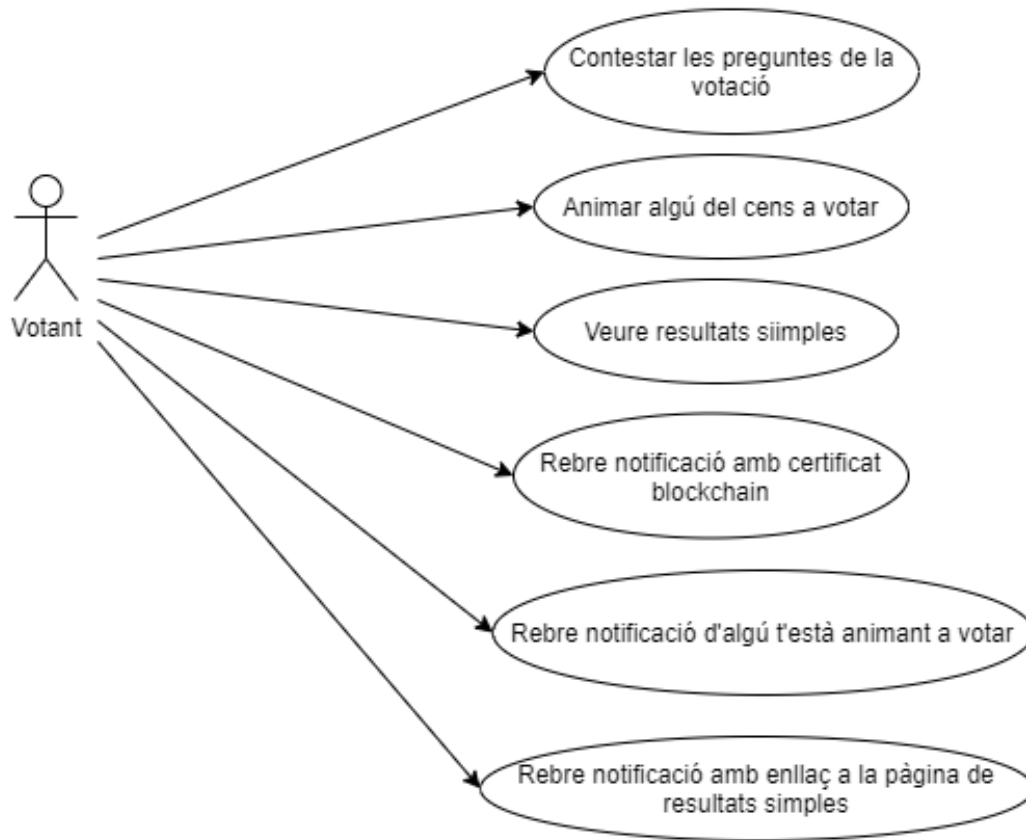


Figura 11: Diagrama de casos d'ús enfocant l'actor votant

Nom	Contestar preguntes de la votació
Autor	Bogdan Cata
Descripció	Respondre les preguntes de la votació en qüestió
Actors	Votant
Precondicions	Pertanyer al cens d'una votació creada i que estigui en el termini entre data d'inici i data de finalització d'aquesta
Postcondicions	Vot executat i comptabilitzat per els resultats
Escenari principal	1.- Accedir a l'enllaç del correu electrònic. 2.- Seleccionar les respostes que es vol per a cada pregunta. 3.- Animar algú a votar si és el cas i es vol

Figura 12: Fitxa del cas d'ús contestar preguntes de la votació feta per l'actor votant

Nom	Animar algú del cens a votar
Autor	Bogdan Cata
Descripció	Animar algú que pertany al cens a votar per la votació en qüestió
Actors	Votant
Precondicions	Pertanyer al cens d'una votació creada de que estigui en el termini entre data d'inici i data de finalització d'aquesta.
Postcondicions	S'ha enviat un correu electrònic a l'usuari que volem animar a votar
Escenari principal	1.- Realitzar el nostre propi vot 2.- Seleccionar el checkbox que volem animar algú a votar. 3.- Seleccionem a quin dels usuaris del cens volem animar a votar. 4.- Seleccionem les opcions de resposta que creiem que contestaria l'usuari al que volem animar a votar per a que si està d'acord només hagi de prémer un botó

Figura 13: Fitxa del cas d'ús animar algú del cens a votar feta per l'actor votant

Nom	Veure resultats simples
Autor	Bogdan Cata
Descripció	Es visualitzen els resultats d'una votació a la qual nosaltres hem participat
Actors	Votant
Precondicions	Haver participat a la votació en qüestió i que l'usuari creador d'aquesta hagi escollit l'opció d'afegir enllaç de seguiment
Postcondicions	Veure els resultats de la votació a la qual hem participat
Escenari principal	1.- Accedir a l'enllaç del correu electrònic 2.- Veure la pàgina de resultats simples on es mostra una taula amb les diferents opcions votades i un gràfic de sectors representant gràficament aquestes dades.

Figura 14: Fitxa del cas d'ús veure resultats simples feta per l'actor votant

Nom	Rebre notificació amb certificat blockchain
Autor	Bogdan Cata
Descripció	Es rep una notificació de correu electrònic amb un enllaç a l'explorador de Blockchain per a veure que la transacció del nostre vot s'ha emès correctament.
Actors	Votant
Precondicions	Haver realitzat un vot
Postcondicions	Veure a l'explorador de Blockchain la transacció del nostre vot
Escenari principal	1.- Accedir a l'enllaç del correu electrònic 2.- Veure a l'explorador de Blockchain Alastria informació detallada sobre la transacció del nostre vot.

Figura 15: Fitxa del cas d'ús rebre notificació amb certificat blockchain feta per l'actor votant

Nom	Rebre notificació algú t'està animant a votar
Autor	Bogdan Cata
Descripció	Es rep una notificació de correu electrònic informant que algú t'està animant a votar a una votació determinada.
Actors	Votant
Precondicions	El tipus de la votació sigui incentivada i algun altre usuari del cens hagi emès un vot i hagi escollit l'opció d'animar-te a votar.
Postcondicions	Poder accedir a la pàgina per a confirmar el vot o bé canviar alguna de les respostes i emetre el vot
Escenari principal	1.- Accedir a l'enllaç del correu electrònic 2.- Confirmar el vot si estem d'acord amb les opcions marcades o bé canviar-ne les que desitgem. 3.- Emetre el vot

Figura 16: Fitxa del cas d'ús rebre notificació algú t'està animant a votar feta per l'actor votant

Nom	Rebre notificació amb enllaç a la pàgina de resultats simples
Autor	Bogdan Cata
Descripció	Es rep una notificació de correu electrònic amb l'enllaç als resultats de la votació que hem participat
Actors	Votant
Precondicions	Haver participat a la votació en qüestió i que l'usuari creador d'aquesta hagi escollit l'opció d'afegir enllaç de seguiment
Postcondicions	Obtenir l'enllaç per a veure els resultats simples
Escenari principal	1.- Accedir a l'enllaç del correu electrònic 2.- Veure els resultats simples

Figura 17: Fitxa del cas d'ús rebre notificació amb enllaç a la pàgina resultats simples feta per l'actor votant

8.3.3 Administrador

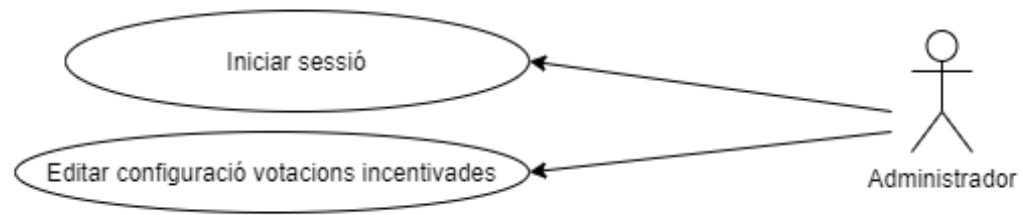


Figura 18: Diagrama de casos d'ús enfocant en l'actor administrador

Nom	Iniciar Sessió
Autor	Bogdan Cata
Descripció	L'usuari inicia sessió com a administrador
Actors	Administrador
Precondicions	L'usuari ha d'estar creat a la base de dades per nosaltres de manera manual
Postcondicions	L'usuari està autenticat
Escenari principal	1.- Accedeix a la pàgina d'administrador afegint /admin a la URL base. 2.- Introdueix el seu correu electrònic i la seva contrasenya. 3.- Si aquests són correctes s'entra a la pàgina d'administrador

Figura 19: Fitxa del cas d'ús iniciar sessió feta per l'actor administrador

Nom	Editar configuració votacions incentivades
Autor	Bogdan Cata
Descripció	Pàgina per editar la configuració que s'aplica a les votacions de tipus incentivat
Actors	Administrador
Precondicions	Estar autenticat com a administrador
Postcondicions	Veure i/o editar la configuració
Escenari principal	<ol style="list-style-type: none"> 1.- L'usuari veu els percentatges de configuració que s'apliquen a les votacions de tipus incentivat. 2.- Si aquest ho desitja els pot editar. 3.- Un cop editats ha de prémer "Guardar" i a partir d'aquell moment, totes les votacions que es creïn de tipus incentivat se'ls hi aplicarà aquesta configuració.

Figura 20: Fitxa del cas d'ús editar configuració feta per l'actor administrador

8.4 Models de la base de dades

Un cop hem analitzat els requisits del sistema i hem vist les fitxes de casos d'ús, definirem el model de dades. La creació de la base de dades s'ha realitzat amb [Mongoose](#), una biblioteca de JavaScript que permet definir esquemes amb dades tipades. Un cop es defineix un esquema, [Mongoose](#) permet crear un model basat en aquest esquema i a partir d'aquest crear un document amb aquest esquema sempre i quan es desitgi.

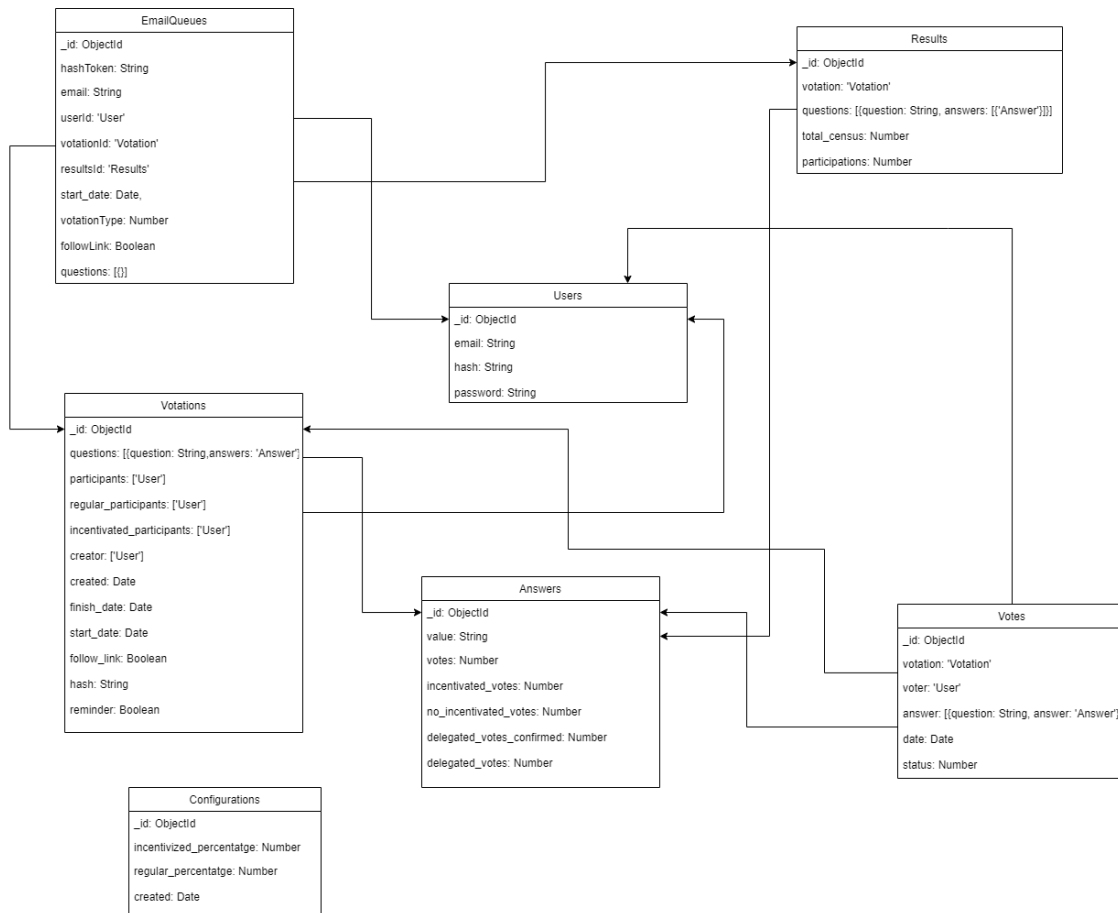


Figura 21: Diagrama de la Base de Dades

8.4.1 Answers

```

1 // load the things we need
2 const mongoose = require('mongoose');
3
4 // define the schema for our model
5 const Schema = mongoose.Schema(
6   {
7     value: String,
8     votes: {type: Number, default: 0}, //votes for all
9     incentivated_votes: {type: Number, default: 0},
10    no_incentivated_votes: {type: Number, default: 0},
11    delegated_votes_confirmed: {type: Number, default: 0}, //confirmed
12    delegated_votes: {type: Number, default: 0} //send
13  }
14 );
15
16 module.exports = mongoose.model('Answer', Schema);
  
```

Listing 1: Model de answers.js

Aquest és el model de answers on emmagatzem els comptadors de vots de la resposta en qüestió.

Té com a atributs en primer lloc *value* que és de tipus *String* i és on hi tenim el valor de la opció de resposta.

Tenim *votes* de tipus *Number* que té un valor per defecte de 0 i és on s'emmagatzema el nombre de vots d'aquesta resposta.

Incentivated_votes i *no_incentivated_votes* de tipus *Number* i valor per defecte 0 on hi guardem els vots que s'han realitzat com a resultat que algú ha animat a votar a l'usuari respectiu i el complementari d'aquest respectivament, o sigui, el nombre de respostes que s'han rebut de accedint des de l'enllaç de la votació.

Per últim tenim *delegated_votes* i *delegated_votes_confirmed* també de tipus *Number* i valor per defecte 0, en aquests atributs emmagatzem el nombre d'animacions a votar que s'han enviat en el cas de *delegated_votes* i en el cas que aquesta animació ha set efectiva i l'usuari al qual s'ha animat a votar ha realitzat el seu vot es comptabilitza a *delegated_votes_confirmed*.

8.4.2 EmailQueues

```
1 // load the things we need
2 const mongoose = require('mongoose');
3
4 // define the schema for our model
5 const Schema = mongoose.Schema(
6   {
7     hashToken: {type: String, default: ''},
8     email: {type: String, default: ''},
9     userId: {type: mongoose.Schema.Types.ObjectId, ref: 'User'},
10    votationId: {type: mongoose.Schema.Types.ObjectId, ref: 'Votation'},
11    resultsId: {type: mongoose.Schema.Types.ObjectId, ref: 'Results'},
12    start_date: { type: Date, default: Date.now },
13    votationType: { type: Number, default: 0 },
14    followLink: { type: Boolean, default: false },
15    questions: [{}],
16  }
17 );
18
19 module.exports = mongoose.model('EmailQueue', Schema);
```

Listing 2: Model de emailQueue.js

Aquest és el model de emailqueue on es guarden tots els e-mails que s'han d'enviar per una votació que comença en un futur. Bàsicament aquesta col·lecció s'emplena quan es crea una votació que comença en un futur i no just ara i via un cron (pro-

cés que s'executa cada X temps, en el nostre cas cada minut) es consulta aquesta col·lecció i s'envia el correu conforme la votació ja està disponible per a votar.

En aquesta col·lecció hi tenim *hashToken* que és de tipus *String* on hi tenim un token el qual és únic per cada votació-email enviat. Tenim aquest atribut bàsicament perquè nosaltres mai puguem rastrejar i saber què ha votat qui, perquè, com bé sabem, les votacions han d'esser anònimes.

A continuació, hi tenim els *email* de tipus *String* que és l'adreça de correu electrònic a la qual hem d'enviar la notificació.

UserId que és una referència a l'identificador de l'usuari en qüestió.

VotationId que és una a l'identificador referència de la votació que pertany aquest e-mail que s'ha d'enviar.

ResultsId que és una referència a l'identificador de resultats.

Start_date que és la data a partir de la qual s'ha d'enviar aquest e-mail.

VotationType que és el tipus de la votació, si és incentivada o normal.

FollowLink de tipus *booleà* que és el flag si s'ha d'enviar enllaç de resultats o no.

Finalment *questions* que és una llista de les preguntes de la votació.

8.4.3 Results

```
1 // load the things we need
2 const mongoose = require('mongoose');
3
4 // define the schema for our model
5 const Schema = mongoose.Schema(
6   {
7     votation: {type: mongoose.Schema.Types.ObjectId, ref: 'Votation'},
8     questions: [{question: {type: String, default: ""}, answers: [{type: mongoose.
9       Schema.Types.ObjectId, ref: 'Answer'}]}],
10    total_census: Number,
11    participations: Number,
12  }
13 );
14 module.exports = mongoose.model('Result', Schema);
```

Listing 3: Model de results.js

Aquest és el model de results, on hi guardem tots els resultats de les votacions.

En primer lloc tenim *votation* la qual fa referència a la votació a la qual pertanyen aquests resultats.

Questions que és una llista de les diferents preguntes que té la votació amb les seves respectives respostes.

Total_census on hi tenim el nombre total de participants que hi ha al cens.

Per últim tenim *participations* que és el nombre de participacions que s'han fet a la votació en qüestió.

8.4.4 Users

```
1 // load the things we need
2 const mongoose = require('mongoose');
3 const Constants = require('../config/constants');
4
5 // define the schema for our user model
6 const Schema = mongoose.Schema(
7   {
8     email: String,
9     role: Number,
10    hash: String,
11    password: String
12  }
13 );
14
15 // create the model for users and expose it to our app
16 module.exports = mongoose.model('User', Schema);
17
18 // Constants
19 module.exports.ROLE_ADMIN = Constants.USER.ROLE_ADMIN;
20 module.exports.ROLE_USER = Constants.USER.ROLE_USER;
```

Listing 4: Model de users.js

En aquesta col·lecció hi guardem els diferents usuaris que tenen com a atributs els següents. En primer lloc tenen *email* que és el seu correu electrònic.

Role per a saber si és un usuari normal o un usuari administrador.

Hash que és el hash identificatiu d'aquest usuari del Blockchain.

Password que és la contrasenya encriptada en el cas que aquest usuari sigui administrador.

8.4.5 Votations

```
1 // load the things we need
2 const mongoose = require('mongoose');
3 const Constants = require('../config/constants');
4
5 // define the schema for our model
6 const Schema = mongoose.Schema(
7   {
8     questions: [{question: {type: String, default: ""}, answers: [{type: mongoose.
9       Schema.Types.ObjectId, ref: 'Answer'}]}],
10    participants: [{type: mongoose.Schema.Types.ObjectId, ref: 'User'}],
11    regular_participants: [{type: mongoose.Schema.Types.ObjectId, ref: 'User'}],
12    incentivated_participants: [{type: mongoose.Schema.Types.ObjectId, ref: 'User'
13      }],
14    creator: {type: mongoose.Schema.Types.ObjectId, ref: 'User'},
15    created: { type: Date, default: Date.now },
16    finish_date: { type: Date, default: Date.now },
17    start_date: { type: Date, default: Date.now },
18    follow_link: { type: Boolean, default: true },
19    type: {type: Number, default: Constants.VOTATION.REGULAR_VOTATION},
20    hash: String,
21    reminder: { type: Boolean, default: false },
22  }
23 );
24
25 module.exports = mongoose.model('Votation', Schema);
26
27 // Constants
28 module.exports.REGULAR_VOTATION = Constants.VOTATION.REGULAR_VOTATION;
29 module.exports.ENCOURAGED_VOTATION = Constants.VOTATION.ENCOURAGED_VOTATION;
```

En aquesta col·lecció hi guardem les diferents votacions amb els seus respectius paràmetres.

En primer lloc tenim *questions* que és una array on els seus elements són objectes compostats per, en primer lloc, *question* que és el valor de la pregunta i llavors *answers* que és una array dels identificadors de les diferents opcions de resposta associades a aquesta pregunta.

Participants que és la llista d'usuaris que pertanyen al cens.

Regular_participants que és la llista d'usuaris per la qual en el cas que la votació sigui de tipus incentivat per els que estan en aquesta llista és una votació normal.

Incentivated_participants és la llista d'usuaris per la qual la votació que és de tipus incentivat realment és de tipus incentivat.

Creator és l'identificador de l'usuari que ha creat la votació.

Created és la data i hora a la que aquesta votació ha sigut creada.

Start_date és la data d'inici de la votació.

Finish_date és la data de finalització de la votació.

Follow_link és el flag que ens diu si es vol enllaç de seguiment per els resultats o no.

Type és el tipus de la votació, si és incentivada o normal.

Hash és el hash identificatiu del Blockchain d'aquesta votació.

Reminder és el flag que ens diu si es vol la opció de recordatori o no.

8.4.6 Votes

```
1 // load the things we need
2 const mongoose = require('mongoose');
3 const Constants = require('../config/constants');
4
5 // define the schema for our model
6 const Schema = mongoose.Schema(
7   {
8     votation: {type: mongoose.Schema.Types.ObjectId, ref: 'Votation'},
9     voter: {type: mongoose.Schema.Types.ObjectId, ref: 'User'},
10    answer: [{question: {type: String, default: ""}, answer: {type: mongoose.
11      Schema.Types.ObjectId, ref: 'Answer'}}],
12    date: { type: Date, default: Date.now },
13    status: {type: Number, default: 0},
14  }
15 );
16 module.exports = mongoose.model('Vote', Schema);
17
18 // Constants
19 module.exports.STATUS_PENDING = Constants.VOTE_STATUS.PENDING;
20 module.exports.STATUS_CONFIRMED = Constants.VOTE_STATUS.CONFIRMED;
21 module.exports.STATUS_DENIED = Constants.VOTE_STATUS.DENIED;
```

En aquesta col·lecció hi guardem els diferents vots amb els seus respectius paràmetres.

En primer lloc tenim *vation* que és una referència a l'identificador de la votació a la qual pertany aquest vot.

Voter que és la referència a l'identificador de l'usuari que ha emès aquest vot.

Answer que és una llista d'objectes on com a primer camp hi tenim el valor de la pregunta i en segon lloc una referència a la resposta contestada, cal remarcar que inicialment es guardaven els vots a la base de dades però des de la integració del Blockchain aquesta col·lecció només s'utilitza en el cas dels vots delegats, o sigui, quan algú fa un vot delegat amb certs valors a les respostes, es crea un vot temporal a la base de dades i llavors si aquest usuari acaba confirmant el vot es guarda al Blockchain, fonamentalment aquest vot és temporal i ens serveix de pasarel·la.

Date que és la data a la qual aquest vot ha estat emès.

Status que és l'estat del vot, bàsicament, quan un usuari confirma el seu vot delegat aquest atribut canvia de valor, però no canviem les respostes en cas que ell les hagi canviat, ja que, si les canviaríem sabríem el què ha votat, en canvi, al no actualitzar-ho la informació de la base de dades i el que finalment està guardat al Blockchain pot ser diferent.

8.4.7 Configurations

```
1 // load the things we need
2 const mongoose = require('mongoose');
3
4 // define the schema for our model
5 const Schema = mongoose.Schema(
6   {
7     incentivized_percentage: {type: Number, default: 50},
8     regular_percentage: {type: Number, default: 50},
9     created: {type: Date, default: Date.now()}
10  }
11 );
12
13 module.exports = mongoose.model('Configuration', Schema);
```

En aquesta col·lecció hi guardem les configuracions que entra l'administrador des de la seva pàgina.

Bàsicament conté els dos percentatges en els quals s'ha de dividir el cens total i la data de creació d'aquesta, donat que, com bé he exposat anteriorment, s'agafa la configuració més recent.

8.5 Model de dades Smart Contract

```
1 pragma solidity ^0.8.2;
```

```

2
3 contract VoteVote {
4     address owner;
5
6     struct Cens{
7         bool exist;
8     }
9     struct CensVote{
10        bool exist;
11        bool voted;
12    }
13    struct Candidate{
14        bool exist;
15        uint32 votes;
16    }
17    struct Question{
18        bool exist;
19        mapping (address => Candidate) candidates;
20    }
21    struct Votation{
22        bool exist;
23        uint256 start;
24        uint256 end;
25        uint48 creationDate;
26        uint32 totalCensus;
27        uint32 totalVots;
28        mapping (address => Cens) census;
29        mapping (address => CensVote) censusVote;
30        mapping (address => Question) questions;
31    }
32    mapping (address => Votation) votations;
33
34 }

```

Listing 5: Part del fitxer de VoteVote.sol on hi consta el model de dades del Smart Contract

8.6 Disseny de les interfícies d'usuari

Un cop disposem del llistat de requisits, els diagrames i fitxes de casos d'ús i els models que tindrem a la base de dades, cal dissenyar les pantalles que veurà l'usuari i les plantilles de correu electrònic per a les notificacions.

8.6.1 Home

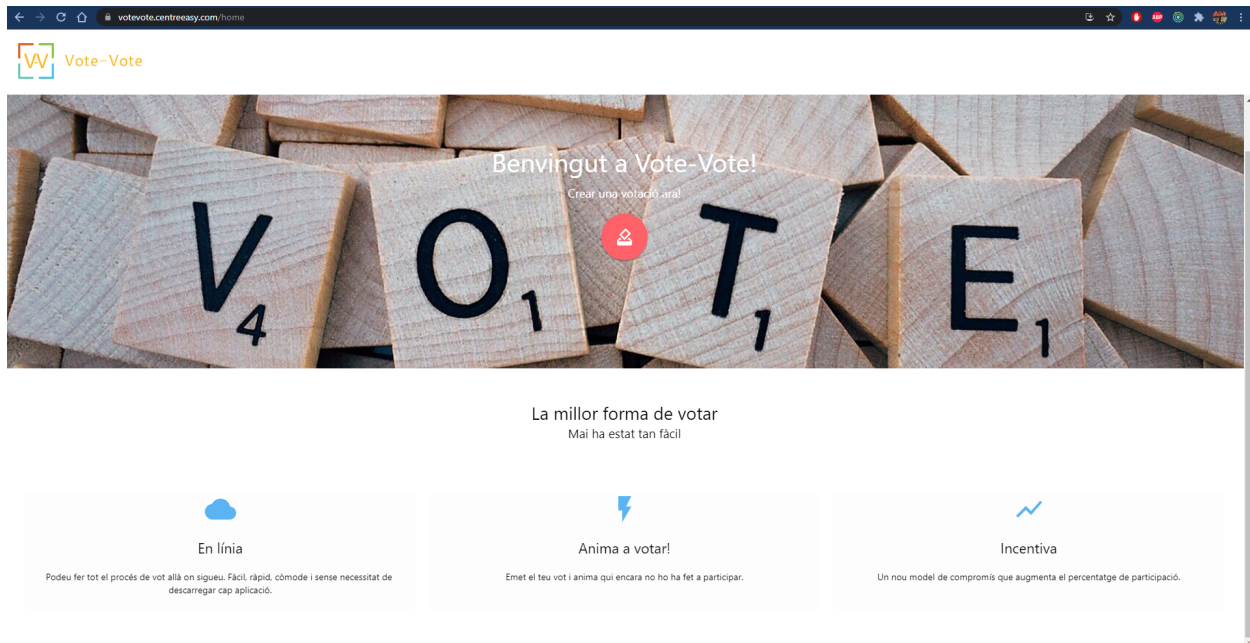


Figura 22: Captura de pantalla de la pantalla de Home de l'aplicació

Pantalla principal on veiem una petita informació sobre la aplicació amb un botó central que ens redirecciona a la pantalla de *Crear Votació*

8.6.2 Crear votació

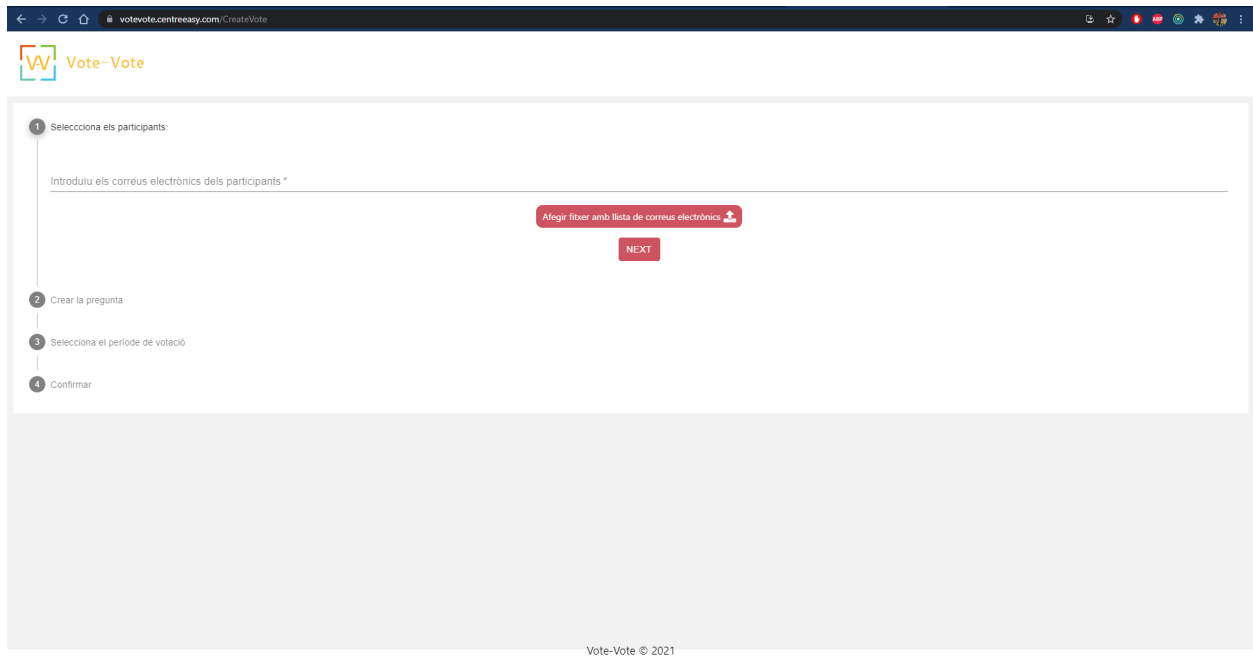


Figura 23: Captura de pantalla de la pàgina de crear la votació

La pantalla de *Crear Votació* consta de 4 passes que s’han de completar per tal de crear la votació.

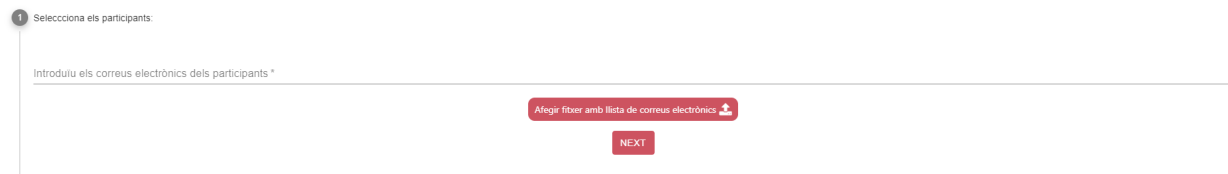


Figura 24: Captura de pantalla del primer pas de la pàgina de crear la votació

En primer lloc tenim la part d’introducció del cens, on hem d’entrar els correus electrònics dels participants a la votació que crearem, ho podem fer de manera manual o bé pujant un fitxer .txt o .csv amb el llistat d’adreces.



Figura 25: Captura de pantalla del segon pas de la pàgina de crear la votació afegint la primera pregunta

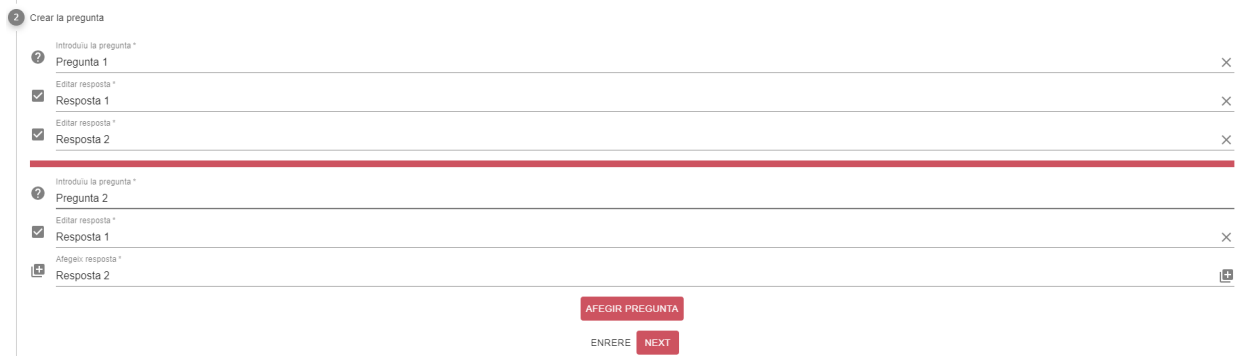


Figura 26: Captura de pantalla del segon pas de la pàgina de crear la votació afegint una segona pregunta

El segon pas consisteix en entrar la o les preguntes de la votació amb les seves respectives opcions de resposta.



Figura 27: Captura de pantalla del tercer pas de la pàgina de crear la votació

El tercer pas consisteix en seleccionar la data d'inici i data de finalització d'aquesta votació.



Figura 28: Captura de pantalla del quart i últim pas de la pàgina de crear la votació

Per últim, tenim un petit resum del que hem anat creat pas a pas per a comprovar que no ens hem deixat res i que està tot correcte. Un cop confirmat tot, la votació serà creada, els usuaris pertanyents al cens seran notificats via correu electrònic i nosaltres com a creadors de la votació veurem la següent pantalla.

8.7 Pantalla post-creació votació

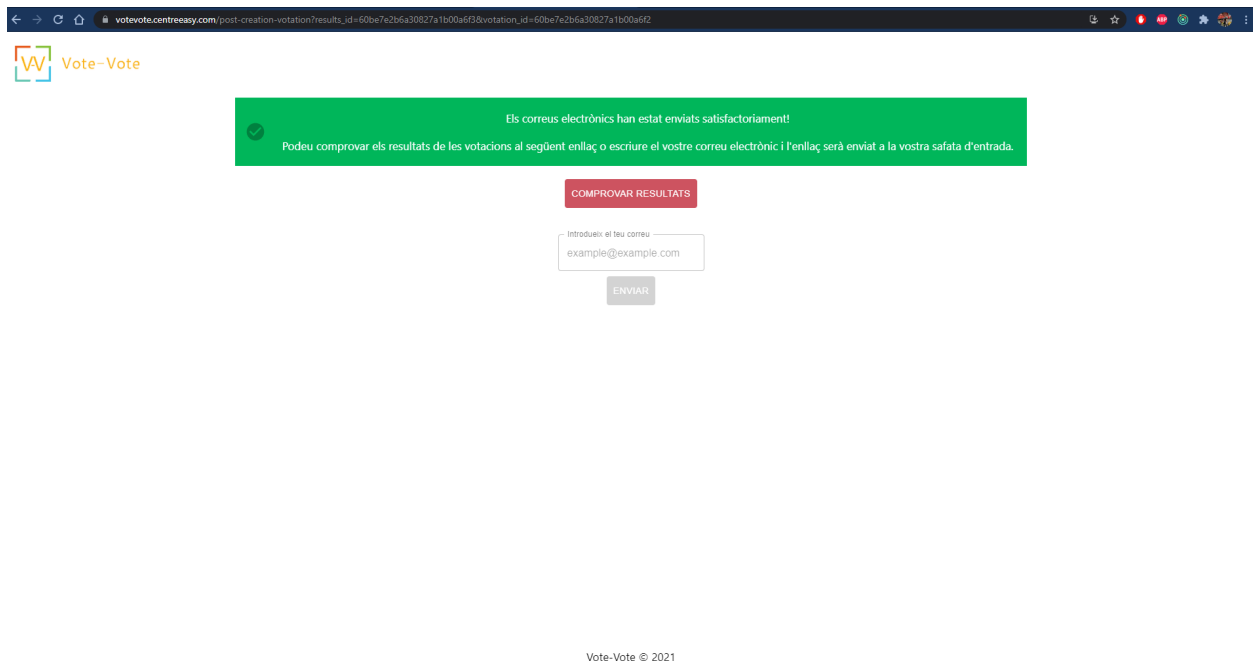


Figura 29: Captura de pantalla de la pantalla que veiem un cop hem creat la votació

Aquesta pantalla consta d'un botó que ens porta directament a la pantalla de resultat i una entrada de text per a entrar la nostra adreça de correu electrònic i aquest enllaç ens sigui enviat via un correu electrònic. Aquest correu té el següent aspecte.

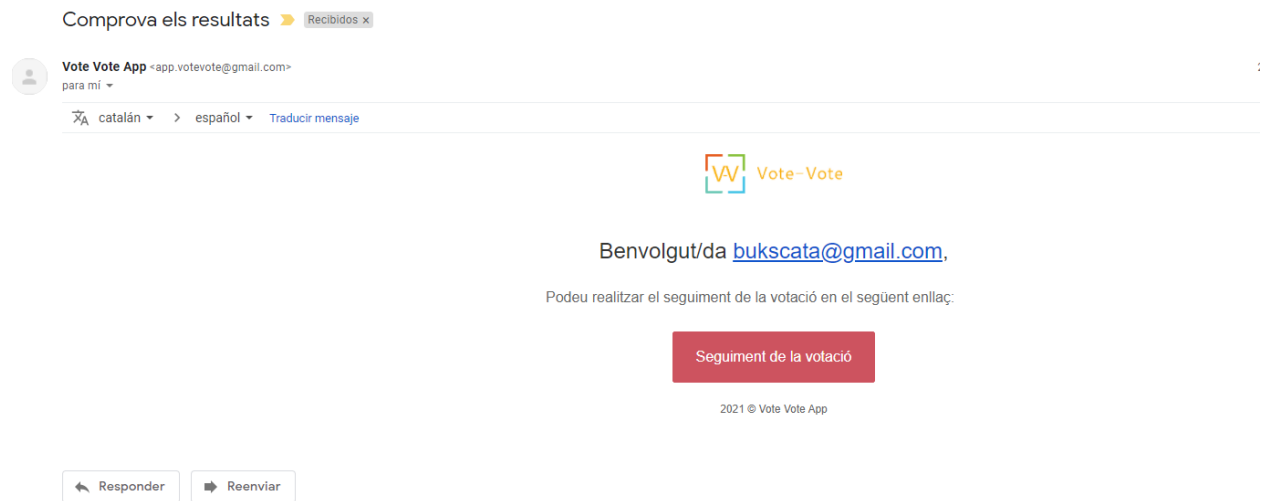


Figura 30: Captura de pantalla del correu electrònic que s'envia quan l'usuari creador introdueix el seu correu electrònic per a rebre l'enllaç de resultats per aquesta via

8.7.1 Plantilla correu votació creada

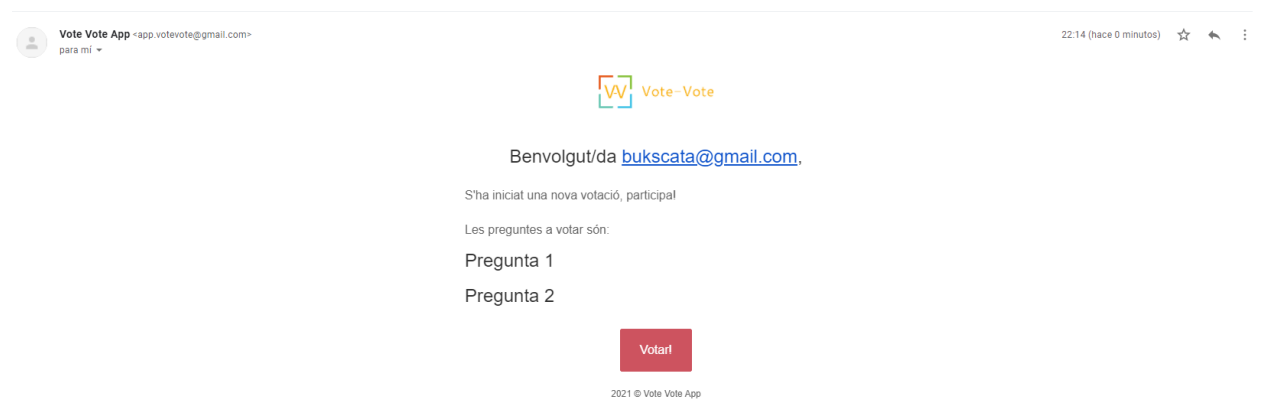


Figura 31: Captura de pantalla del correu electrònic que s'envia quan s'inicia una votació normal



Benvolgut/da bogdancata@easyinnova.com,

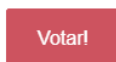
S'ha iniciat una nova votació, participa!

Anima altres participants a votar amb el nostre sistema d'incentivació de vot

Les preguntes a votar són:

Pregunta 1

Pregunta 2



2021 © Vote Vote App

Figura 32: Captura de pantalla del correu electrònic que s'envia quan s'inicia una votació de tipus incentivat

8.7.2 Votar



El teu correu electrònic
bukscata@gmail.com

Pregunta 1

Resposta 1

Resposta 2

Vot en blanc

Pregunta 2

Resposta 1

Resposta 2

Vot en blanc

VOTE

Figura 33: Captura de pantalla de la pantalla de votar per una votació normal

Pantalla de votació, la qual s'accedeix via el correu electrònic de votació creada en la qual veiem el llistat de preguntes amb les seves opcions de resposta.

El teu correu electrònic
bogdancata@easyinnova.com

Pregunta 1

Resposta 1

Resposta 2

Vot en blanc

Pregunta 2

Resposta 1

Resposta 2

Vot en blanc

Vota per algú altre

VOTE

Figura 34: Captura de pantalla de la pantalla de votar per una votació de tipus incentivat part superior

Seleccioneu un correu electrònic del cens

Other email
bukscata@mailinator.com

Recordatori

Pregunta 1

Resposta 1

Resposta 2

Vot en blanc

Pregunta 2

Resposta 1

Resposta 2

Vot en blanc

VOTE

Figura 35: Captura de pantalla de la pantalla de votar per una votació de tipus incentivat part inferior

8.7.3 Plantilla correu animar a votar

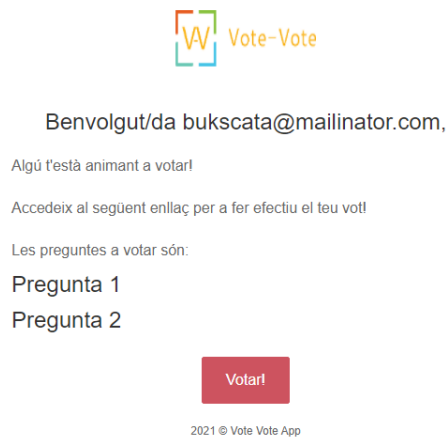


Figura 36: Captura de pantalla del correu electrònic que s'envia quan algú t'ha animat a votar

Pantalla que veuríem en el cas que la votació sigui de tipus incentivat.

8.7.4 Plantilla correu certificat blockchain

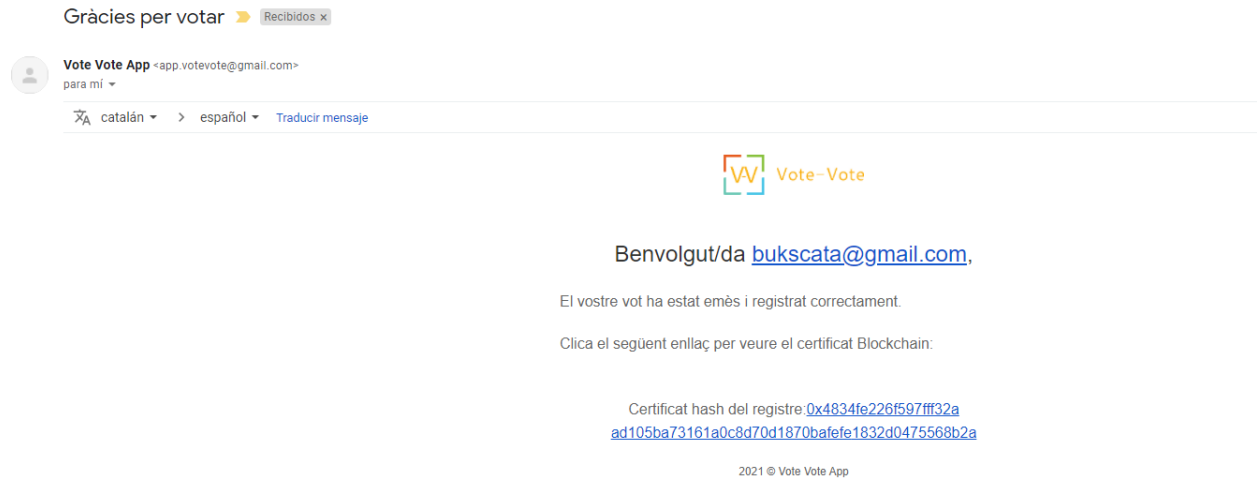




Figura 37: Captura de pantalla del correu electrònic que s'ha comptabilitzat un vot correctament al Blockchain

Un cop hem emès el nostre vot, rebrem un correu electrònic de confirmació conforme el nostre vot s'ha emès correctament amb un enllaç a l'explorador de Blockchain per a veure detalls sobre la transacció que li correspon

8.7.5 Resultats públics

 Vote-Vote

 Resultats de la votació pregunta 1

Pregunta: Pregunta 1

Respostes	Vots
Resposta 1	0
Resposta 2	1
Vot en blanc	0

Figura 38: Captura de pantalla de la primera secció de la primera pregunta de la pàgina de resultats públics

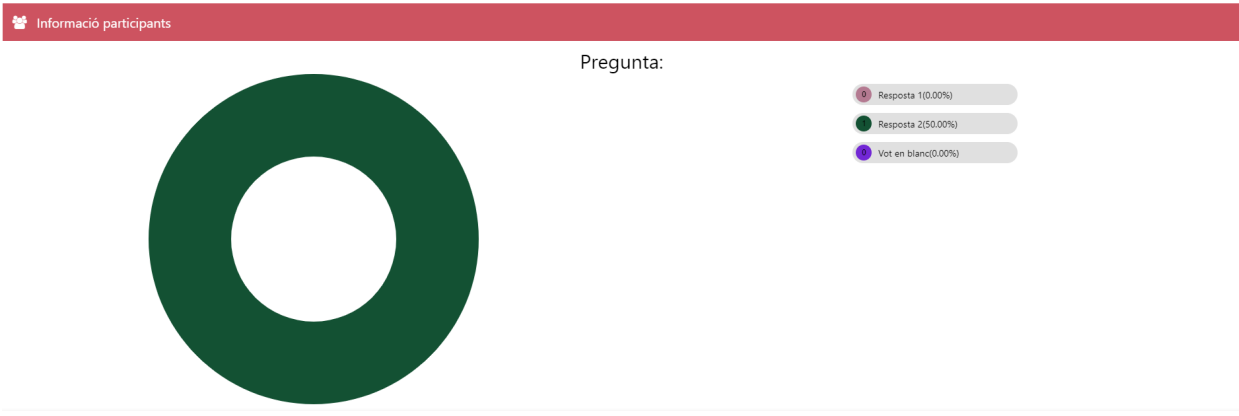


Figura 39: Captura de pantalla de la primera secció de la primera pregunta de la pàgina de resultats públics

Resultats de la votació pregunta 2

Pregunta: Pregunta 2

Respostes	Vots
Resposta 1	1
Resposta 2	0
Vot en blanc	0

Figura 40: Captura de pantalla de la primera secció de la primera pregunta de la pàgina de resultats públics

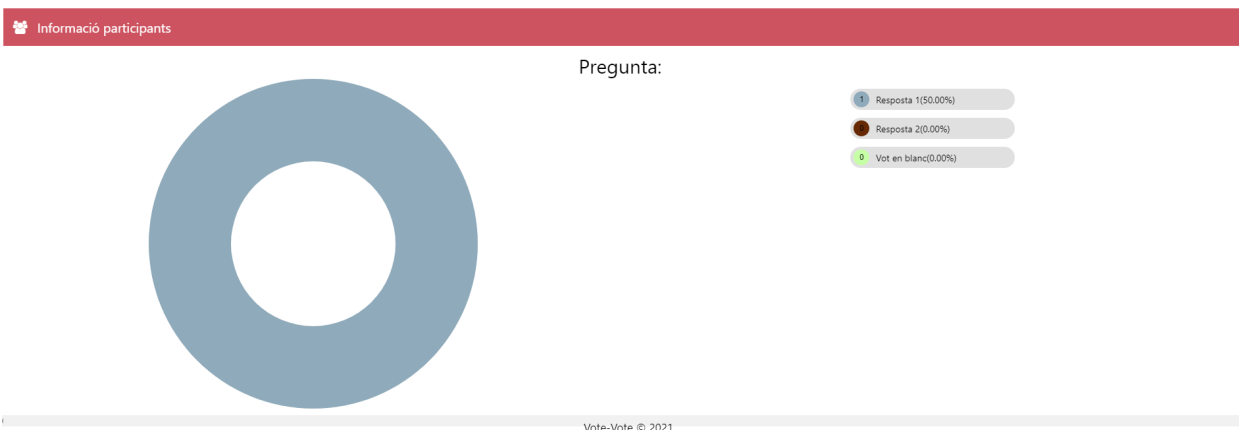


Figura 41: Captura de pantalla de la primera secció de la primera pregunta de la pàgina de resultats públics

Pàgina que veurien els usuaris pertanyents al cens amb el resultat de la votació en el cas que el creador de la votació ha activat el flag de *Enllaç de seguiment*

8.7.6 Resultats creador votació



Resultats de la votació pregunta 1

Pregunta: Pregunta 1
Participants pendents: 2
Percentatge participació: 0.00%

Resposta	Vots	Vots incentivats	Vots incentivats confirmats	% Vots incentivats confirmats
Resposta 1	0	0	0	0
Resposta 2	0	0	0	0
Vot en blanc	0	0	0	0
TOTAL	0	0	0	0

Figura 42: Captura de pantalla de la primera secció de la primera pregunta de la pàgina de resultats

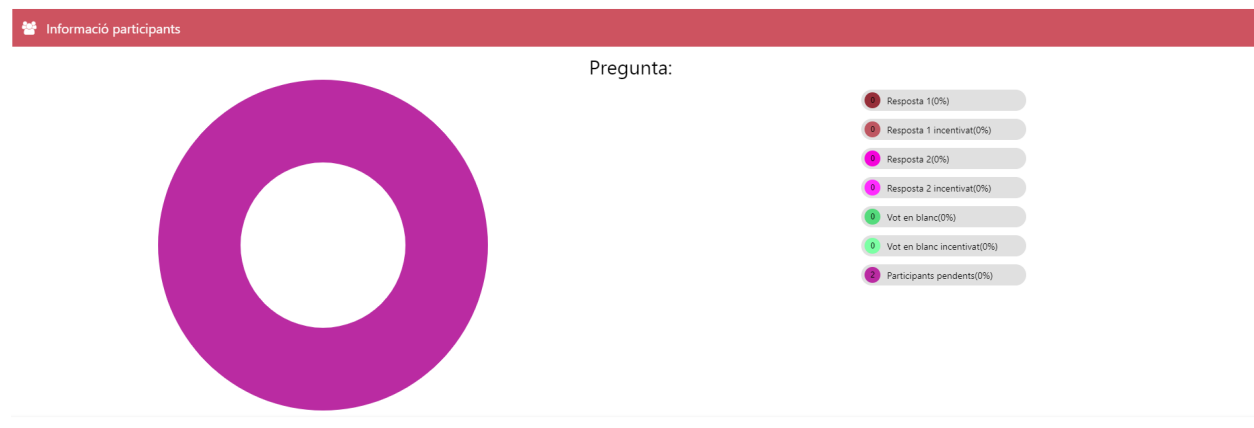


Figura 43: Captura de pantalla de la segona secció de la primera pregunta de la pàgina de resultats

Resultats de la votació pregunta 2

Pregunta: Pregunta 2
Participants pendents: 2
Percentatge participació: 0.00%

Resposta	Vots	Vots incentivats	Vots incentivats confirmats	% Vots incentivats confirmats
Resposta 1	0	0	0	0
Resposta 2	0	0	0	0
Vot en blanc	0	0	0	0
TOTAL	0	0	0	0

Figura 44: Captura de pantalla de la primera secció de la segona pregunta de la pàgina de resultats

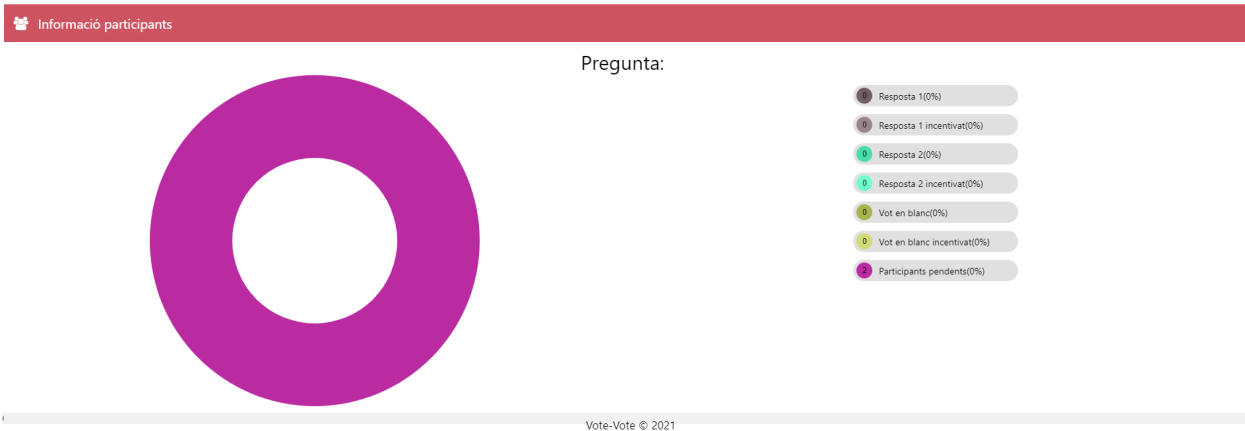
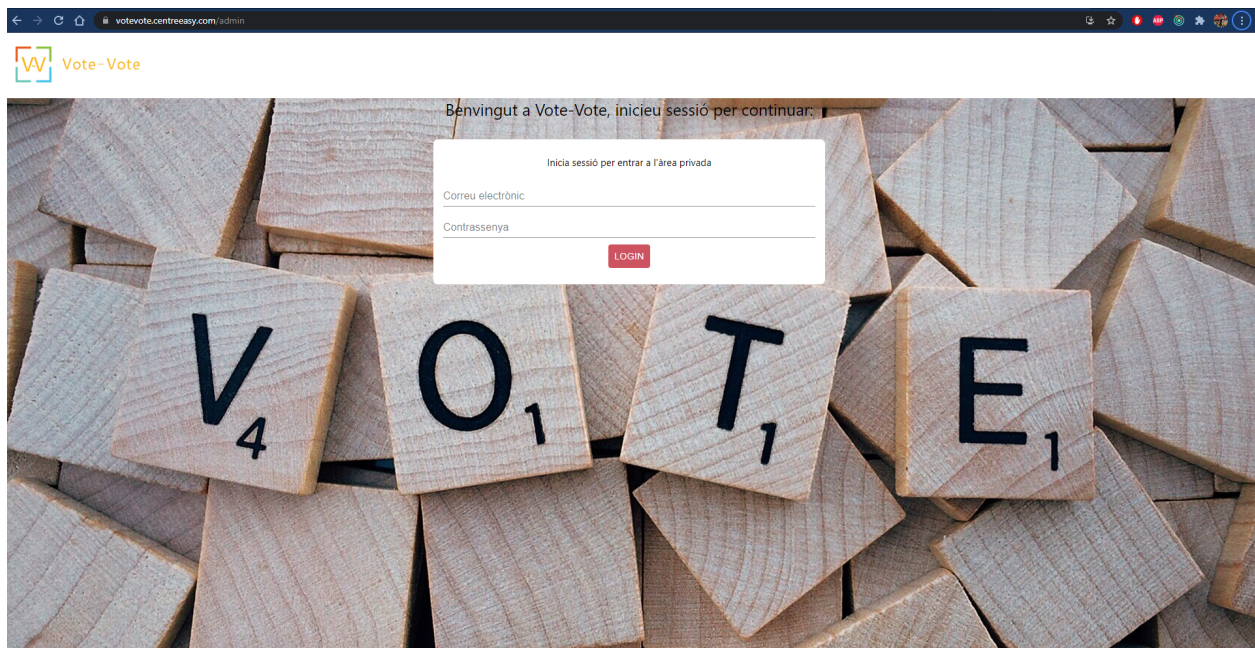


Figura 45: Captura de pantalla de la segona secció de la segona pregunta de la pàgina de resultats

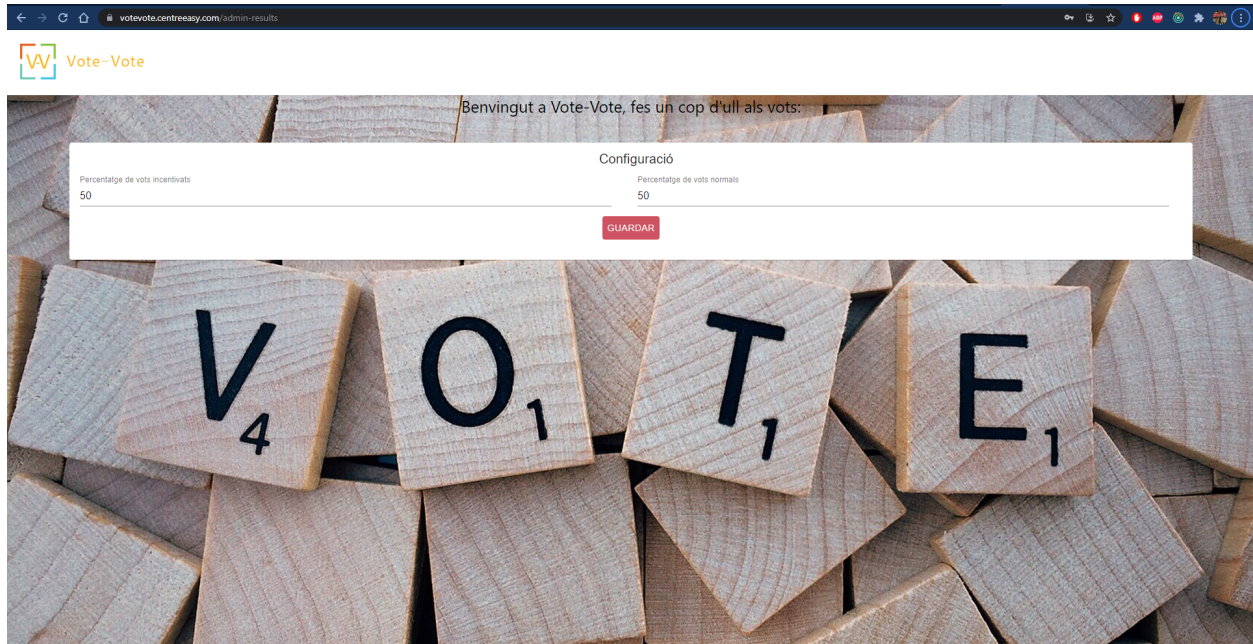
Pàgina que veuria el creador de la votació amb els resultats d'aquesta.

8.7.7 Login



Pantalla de login la qual s'accedeix entrant a la següent URL
<https://votevote.centreeasy.com/admin>

8.7.8 Pantalla editar configuració



Pantalla que veuria l'administrador en el cas que s'hagi autenticat correctament.

9 Implementació i proves

Un cop hem realitzat tot l'anàlisi del sistema, tenim el disseny en ment i tenim clar com ha de funcionar tot cal començar a implementar l'aplicació i realitzar les proves corresponents per tal de verificar el seu correcte funcionament.

Alhora de desenvolupar per provar els diversos casos d'ús s'ha fet en local, utilitzant la mateixa màquina tant per client com per servidor, en el cas particular el client corria al port 19006 per tant, al accedir a <http://localhost:8080> teníem corrent al client i el servidor corria al port 3032, per tant, si féssim crides amb un client extern, per exemple, un [Postman](#) a la url <http://localhost:3032/api/Ruta Web Service> doncs hem executat i debugat per anar provant els web services però ara, ho volem provar de manera pública per tant, haurem de configurar l'entorn de producció i per a fer-ho hem utilitzat i configurant el següent.

Abans de tot mencionar que a la màquina on ho hem allotjat té com a sistema operatiu un Ubuntu 18 així que ho explicarem tot per aquest cas concret.

9.1 Primers passos

En primer lloc hem de clonar el repository de git on tenim tot el codi font de l'aplicació, en el cas que no tinguem instal·lat el [Git](#) ho podem fer amb la següent comanda.

```
1 sudo apt install git
```

Per a clonar el repositori utilitzarem la següent comanda:

```
1 git clone git@bitbucket.org:centreeasy/vote-vote-react.git
```

Un cop clonat el repositori entrarem dins la carpeta creada i ens baixarem els últims canvis en el cas que haguem fet algun canvi al codi font des del clonatge del repositori, per a fer-ho utilitzarem la següent comanda:

```
1 git pull
```

Ara estem segurs que tenim la ultima versió del nostre codi.

9.2 Pm2

En primer lloc necessitem un administrador de processos, per tant, s'ha fet una petita recerca i s'ha escollit [PM2](#) ja que, és el que estic més familiaritzat a utilitzar i el més utilitzat per a projectes *Node.JS*. Per a instal·lar-lo haurem d'instal·lar el *Node.JS* al servidor en el cas que no ho estigui, per a fer-ho utilitzarem la següent comanda.

```
1 sudo apt install nodejs
```

Un cop tenim instal·lat node, ja podem utilitzar el gestor de paquets d'aquest (npm), factor que ens desbloqueja per a poder instal·lar el [PM2](#), per a fer-ho utilitzarem la següent comanda:

```
1 npm install pm2 -g
```

Ara ja tenim el pm2 instal·lat a la nostra màquina podem crear l'entrada d'aquest per el nostre projecte, per a fer-ho utilitzarem la següent comanda:

```
1 pm2 start app.js --interpreter=/home/easy/.nvm/versions/node/v10.15.3/bin/node --name="vote-vote-react" --log-date-format 'YYYY-MM-DD HH:mm:ss' -- --network alastria
```

En aquesta comanda s'han utilitzat certs paràmetres, com pot ser el de `-interpreter` on especifiquem la versió de node que volem que s'utilitzi, `-name` el nom que volem que tingui aquesta entrada, `-log-date-format` el format de la data dels logs i llavors a l'`app.js` li passem com a paràmetre `-network alastria` per a fer que el projecte faci les crides al contracte publicat a alastria i no al de la xarxa local.

Per a veure que s'ha creat correctament l'entrada del pm2 podem executar una de les següents dues comandes i veure detalls.

```
1 pm2 list
```

```
1 pm2 describe vote-vote-react
```

9.3 MongoDB

Un cop tenim l'entrada de pm2 amb el nostre projecte, necessitem tenir instal·lat el Mongo al servidor, per tant, en el cas que no el tinguem, ho farem utilitzant la següent comanda.

```
1 sudo apt-get install -y mongodb-org
```

Un cop instal·lat el farem arrancar, per a fer-ho utilitzarem la següent comanda.

```
1 sudo systemctl start mongod
```

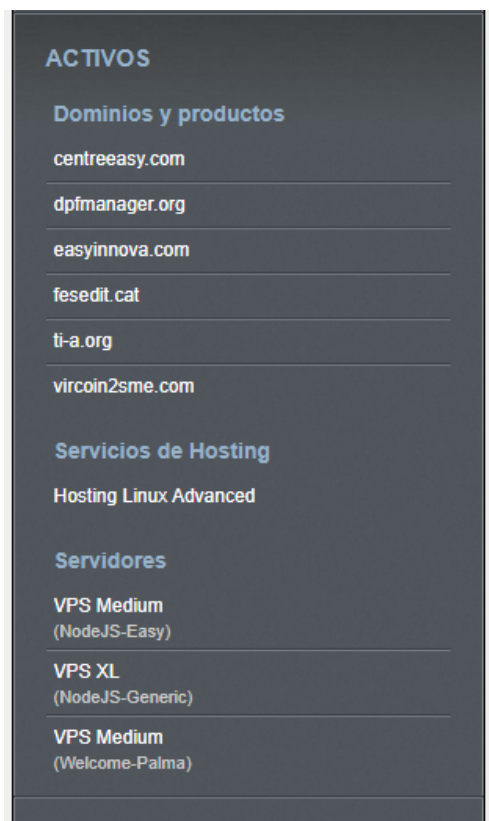
Per a comprovar que tot funciona correctament podem veure l'estat de mongo utilitzant

```
1 sudo systemctl status mongod
```

9.4 Nominalia

El servei de hosting que utilitzem al Centre TECNIO Easy és [Nominalia](#), per tant, hem d'entrar al panell de control d'aquest servei per tal de crear el nostre subdomini.

Per a fer-ho després d'autoritzar-nos anirem a la pestanya *Activos-Dominios y productos-centreeasy.com*, atès que, volem crear un subdomini d'aquest domini:
















Un cop dins aquesta pàgina, se'ns exposen les diferents opcions que podem configurar, en el nostre cas, *Dominio y Dns*:

PANEL DE CONTROL

centreeasy.com

AMPLIA TU PRESENCIA ONLINE CON DOMINIOS PREMIUM [Más información](#)

GESTIONA TU DOMINIO Y TUS PRODUCTOS

 DOMINIO Y DNS	 ASOCIACIÓN DOMINIO	 EMAIL
 SMTP	 ESPACIO WEB	 HOSTING LINUX
 HOSTING WINDOWS ¿QUÉ ES?	 EDITOR WEB	 SIMPLY SITE ¿QUÉ ES?
 TIENDA ONLINE ¿QUÉ ES?	 MARKETING ONLINE	 SITELOCK ¿QUÉ ES?
 OFFICE 365 ¿QUÉ ES?		

A continuació *Gestionar DNS*:



Dominio y DNS

Gestiona tu dominio, revisa los datos o modifica los DNS.

centreeasy.com

Visualiza todos los [subdominios](#) asociados a este producto o modifica las [asociaciones](#) actuales.

✕

Área de gestión

 Asociaciones y subdominios	 Gestionar DNS
 Asociación dominio	 Estado LOCK del dominio
 Datos y documentación	<input type="checkbox"/> Protégelo activando esta opción
 Transferir dominios	 Modificar datos de registro del dominio
 Código de autorización (Auth-info code)	 Cambio de Titular

En aquesta pàgina ens sortirà una llista dels DNS que tenim ja registrats, com no tenim cap amb el nom de vote-vote, anirem a *Afegir registre a*:

Configuració actual de DNS

Gestió guiada

Gestió avançada

Et recordem que, per poder gaudir dels nostres serveis addicionals, hauràs d'associar al teu domini els DNS per defecte.

[+ AFEGIR REGISTRE A](#) [+ AFEGIR CNAME](#) [+ AFEGIR REGISTRE MX](#)

Llista de DNS

Nom	Tipus	Valor	Cancel·lar
agile.centreeasy.com	A	84.88.154.252	Cancel·lar
answers.centreeasy.com	A	151.236.55.209	Cancel·lar
centreeasy.com	A	185.2.4.17	Cancel·lar
clientesvip.centreeasy.com	A	84.88.145.23	Cancel·lar
coso.centreeasy.com	A	84.88.145.23	Cancel·lar

Ens sortirà el següent modal on haurem d'introduir el nom del subdomini i la IP a la qual ha d'apuntar:

Crear / Modificar un registre A

Fes servir aquest formulari per crear o modificar un registre A (Address Record) per al teu domini.

Nom: .centreeasy.com

Afegir un registre A per a la zona

El registre A (Address Record) et permet associar un "nom" (Hostname) a una adreça IP. Cada domini té un registre A primari que associa el teu domini a una adreça IP.

Introdueix l'adreça IP: . . .

Un o més registres A poden estar associats a la mateixa adreça IP, per exemple, perquè aquesta mateixa adreça IP proporciona més serveis o bé perquè cada registre A (pràcticament cada subdomini) pot associar-se a una IP diferent. Pots crear tots els registres A que vulguis i associar-los a les adreces IP que prefereixis. Et recomanem fer servir només caràcters alfanumèrics.

[CREAR/MODIFICAR](#) [CANCEL·LAR](#)

Un cop fets tots aquests passos ja tenim el DNS associat al nostre servidor, ara ens toca configurar com toca les redireccions pertinents per al seu funcionament.

9.5 Nginx

Un cop tenim el domini apuntant al nostre servidor com hem explicat a l'apartat anterior hem de fer les redireccions pertinents, per això hem utilitzat [Nginx](#), en el cas que no el tingem instal·lat ho fem amb la següent comanda:

```
1 sudo apt install nginx
```

Iniciem l'nginx per a començar a configurar-lo:

```
1 sudo service nginx start
```

Fet això hem d'entrar al directori on estan situades les configuracions de *Nginx* i crearem el nostre fitxer de configuració per *votevote*, per a fer-ho utilitzarem el següent.

```
1 cd /etc/nginx/sites-available //entrar dins el directori de configuracions
2 nano votevote.centreeasy.conf //per crear el fitxer de configuracio
```

```
1 # NodeJS App
2 upstream node_votevote {
3     server 127.0.0.1:3032;
4 }
5
6 # Nginx Proxy
7 server {
8     # Server params
9     server_name votevote.centreeasy.com;
10
11     # Proxy redirect
12     location /api/ {
13         proxy_set_header X-Real-IP $remote_addr;
14         proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
15         proxy_set_header Host $http_host;
16         proxy_set_header X-NginX-Proxy true;
17         client_max_body_size 100M;
18
19         proxy_pass http://node_votevote;
20         proxy_redirect off;
21         break;
22     }
23
24     # Proxy redirect
25     location /socket/ {
26         proxy_set_header X-Real-IP $remote_addr;
27         proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
28         proxy_set_header Host $http_host;
29         proxy_set_header X-NginX-Proxy true;
30         client_max_body_size 100M;
31
```

```

32         proxy_pass http://node_votevote;
33         proxy_redirect off;
34         break;
35     }
36
37     # Client static files
38     root /home/easy/vote-vote-react/Client/web-build;
39     index index.html;
40
41     # Proxy redirect
42     location / {
43         #proxy_set_header X-Real-IP $remote_addr;
44         #proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
45         #proxy_set_header Host $http_host;
46         #proxy_set_header X-NginX-Proxy true;
47
48         #proxy_pass http://node_votevote;
49         #proxy_redirect off;
50         try_files $uri /index.html;
51     }
52
53
54     listen 443 ssl; # managed by Certbot
55     ssl_certificate /etc/letsencrypt/live/votevote.centreeasy.com/fullchain.pem; #
56     managed by Certbot
57     ssl_certificate_key /etc/letsencrypt/live/votevote.centreeasy.com/privkey.pem; #
58     managed by Certbot
59     include /etc/letsencrypt/options-ssl-nginx.conf; # managed by Certbot
60     ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem; # managed by Certbot
61 }
62 server {
63     if ($host = votevote.centreeasy.com) {
64         return 301 https://$host$request_uri;
65     } # managed by Certbot
66
67     server_name votevote.centreeasy.com;
68     listen 80;
69     return 404; # managed by Certbot
70 }

```

Listing 6: Fitxer de configuració nginx

9.6 Comprovació del funcionament

Un cop seguits tots aquests passos anem a la url <https://votevote.centreeasy.com> i si veiem imatge és que tot està funcionant correctament, per tant, provarem de crear una votació si funciona, vol dir que tot està amunt i preparat per a realitzar una bateria de proves.

10 Implantació i resultats

En primer lloc es va parlar amb el tutor pertanyent al grup de recerca [Centre TECNIO Easy](#) on va exposar el que es volia fer, i a partir d'allà analitzant el que es volia es va decidir l'entorn de desenvolupament de l'aplicació, així com les eines (software) i els llenguatges de programació a utilitzar a partir d'un estudi de requisits i d'opcions disponibles. El següent pas va ser estudiar el funcionament de les eines triades i començar a implementar l'aplicació amb la metodologia mostrada a l'apartat número 3. Un cop l'aplicació acabada, es va exportar a un entorn públic per adequar el seu funcionament, i els últims dies es van dedicar a refinar la documentació i arreglar petits detalls que s'han trobat.

10.1 Resultats obtinguts

Tots els objectius plantejats en aquest projecte s'han assolit de manera satisfactòria, obtenint una aplicació completament desenvolupada i a punt per ser utilitzada.

11 Conclusions

L'objectiu del projecte consistia en desenvolupar una aplicació de votació electrònica basada en Blockchain que intenta augmentar la participació amb un mecanisme d'incentivació, un cop acabada l'aplicació, puc afirmar que tots els requisits plantejats inicialment s'han completat de manera satisfactòria.

11.1 Aprentatge

Una de les fonamentals raons per les quals es realitza un Treball de Final de Grau és bàsicament per incrementar el coneixement en un dels sectors que desitgem i que possiblement durant la carrera no hem acabat d'aprendre, en el meu cas ha estat el de desenvolupar una aplicació web que utilitza tecnologia Blockchain, per tant, un cop acabat aquest Treball puc concloure que he après considerablement el següent:

1. Desenvolupar una aplicació completa i real
2. Utilitzar els coneixements adquirits durant la carrera com a base i aplicar-los
3. Millorar els coneixements de React-Native
4. Millorar els coneixements de JavaScript
5. Millorar els coneixements de MongoDB
6. Aprendre i millorar els coneixements sobre Blockchain i Smart Contracts
7. Aprendre a configurar un entorn de producció

11.2 Conclusions personals

Estic molt i molt satisfet dels objectius aconseguits, del treball realitzat i del munt de coneixements que he adquirit en aquest tram final del Grau. Considero que la meva aplicació satisfà totes les necessitats plantejades, és intuïtiva i personalment considero que té un aspecte atractiu.

12 Treball futur

En un futur es podrien aplicar les següents noves funcionalitats / millores a l'aplicació:

1. Com a usuari administrador poder veure un llistat de totes les votacions amb el seu estat (En curs, No iniciada encara, Finalitzada) i un resum de resultats en directe
2. Enviar un e-mail quan la votació ha finalitzat per temps amb l'enllaç de resultats
3. Fer un filtre al selectable de persones a les quals vols incentivar i mostrar només les persones que encara no han votat en aquella votació.
4. Poder crear votacions en les que es puguin seleccionar diverses respostes per pregunta

13 Bibliografia

[W3Schools](#): Consultat per dubtes sobre Node.js, JavaScript i derivats.

[StackOverflow](#): Consultat per tots els errors ocasionats durant el desenvolupament.

[Material UI](#) Utilitzat per a consultar documentació sobre components de la part de Front-End.

[Truffle](#) Utilitzat per a consultar documentació sobre la configuració de Truffle per l'Smart Contract.

[Solidity](#) Documentació sobre solidity.

[Nginx](#) Consultat per a documentació i configuració de Nginx.

[Mozilla](#) Consultat per a documentació de JavaScript.

[MongoDB](#) Documentació sobre consultes i derivats de MongoDB

14 Annexos

14.1 Pressupost

Un pressupost és la previsió de despeses i ingressos per a un determinat lapse de temps en el qual permet a les empreses, els governs, les organitzacions privades i les famílies establir prioritats i avaluar la consecució dels seus objectius.

Aquest procés consisteix en sumar els costos estimats del conjunt **d'activitats individuals o paquets de treball** per tal d'establir una línia base de cost autoritzada. Aquesta línia base inclou tots els pressupostos autoritzats, però exclou les reserves de gestió. Els pressupostos del projecte constitueixen els fons autoritzats per a executar el projecte. L'acompliment dels costos del projecte es mesurarà pel que fa a el pressupost autoritzat.

En aquesta secció de la documentació del projecte es realitzaran dues propostes de pressupost utilitzant dos tipus de mètriques diferents i finalment es realitzara un anàlisi per esbrinar quina sembla mes acurada. La primera es tracta d'una estimació de costos pel mètode de punts de funció i la segona fa referència a l'estimació per paquets de treball.

14.1.1 Mètode d'estimació per paquets de treball

El mètode d'estimació per paquets de treball consta en dividir el projecte en un conjunt de paquets de treball on cadascun d'ells té assignada un temps de realització. Cada paquet de treball consta d'un subconjunt d'activitats a desenvolupar per la consecució del mateix amb les quals podem aproximar amb mes fiabilitat la seva durada.

Un cop es té dividit el projecte en paquets de treball amb les relatives durades, s'assigna un cost per temps de treball i s'obté el cost final del projecte.

14.1.2 Estimació del temps per cada paquet de treball

Per començar a realitzar un pressupost amb la tècnica de paquets de treball, és imprescindible acordar el conjunt de paquets de treball a desenvolupar al projecte per tal de poder mesurar el cost en temps referent a cadascun amb l'objectiu d'aproximar el seu cost monetari amb la màxima precisió possible.

Per tal d'establir el valor monetari d'aquest projecte, començarem per establir els períodes de temps necessaris per realitzar els diferents paquets de treball, tasca la qual ha estat feta prèviament a l'apartat X(Estructura de descomposició de treball) de la documentació d'aquest projecte. La unitat de mesura emprada fou el **dia**. Els dies assignats a cada paquet de treball van ser els següents:

1. **Planificació - 7 dies**
2. **Documentació del projecte - 14 dies**
3. **Requisits funcionals - 7 dies**
4. **Dependències - 7 dies**
5. **Disseny de l'estructura de dades - 14 dies**
6. **Disseny de les funcionalitats - 7 dies**
7. **Disseny de les interfícies i pantalles - 7 dies**
8. **Implementació de les funcionalitats - 45 dies**
9. **Implementació de les interfícies - 45 dies**
10. **Implementació de mailing - 20 dies**
11. **Testeig de les funcionalitats - 14 dies**
12. **Testeig de les interfícies - 14 dies**
13. **Proves de rendiment - 14 dies**

La durada total del projecte és igual a 215 dies. El repartiment de la càrrega de treball en les diferents fases del projecte s'estableix de la següent manera:

1. **GESTIÓ DEL PROJECTE:** Planificació + Documentació del projecte – 21 dies; 10%
2. **DEFINICIÓ DE REQUISITS:** Requisits funcionals + Dependències – 14 dies; 5%
3. **DISSENY DEL PRODUCTE:** Disseny de l'estructura de dades + Disseny de les funcionalitats + Disseny de les interfícies i pantalles – 28 dies; 15%
4. **IMPLEMENTACIÓ:** Implementació de les funcionalitats + Implementació de les interfícies + Implementació de mailing – 110 dies; 50%
5. **ASSAIG I REVISIÓ:** Testeig de les funcionalitats + Testeig de les interfícies + Proves de rendiment – 42 dies; 20%

14.1.3 Estimació del cost monetari del projecte

Finalment, en la darrera part d'aquest pressupost, es definirà una aproximació del cost monetari del projecte en funció dels temps assignats en les del diferents fases i del salari mitjà d'un enginyer informàtic.

Basant-me en la meua experiència laboral opto per assignar com a sou mitjà d'un Enginyer Informàtic amb certa experiència al món laboral és d'uns 30000 euros de SBA (Sou Brut Anual), cosa que mensualment serien uns 2500 SBM (Sou Brut Mensual).

Suposant aquest sou mig de 2500 euros bruts i la distribució de temps anterior, s'ha determinat que el cost total del projecte és de **17625 euros**, dels quals un 50% pertanyen a la implementació de les funcionalitats, és a dir, **8812 euros**. A la següent taula podem trobar detallada la distribució dels costos en funció de les diferents fases del projecte:

#	NOM	%	Temps (mesos)	COST
1	GESTIÓ DEL PROJECTE	10	0.7	1750 €
2	DEFINICIÓ DE REQUISITS	5	0.35	875 €
3	DISSENY DEL PRODUCTE	15	1	2500 €
4	IMPLEMENTACIÓ	50	3.5	8750 €
5	ASSAIG I REVISIÓ	20	1.5	3750 €
TOTAL				17625 €

Cal dir que dita aproximació només inclou les despeses en sous de personal capacitats (enginyers). Aquí queden excloses les diverses despeses com:

1. Costos de hardware i llicències de software
2. Despeses de Seguretat Social, Assegurances...
3. Lloguer d'oficines, llum, xarxa...
4. Administració

15 Manual d'usuari

Accedir a la URL <https://votevote.centreeasy.com>.