



This is the **published version** of the bachelor thesis:

Vallbona Comas, Pau; Musach, Ramón, tut. Desenvolupament d'un software per la gestió de tasques i valor generat dins d'un equip. 2025. (Enginyeria Informàtica)

This version is available at <https://ddd.uab.cat/record/308789>

under the terms of the  license

Desenvolupament d'un software per la gestió de tasques i valor generat dins d'un equip

Pau Vallbona Comas

6 de febrer de 2025

Resum— Aquest treball de final de grau presenta el desenvolupament d'un programari de gestió de tasques i seguiment del valor enfocat a l'optimització del lideratge d'equips. L'aplicació, dissenyada com una Progressive Web App (PWA), facilita la delegació, el seguiment i l'avaluació eficient de tasques, proporcionant transparència i responsabilitat dins dels equips. Les funcionalitats principals inclouen la creació de tasques, l'anàlisi del rendiment basat en el valor generat, subhastes automatitzades de tasques i el seguiment detallat del progrés mitjançant eines visuals com Kanban. Construïda amb Vue.js, Node.js i MongoDB, la plataforma garanteix operacions segures, col·laboratives i escalables. Amb l'objectiu de superar les ineficiències de les estructures jeràrquiques tradicionals, aquesta solució empodera els líders per optimitzar els fluxos de treball, fomentar l'autonomia i maximitzar l'ús dels recursos. Integrant funcionalitats innovadores, adreça forats de les eines actuals del mercat.

Paraules clau— Gestió de tasques, Equips, Lideratge, Transparència, Anàlisi del valor generat, Subhastes de tasques, Kanban, Seguiment del progrés, Delegació eficient, Vue.js, Node.js.

Abstract— This end-of-degree project presents the development of a task management and value tracking software tailored for team leadership optimization. The application, designed as a Progressive Web App (PWA), facilitates efficient task delegation, tracking, and evaluation, providing transparency and accountability within teams. Key features include task creation, value-based performance analysis, automated task auctions, and detailed progress tracking through Kanban and other visualization tools. Built with Vue.js, Node.js, and MongoDB, the platform ensures secure, collaborative, and scalable operations. Aimed at overcoming inefficiencies in traditional hierarchical structures, this solution empowers leaders to streamline workflows, foster autonomy, and maximize resource utilization. By integrating innovative functionalities, it addresses gaps in existing market tools.

Keywords— Task management, Teams, Leadership, Transparency, Accountability, Value-driven analysis, Task auctions, Kanban, Progress tracking, Efficient delegation, Scalability, Vue.js, Node.js.

1 Introducció - Context del treball

En qualsevol equip o organització amb una estructura jeràrquica, una de les principals dificultats que es presenta és la ineficiència en la assignació de tasques i el control de la feina realitzada per cada membre de l'equip. Aquest tipus d'estructures sovint comporten una burocràcia excessiva i presenten problemes en la

fluïdesa de la comunicació, fent que la distribució de responsabilitats i la presa de decisions es vegi limitada per la complexitat de la jerarquia.

Si es treballa en un entorn on els marges de benefici són estrets i la innovació constant, qualsevol ineficiència pot suposar un desavantatge important. Per tant, una gestió eficient dels recursos és clau per maximitzar el rendiment, respondre més ràpidament a les demandes del mercat i garantir la sostenibilitat en el llarg termini.

El problema enfocat en aquest treball es situa en un equip que necessita controlar detalladament la seva utilització de recursos. Més concretament s'ha determinat

- E-mail de contacte: pau.vallbona@autonoma.cat
- Menció realitzada: Tecnologies de la Informació
- Treball tutoritzat per: Ramon Musach Pi
- Curs 2024/25

que el principal coll d'ampolla és la falta de bons líders; i s'intenta presentar eines informàtiques de gestió i sistemes d'informació per fer més assequible el lideratge de l'equip.

L'arrel del problema és que quan es treballa en equip ha d'existir una figura que assumeixi la responsabilitat d'assegurar-se de coordinar que totes les tasques es distribueixin i es compleixen dins del pressupost disponible. Aquesta figura és el líder, que ha de disposar d'una visió global per poder dirigir l'equip. Un bon líder ha d'aconseguir implicar directament cada membre de l'equip i garantir-li responsabilitat dels seus èxits, fracassos i errors. És una feina molt costosa per una sola persona, però només així es poden evitar situacions de descoordinació i duplicació de feina, i garantir que cada persona actui amb autonomia i responsabilitat en el seu àmbit.

En aquest informe es presentarà la memòria d'aquest treball en un ordre aproximat al cronològic en el que s'ha generat la informació. Per això es començarà amb la definició d'objectius, seguit dels requisits per complir-los. Seguidament s'analitzarà l'estat de l'art i la viabilitat funcional dels objectius plantejats, abans de començar l'execució. Finalment es plantejarà la metodologia necessària seguida per l'execució del projecte, i els resultats obtinguts. Per últim, s'acabarà el projecte validant el compliment dels objectius, a l'apartat de conclusió.

2 Objectius

Tal com s'ha contextualitzat anteriorment, l'objectiu d'aquest treball és desenvolupar eines informàtiques de gestió i sistemes d'informació per fer més assequible el bon lideratge d'un equip. Més concretament, es vol aconseguir una forma de mesura ràpida i traçable de la feina realitzada i benefici aportat per cada membre d'un equip. També es destaca la importància de la seguretat de la informació, ja que es tracta amb dades sensibles.

Això es vol assolir a través del desenvolupament d'una eina digital, en forma d'una aplicació web accessible des de qualsevol dispositiu. Aquesta aplicació ha de ser col·laborativa i segura, i permetre facilitar a qualsevol usuari la capacitat de liderar un equip de manera eficient. Ha de permetre reduir la feina administrativa del líder a registrar les tasques pendents i validar el seu compliment correcte.

A més, la plataforma ha de permetre registrar i monitorar la feina individual de cada membre, incloent-hi el líder, per garantir la transparència, la propietat dels resultats i el reconeixement del valor aportat per cadascú. L'eina també ha d'oferir diverses vistes de l'estat de cada projecte, afavorint la visibilitat i el seguiment del progrés, i millorant la col·laboració i la responsabilitat dins de l'equip.

Les noves funcionalitats que aquest treball vol acon-

seguir i que es creu que no existeixen en altres solucions existents, que més tard s'exploraran en l'estat de l'art, inclouen diverses millores per optimitzar la gestió de tasques i el seguiment del projecte. Més concretament es vol introduir la capacitat de relacionar el valor generat per cada tasca, així com una anàlisi individual del valor aportat per cada membre i el compliment de les responsabilitats preses. També es vol poder crear subtasques amb percentatges de temps estimats, permetent un control més detallat de l'evolució del projecte. A més, es vol incorporar un sistema de subhasta per assignar tasques, on els membres competiran per triar tasques segons les seves capacitats de dedicar-hi menys temps. Finalment es podrà fixar un percentatge de marge que es vol aconseguir per tasca, establint límits clars de temps i recursos, per determinar el valor afegit corresponent a cada membre de l'equip.

3 Requisits

Per tenir una visió més concreta de les fites del desenvolupament del projecte, s'ha definit el backlog que conté la llista de tots els requisits.

S'ha plantejat els requisits en forma de casos d'ús, i els principals actors que s'ha determinat són un usuari amb el rol de líder, un avaluador, i un membre de l'equip. El líder és l'encarregat de crear tasques, que són executades per un membre, i un cop completades són avaluades per un avaluador. A partir d'aquesta descripció, s'ha determinat que cada tipus d'usuari ha de poder realitzar les següents funcions:

3.1 Requisits funcionals

Casos d'ús d'un nou usuari

- Iniciar sessió.
- Modificar el seu usuari (assignar nom, organització, i equips).
- Crear equip (atorga permís de líder sobre aquell equip).
- Rebre invitacions de altres equips (atorga el nivell de permís assignat en la invitació)

Casos d'ús d'un Líder

- Convidar usuaris (existents o nous per correu) a l'equip.
- Modificar els permisos dels usuaris de l'equip.
- Crear tasques dins l'equip.
- Assignar a cada tasca el nombre d'hores aproximat per completar-la.
- Associar a les tasques el valor generat aproximat al completar-se.
- Assignar cada tasca dins l'equip a un usuari.
- Veure el progrés general de totes les tasques assignades dins l'equip
- Posar a subhasta tasques amb places per un o més membres.

- Veure valor generat per cada membre de l'equip.
- Tornar a rebre tasques que s'ha delegat però han estat renunciades.
- Tots els permisos de un avaluador de l'equip.

Casos d'ús d'un Membre

- Veure tasques proposades a l'usuari
- Acceptar o rebutjar qualsevol tasca proposada
- Veure tasques pendents a l'usuari
 - En format de taula (visual board).
 - En format de taula Kanban (capacitat).
 - En format de Diagrama de Gantt (timeline i dependències).
- Iniciar o parar qualsevol tasques assignada.
- Renunciar a una tasca, quan no es pot completar.
- Completar una tasca assignada i introduir el nombre d'hores dedicades.
- Crear tasques filles d'una tasca assignada a l'usuari.
- Assignar les subtasques que ha creat l'usuari.
- Veure tasques subhastades a l'equip.
- Afegir a una subhasta una ofertes d'hores necessàries per completar una tasca

Casos d'ús d'un Avaluador

- Veure les tasques que han estat completades però no validades
 - hores dedicades a la tasca
 - valor generat assignat per l'administrador (o calculat en subtasques, a partir del valor les tasques pares)
- Validar i valorar el compliment de cada tasca (en format de 1-5 estrelles).
- Marcar una mala atribució d'hores necessàries per una tasca.
- Tots els permisos d'un membre de l'equip

3.2 Requisits de seguretat

Degut a que l'aplicació ha de poder ser utilitzada en equips que treballen amb informació sensible, és molt important que es garanteixi la seguretat de la informació. Per això, s'ha determinat una llista de requisits específics a la seguretat:

- Cap usuari ha de poder veure o modificar informació sobre altres usuaris, equips, o tasques per les quals no té permís explícit.
- Sempre que s'envii informació entre diferents ubicacions geogràfiques, aquesta ha d'estar xifrada durant tot el trajecte.
- Un usuari no hauria de poder executar codi arbitrari en el servidor.

- Un usuari no hauria de poder executar codi arbitrari en la base de dades.
- Un usuari no hauria de poder executar codi en dispositius d'altres clients.
- La pàgina ha de poder funcionar sense interrupcions, durant un temps indefinit.
- Un usuari amb males intencions no hauria de poder afectar el bon funcionament de l'aplicació per altres usuaris.

4 Estudi de Viabilitat

Un cop definida la idea, s'ha dut a terme un procés de recerca per avaluar la validesa del plantejament del treball.

En primer lloc s'ha verificat que existeixi la necessitat a nivell comercial d'una nova eina per complir els requisits plantejats, i fins a quin punt pot ser necessari o útil desenvolupar un nou software per completar els objectius del treball. També s'ha investigat sobre els requisits a nivell legal que podrien condicionar o impedir el desenvolupament del projecte.

Un cop validada la idea, s'ha analitzat la capacitat tecnològica per desenvolupar la idea dins dels límits de l'extensió d'aquest treball. S'ha avaluat la possibilitat d'utilitzar eines ja existents, ja que el desenvolupament web és un format bastant comú per la implementació de projectes de software.

Finalment s'ha identificat els recursos necessaris per la implementació del projecte, i la capacitat d'executar el desenvolupament amb els recursos disponibles en el treball.

4.1 Estat de l'art - Viabilitat comercial

En primer lloc, s'ha revisat solucions semblants o altres eines relacionades amb la gestió de tasques, responsabilització individual, calcul del valor generat, i seguiment del progrés. Resulta que algunes d'aquestes funcionalitats estan presents en varis tipus de programis, enfocats en la productivitat, la transparència i la responsabilitat dins dels equips.

Per una banda els softwares de gestió de projectes han estat evolucionant significativament en els últims anys amb un enfoc similar, incorporant diverses funcionalitats per millorar l'eficiència i la col·laboració entre membres d'un equip.

Per altra banda, també s'ha considerat sistemes d'informació ERP que incloguin mòduls per la gestió de tasques i mòduls de gestió de la facturació. D'aquesta manera es vol intentar veure fins a quin punt es poden integrar aquests dos mòduls per acostar-se més als objectius proposats.

S'ha fet la taula [Taula 1] per valorar si algun dels programis analitzats inclouen totes les funcionalitats que es busquen, explicades als objectius.

	Trello	Asana	Monday	Jira	ClickUp	Wrike	Odoo	SAP
Creació de subtasques i agrupacions de carrega de treball	5	10	10	5	10	5	10	10
Sistemes de subhasta o assignació automàtica de tasques	0	3	5	5	5	5	5	5
Relació d'una tasca amb el valor generat	0	0	0	0	4	0	10	7
Anàlisi del valor aportar individualment	0	0	0	0	0	0	10	6
Total	5/40	13/40	15/40	10/40	19/40	10/40	35/40	28/40

TAULA 1: COMPARACIÓ D'ASSOLIMENT DE LES FUNCIONALITATS BUSCADES ENTRE DIFERENTS EINES EXISTENTS. VALORADES SOBRE 0-10.

En resum, les opcions que més s'acosten són Odoo i SAP, gràcies a que la integració amb la facturació i la capacitat de relacionar tasques amb el valor generat. Especialment Odoo destaca com la millor opció, ja que la implementació és molt més fàcil de configurar i gairebé funciona amb les opcions per defecte.

Tot i així, fins i tot considerant la millor solució existent, cap no compleix totes les característiques que es necessiten en el software a desenvolupar. Sobre tot la funcionalitat que cap software compleix actualment és l'assignació de tasques per subhasta, on s'assigni una tasca al membre de l'equip que digui que la pot acabar en menys temps. S'ha considerat que la falta de funcionalitats en totes les opcions actuals és suficient raó per seguir endavant amb el desenvolupament de l'eina que es vol crear amb aquest projecte.

4.2 Viabilitat legal

Degut a que el software desenvolupat es vol protegir amb una llicència i publicar per poder ser utilitzat per qualsevol equip o organització, cal tenir en compte quines responsabilitats legals pot tenir el desenvolupador del codi original.

Els principals àmbits que s'ha trobat que cal avaluar són la protecció de les dades

S'ha analitzat els requisits legals per garantir el compliment normatiu i la seguretat dels usuaris, i s'ha determinat que les principals proteccions legals per el propietari del codi són assegurar la protecció de les dades i definir les condicions d'ús.

4.2.1 Protecció del codi

Si es vol evitar l'ús no autoritzat del codi desenvolupat, cal aplicar drets d'autor i una llicència restrictiva, així com notes de copyright visibles a cada fitxer i a la interfície.

També cal considerar quines llicències són aplicables, tenint en compte que si el codi utilitza llibreries ja existents cadascuna pot tenir la seva pròpia llicència que restringeixi la utilització d'aquella peça de codi en altres productes.

4.2.2 Responsabilitat legal

Per limitar la responsabilitat legal de l'aplicació i protegir-nos sobre disputes, cal definir uns Termes i Condicions d'ús del software.

Aquest document ha de ser vist i acceptat per qualsevol usuari abans de la utilització del software i ha d'incloure:

- Normes d'ús de la plataforma
- Limitacions de responsabilitat, especialment en cas de pèrdua de dades o interrupcions del servei.
- Condicions per a la suspensió o eliminació de comptes en cas d'ús inadequat.

4.2.3 Protecció de les dades

Com que es gestionarà informació personal, s'ha d'adaptar al reglament europeu RGPD. Les principals consideracions són:

- Es necessita el consentiment explícit dels usuaris per recollir i emmagatzemar les dades, per poder tenir una base legal per el tractament de les dades.
- Cal garantir que només s'emmagatzemen les dades estrictament necessàries per al funcionament de l'aplicació.
- Cal garantir la seguretat de les dades. Per tant, si es vol utilitzar proveïdors cloud s'ha de revisar les polítiques d'ús de les infraestructures garantint el xifrat i la ubicació segura de les dades.
- Els usuaris han de poder sol·licitar l'eliminació de les seves dades.
- Cal redactar un document de política de privacitat, que expliqui clarament com es gestionen les dades dels usuaris.

A més, amb aquestes mesures també es compleixen la majoria de lleis a altres països, com la CCPA als Estats Units.

4.3 Viabilitat tecnològica

S'ha determinat que tots els requisits proposats poden ser complerts amb el desenvolupament d'una aplicació web progressiva (PWA).

Una aplicació d'aquest tipus pot ser desenvolupada amb qualsevol arquitectura de desenvolupament web que inclogui un o varis back-ends (servidor web + APIs) i un front-end (programa executat per el client).

En el cas particular del desenvolupament web, que és el que es vol treballar, ja existeixen una gran quantitat de llibreries dedicades a implementar aquesta arquitectura i facilitar la implementació de cadascuna de les parts.

S'ha decidit utilitzar Vue.js al front-end, i Node.js + Express.js al back-end. Aquestes tecnologies, ja molt testejades per grans comunitats d'usuaris durant més de 15 anys, permeten reduir el desenvolupament de l'aplicació web que es vol aconseguir. A més, permeten reduir la feina del projecte a una càrrega raonable per el temps disposat.

4.4 Viabilitat financera

Els recursos previstos per el desenvolupament del projecte són els següents

- 1 estudiant d'enginyeria informàtica (capacitat de 300 hores per tot el projecte)

: I els materials:

- 1 domini web (aproximadament 5€ el primer any)
- 1 certificat https
- infraestructura que executa el backend de l'aplicació
- infraestructura que serveix al client el frontend de la pagina web
- infraestructura que emmagatzema les dades de l'aplicació
- infraestructura garanteix la comunicació segura entre el client i la base de dades

Tenint en compte que el projecte dura 19 setmanes, i ignorant dies festius, es disposa d'unes 300 hores d'un estudiant d'enginyeria. S'estima que aquesta quantitat de temps ha de ser més que suficient per desenvolupar totes les etapes del projecte.

5 Desenvolupament

5.1 Metodologia

Una gran part de la carrega de feina d'aquest treball es troba en el desenvolupament de software. També cal tenir en compte que tenim un temps limitat per el desenvolupament i necessitem sempre algun prototip funcional encara que no inclogui totes les funcionalitats

originalment desitjades. Per aquestes raons he volgut escollir una metodologia del tipus agile, més concretament el sistema anomenat Kanban.

Aquesta metodologia permet el desenvolupament de software amb un procediment molt flexible i visual, basat en iteracions. Consisteix en gestionar el treball mitjançant l'actualització diària d'un tauler visual que categoritza les tasques en diverses etapes (per exemple: "Per fer", "En curs", "Fet").

Aquesta metodologia és molt útil per a treballs continus i per evitar la sobrecàrrega de tasques i es pot utilitzar per el desenvolupament individual. Disposa dels següents components:

- Tauler Kanban: Utilitza un tauler (físic o digital) on les tasques es mouen a través de diferents columnes (Per fer, En curs, Revisió, Fet).
- Gestió del flux de treball: Limita el nombre de tasques que es poden tenir en "En curs" per evitar la sobrecàrrega de treball i assegurar que el focus es mantingui en tasques prioritàries.
- Revisió i retroalimentació: Al final de cada etapa, fa una revisió del treball completat i ajusta la prioritat de les tasques següents.

Aplicat al desenvolupament de software, les principals fases d'aquesta metodologia son:

- Backlog de les funcionalitats necessàries
- requeriments funcionals i no funcionals
- planificació del cicle diari de desenvolupament
- implementació
- revisió del cicle, actualització del tauler i modificació de requeriments no funcionals
- retrospectiva del cicle i tornar a la planificació

5.2 Seguiment

Per seguir el desenvolupament del treball es va utilitzar l'eina Microsoft Project. Així doncs es va dividir el projecte en cinc fases, compostes per les varies tasques del treball.

D'aquesta manera es podien visualitzar les tasques sobre un diagrama de Gantt (figura [13]), i analitzar el progrés del projecte respecte a la línia de base. En cada informe de seguiment es va incloure els informes rellevants sobre el progrés del desenvolupament i els retards acumulats.

En primer lloc, en la fase de planificació inicial (del 15 de setembre al 6 d'octubre) es varen definir els requisits del sistema, es va dissenyar l'arquitectura del sistema, i es va seleccionar les tecnologies a utilitzar per implementar la aplicació.

Seguidament es va començar el desenvolupament del backend (del 6 d'octubre al 26 d'octubre), on es va implementar l'esquema de base de dades, i es va desenvolupar el servidor amb l'API per la gestió de tasques.

Per altra banda, al mateix moment començava la fase de desenvolupament del fronten (del 6 d'octubre al 10

de desembre) on es dissenyava la interfície d'usuari, es desenvolupava el dashboard principal de tasques i estat del projecte, i es creaven la resta de pantalles rellevants per complir els requisits. Entre aquestes pantalles hi ha la de bústia de recepció de noves tasques i valoració de tasques completades, la de subhasta de tasques, la de configuració d'usuari i equips, i la de estadístiques.

Seguidament es va continuar amb la fase d'integració de les dues parts desenvolupades i de proves (16 de desembre - 15 de gener). En aquesta fase es varen dur a terme totes les proves de funcionalitat entre el backend i el frontend, modificant les parts necessàries per assegurar el funcionament conjunt. Un cop acabat el desenvolupament, es va seguir amb les proves de seguretat per validar la seguretat de la informació tractada. També es va dur a terme diverses proves de rendiment per assegurar que l'aplicació funcionés sense necessitar masses recursos, sobretot amb la funció de sincronització.

Finalment es va executar el desplegament i llançament de l'aplicació (del 16 de desembre al 24 de gener). En aquesta fase es va configurar l'entorn de producció a AWS, i es va desplegar la versió final del software.

6 Resultats

El desenvolupament del treball ha resultat en una aplicació web del tipus PWA. Consisteix en una web molt intensiva a la part del client, que s'instal·la al navegador i descarrega del servidor totes les dades que necessita per funcionar. Múltiples usuaris poden accedir a aquesta aplicació, concurrentment i de forma segura, a través del domini <https://fynix.pro>.

Per mostrar més detalladament la interacció entre els components de l'aplicació, s'ha dividit aquesta secció en dos apartats. Primerament a continuació es descriu l'arquitectura de la infraestructura que manté en funcionament el backend de l'aplicació, seguida de la documentació del procés de funcionament de l'aplicació.

6.1 Arquitectura de l'aplicació

L'aplicació està sostinguda per tres components que s'executen paral·lelament, que són una instància de NodeJS, una instància de MongoDB, i la aplicació instal·lada al client. Aquesta arquitectura es pot visualitzar més clarament amb la imatge [1].

6.1.1 Instància de Node.js

És el punt d'entrada per tots els usuaris. Disposa d'una adreça pública i un domini URI, on escolta totes les peticions del protocol http o https (port 80 i port 443). D'aquesta manera qualsevol usuari hi pot contactar des de qualsevol navegador estàndard, i distribueix els recursos necessaris perquè el client pugui executar la seva part de l'aplicació.

Primerament serveix el programa principal de l'aplicació que ha d'executar el client, i totes les dependèn-

cies i altres fitxers principals que permeten executar i instal·lar l'aplicació al client.

També és l'encarregat de respondre a totes les peticions de l'API establerta amb el programa executat al client, validar l'autenticació i transferir les dades corresponents o fer les peticions necessàries a la base de dades.

El programa principal s'executa al núvol sobre una instància de comput EC2 de AWS. També s'ha configurat un grup d'escalat automàtic, de manera que si la capacitat de processament del servidor supera el 80% durant un període de 5 minuts, automàticament s'escala horitzontalment amb noves instàncies temporals iguals a la primera. Actualment està limitat a 4 instàncies per evitar un augment inesperat i excessiu del consum de recursos, i retorna a una sola instància quan torna a baixar el consum de recursos.

S'autoconfigura al iniciar-se degut a les instruccions introduïdes en un template d'AWS, i descarrega els fitxers de configuració i de l'aplicació a través de un repositori Github.

6.1.2 Instància de MongoDB

És la instància on es guarden totes les dades de l'aplicació. El servidor NodeJS fa les peticions corresponents per assegurar que les dades es s'introdueixen, es recuperen, i es modifiquen corresponentment amb les peticions dels clients.

També s'executa al núvol, sobre una instància de AWS DocumentDB, amb escalabilitat horitzontal automàtica. Està configurat en el mode de Multi-AZ, que automàticament assegura que les dades es replicaran a diverses zones de disponibilitat dins de la regió seleccionada.

Amb aquesta configuració, es designa una base de dades principal que serveix per lectura i escriptura de totes les dades. Per altra banda, es creen diverses instàncies que repliquen les dades de la primera però que només permeten la lectura de les dades. D'aquesta manera es distribueixen les peticions de lectura, que són les més comunes, entre totes les instàncies.

6.1.3 Client executant part instal·lada de l'aplicació de Vue.js

Està formada en primer lloc per el fitxer index.html retut al fer una petició al servidor. Al executar el fitxer principal es descarreguen i importen el controlador principal i tots altres fitxers dels que depèn l'aplicació per ser visualitzada, fent més peticions als recursos corresponents del servidor.

És el procés que s'executa en cadascun dels usuaris que utilitzen l'aplicació. Utilitza una arquitectura de model-vista-controlador, ja que és el model estàndard utilitzat per cada component desenvolupat amb el framework Vue.js.

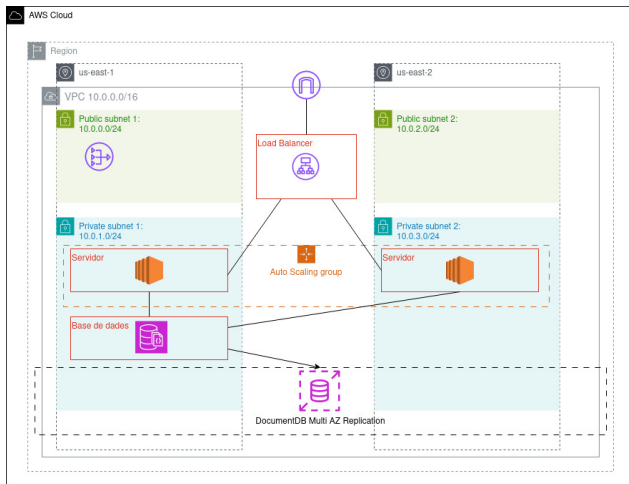


Fig. 1: Representació de l'arquitectura del backend de l'aplicació, que processa totes les peticions dels clients. Començant per la part superior central podem veure la gateway (porta d'entrada d'Internet) per on entren les peticions, seguida del load balancer (balancejador de càrrega) que té assignada la IP pública del backend.

6.2 Processos de l'aplicació

Els tres components de l'aplicació descrits anteriorment es coordinen per complir cadascun dels objectius determinats en els requisits de l'aplicació. Seguidament s'entra en més detall en la interacció i la funció de cadascun dels components de l'aplicació, per complir els requisits de les especificacions del projecte.

6.2.1 Accés a l'aplicació

Per accedir a l'aplicació simplement s'ha d'accedir a un navegador estàndard des de qualsevol dispositiu, i buscar el nom del domini <https://fynix.pro>. Seguidament apareix un formulari que requereix usuari i contrasenya (figura [2]), ja que cal iniciar sessió per començar a utilitzar la aplicació. L'usuari introdueix les dades i s'envien al servidor, que les saneja, valida, i si són correctes assigna al client un JWT (token d'autenticació de javascript) que garanteix la identitat de l'usuari. Amb el nou token, el programa del client demana la llista d'equips de l'usuari i les dades de totes les tasques de cadascun dels equips. Al rebre la resposta del servidor, el programa del client mostra la pàgina principal de visió de tasques assignades. Aquesta pàgina es mostra a la imatge [3].

6.2.2 Instal·lació de l'aplicació

L'aplicació disposa dels requisits per instal·lar-se com a aplicació en qualsevol navegador que disposi d'aquesta funció. Al entrar a la pàgina principal, si la aplicació no està instal·lada el programa del client demana permisos al navegador per instal·lar-se. Si el client accepta l'aplicació s'instal·la, i es pot afegir la icona a l'escriptori o la ubicació seleccionada per el client. D'aquesta manera els principals fitxers i dades de l'aplicació queden guardats, i l'aplicació es pot utilitzar offline fins que torni a tenir connexió i es pugui sincronitzar amb el servidor.

6.2.3 Sincronització de l'aplicació

Per poder utilitzar la aplicació offline, el client i el servidor han de mantenir un historial de canvis. El client demana al servidor el seu historial a partir de la data de la última sincronització, i després compara la resposta amb el seu propi historial. El client ordena les entrades i les torna a executar en l'ordre correcte sobre la còpia local de les dades. Les peticions no sincronitzades es filtren segons la compatibilitat amb l'historial del servidor i s'envien al servidor.

6.2.4 Edició d'un equip i invitació d'usuaris

Per editar un equip o els usuaris afegits, cal entrar a la pàgina "Teams". Aquí apareix la opció de crear un nou equip, editar un equip ja existent, o desplegar la llista de subequips de un equip. Si despleguem la llista també es poden crear subequips o aplicar qualsevol de les accions anteriors recursivament sobre l'arbre d'equips.

Dins de la pàgina d'edició d'un equip es poden afegir o eliminar altres equips als que es poden transferir tasques (sense ser subequips), o altres equips dels que es poden rebre tasques. Això es fa a través del codi d'un equip, que és un secret que es regenera manualment en qualsevol moment, però que només ha de saber el líder d'un equip i donar-lo en cas de necessitar utilitzar aquesta funcionalitat.

També dins de la pàgina d'edició d'un equip es pot editar el nom de l'equip, editar els permisos dels usuaris de l'equip, o afegir un nou usuari a través del seu correu. Al guardar, si s'ha afegit algun usuari, el servidor fa una petició al servidor de email perquè es generi un correu d'invitació, amb un botó per acceptar i un codi secret no visible per l'usuari. Si l'usuari accepta, s'envia el codi de la invitació al servidor, i el servidor completa el procés de crear l'usuari o afegir l'equip a la llista d'equips, i de modificar la llista de permisos de l'equip.

6.2.5 Creació de tasques

Per crear una tasca cal clicar el botó "Create task" de la barra de navegació. Aquesta acció desplega un menú on cal introduir les dades necessàries per crear una nova tasca. No hi ha cap valor obligatori, ja que per defecte s'assigna l'usuari que ha creat la tasca i el seu equip principal, i la data de finalització a el dia actual però en estat "Unassigned". Per tant, per defecte s'envia aquesta tasca a dalt de tot de la bústia de tasques per gestionar. Les dades que es poden modificar a una tasca són:

- títol, descripció, prioritat
- capacitat de crear subtasques
- capacitat d'afegir comentaris

A més, si la tasca no té subtasques també es pot modificar:

- estat (amb botons de passar al següent o tornar a l'anterior, depenent de l'estat actual)
- duració real
- percentatge de completada

I si l'usuari és líder de l'equip o ha creat la tasca:

- duració estimada
- valor aportat
- assignar usuaris

Per defecte aquestes modificacions es guarden al sortir de la tasca, a no ser que es seleccioni el botó de descartar canvis. També hi ha un altre botó per renunciar a la tasca, independentment de l'estat de desenvolupament en el que es troba.

6.2.6 Visualització de tasques

Al accedir a la pàgina principal es mostra una taula Kanban (figura [3], que permet visualitzar les tasques assignades a un mateix i desplaçar-les entre estats, a més de poder clicar per accedir a editar la tasca. Si s'obre el menú flotant de filtres, es poden aplicar varis filtres. Un dels més importants és la checkbox de "visió de gestió", ja que permet que l'usuari no només vegi les tasques assignades a ell mateix, sinó l'estat general de totes les tasques principals de l'equip.

6.2.7 Recepció i/o assignació de tasques no assignades a un usuari o equip

Quan entra una nova tasca a la bústia de l'usuari, es manté en aquest estat fins que és gestionada. S'ha d'accedir a la bústia per veure la llista de tasques no gestionades, ordenades per data de finalització. Si es vol gestionar una tasca, es poden desplegar les possibles accions clicant la tasca. Primerament apareixen dos botons:

- Acceptar.
- Renunciar.

Si l'usuari és líder de l'equip i veu aquella tasca, és degut a que aquella tasca encara no ha estat assignada a cap usuari. En aquest cas, també les següents opcions:

- Assignar. Permet seleccionar el membre de l'equip al que es vol assignar la tasca.
- Subhastar. Cal introduir una data en la que es tancarà la subhasta, i seguidament inicia un procés de subhasta entre els membre de l'equip al que està assignada la tasca.
- Transferir. Permet seleccionar un nou equip al que transferir la tasca, d'entre els subequips o altres equips que han donat permís.

- usuari rep notificació que té nova tasca no iniciada a la bústia - pàgina de bústia - hi ha tasques per assignar? - botó acceptar - botó renunciar - és administrador? - botó assignar - botó subhastar - botó transferir equip - es modifica la tasca corresponent

6.2.8 Avaluació de tasques

A la bústia de l'usuari, si l'usuari és avaluador o líder d'algun equip, també apareix una llista de tasques per avaluar. Aquestes tasques són les que un membre ha passat a l'estat de completades però que encara no han estat revisades. Es pot clicar sobre qualsevol tasca per desplegar el formulari d'avaluació. El formulari demana un comentari de feedback sobre la tasca completada, i una selecció de review entre 1 i 5 estrelles. Al completar l'avaluació la tasca es marca com a avaluada.

6.2.9 Subhasta de tasques

A la pàgina de subhastes apareixeràn les tasques que es troben actualment en subhasta, acompanyades d'un comptador decreixent de temps restant (suposant que el líder ja ha iniciat alguna subhasta, des de la bústia). Si s'acaba el temps d'una subhasta, la tasca s'assigna a l'usuari que ha introduït l'aposta del menor temps necessari per completar la tasca, o al líder de l'equip si no hi ha apostes. Qualsevol usuari de l'equip pot seleccionar una tasca i afegir el nombre d'hores que estima que tardarà a completar-la, i veu els valor introduïts per els altres usuaris.

6.2.10 Estadístiques d'equip

Si és el líder d'algun equip, accedint a la pàgina d'estadístiques de l'equip es poden veure un recull de mètriques interessants per la gestió. Després de seleccionar l'equip que es vol visualitzar, es mostra les estadístiques del valor generat per les tasques assignades a cada membre de l'equip (calculat multiplicant el valor generat per una subtasca i el seu percentatge de completada). També es desglossa per l'estat de les tasques, ja que poden tenir molt valor completat però no estar acabades.

6.3 Anàlisi de seguretat

L'aplicació està protegida contra atacs de "Person in the Middle", ja que totes les connexions estan encriptades amb el protocol tls (https en la connexió entre el servidor i el client), tant entre el client i el servidor com entre la base de dades i el servidor.

També es protegeix contra injecció de codi a tots els formularis, aplicant la conversió de tots els caràcters d'entrada a caràcters segurs per ser interpretats tant per el servidor com per una pàgina HTML en cas dels comentaris.

També es protegeix contra el robatori del credencials (el JWT emmagatzemat al navegador del client), aplicant una caducitat de dues hores.

Finalment l'aplicació també és bastant robusta pels en el nombre de connexions, ja que la infraestructura AWS escala automàticament al augmentar la càrrega en el servidor o la base de dades.

7 Conclusions

Degut a les característiques d'aquest treball enfocat a desenvolupar un producte funcional, podem comprovar l'èxit del desenvolupament tornant a revisar el compliment de cadascun dels objectius.

7.1 Compliment dels objectius

En primer lloc s'ha pogut comprovar que s'ha complert l'objectiu principal, ja que el desenvolupament del treball ha resultat en una aplicació web del tipus PWA, on múltiples usuaris poden accedir concurrentment i de forma segura a través del domini <https://fynix.pro>. A més, tal com s'ha verificat en l'apartat de resultats, es compleixen tots els estàndars de seguretat per aquest tipus d'aplicació.

Aquesta aplicació és una eina informàtica de gestió i un sistema d'informació que aconsegueix fer més assequible el bon lideratge d'un equip. Més concretament, s'ha aconseguit una forma de monitorar i medir de forma ràpida i traçable de la feina realitzada i benefici aportat per cada membre d'un equip.

El resultat ha complert tenir la forma d'una aplicació web accessible des de qualsevol dispositiu, que a més és col·laborativa i segura. Permet facilitar a qualsevol usuari la capacitat de liderar un equip de manera eficient, i redueix la feina administrativa del líder a registrar les tasques pendents i validar el seu compliment correcte.

La nova plataforma permet registrar i monitorar la feina individual de cada membre, incloent-hi el líder, i és transparent en garantir la propietat dels resultats i el reconeixement del valor aportat per cadascú. L'eina també ofereix diverses vistes de l'estat de cada projecte, afavorint la visibilitat i el seguiment del progrés, i millorant la col·laboració i la responsabilitat dins de l'equip.

Respecte a la idea inicial, les úniques funcionalitats que no s'ha implementat és el processament de dades de costos, ja que es va decidir durant l'estudi de viabilitat financera no era un requisit necessari per el funcionament ni la seguretat de l'aplicació i que calia reduir el treball degut a la limitació de 300 hores per desenvolupar el treball.

Tot i així, es considera que l'aplicació desenvolupada segueix sent una eina innovadora i que aporta un gran valor afegit per els líders de qualsevol equip. Així doncs, es considera que el projecte ha estat un èxit.

7.2 Línies de continuació

Tot i que ja es disposa d'una aplicació funcional, i que ja hi ha una versió desplegada, sempre es poden afegir modificacions al codi per afegir funcionalitats. Algunes modificacions que podrien ser interessants si es decideix executar altres projectes de continuació serien les següents:

7.2.1 Protecció addicional per garantir la disponibilitat del servei

Tot i que actualment no s'ha considerat necessari, ja que el software desenvolupat no ha estat comercialitzat i no té un nombre d'usuaris suficientment gran, es podria implementar mesures de protecció contra atacs de denegació de servei. Per exemple, utilitzant Cloudflare, qualsevol petició d'un client seria primer rebuda per una instància dedicada de Cloudflare de cache i protecció contra DDoS. Si l'usuari supera els filtres configurats a Cloudflare, es serveix el contingut cachejat o es redirigeix la petició al servidor d'AWS. Si no supera algun dels filtres, com la comprovació de la seguretat de l'origen de la IP, es serviria un CAPTCHA per assegurar que l'usuari és un humà i evitar la saturació de peticions de bots automatitzats.

7.2.2 Sistema de notificacions dins l'aplicació

Ja que l'aplicació pot ser instal·lada com a PWA, es pot implementar un sistema de notificacions. Al transferir-se una nova tasca a la bústia d'un usuari es podria enviar una notificació a la aplicació del client, per assegurar que sàpiga que té noves tasques sense necessitat d'entrar a l'aplicació.

També es podria enviar fer el mateix a través de correus a través del servidor de mail a l'adreça de l'usuari.

Referències

- [1] Bani Ali, A. S., Anbari, F. T., & Money, W. H. (2008). Impact of Organizational and Project Factors on Acceptance and Usage of Project Management Software and Perceived Project Success. *Project Management Journal*, 39(2), 5-33. [En línia] Disponible a: <https://doi.org/10.1002/pmj.20041> [Últim accés: 10-12-2025]
- [2] Forbes. *10 Best Task Management Software*. Forbes.com, 2024. [En línia] Disponible a: <https://www.forbes.com/advisor/business/software/best-task-management-software/> [Últim accés: 15 Gener 2025]
- [3] Atlassian. *Manage Your Team's Projects From Anywhere | Trello*, Trello.com, 2024. [En línia] Disponible a: <https://trello.com/> [Últim accés: 15 Gener 2025]
- [4] Forbes. *Asana Review*. Forbes.com, 2024. [En línia] Disponible a: <https://www.forbes.com/advisor/business/software/asana-review/> [Últim accés: 15 Gener 2025]
- [5] Clickup. *Clickup | One app to replace them all*. Clickup.com, 2024. [En línia] Disponible a: <https://clickup.com/> [Últim accés: 10 Desembre 2024]
- [6] Monday.com. *Monday.com | Your go-to work platform*. Monday.com, 2024. [En línia] Disponible a: <https://monday.com/> [Últim accés: 10 Desembre 2024]

- [7] SAP. *Software de gestión para empresas y soluciones / SAP*. Sap.com, 2024. [En línia] Disponible a: <https://www.sap.com/spain/> [Últim accés: 15 Gener 2025]
- [8] Odoo. *Open Source ERP and CRM / Odoo*. Odoo.com, 2024. [En línia] Disponible a: <https://www.odoo.com/> [Últim accés: 23 Gener 2025]
- [9] Node.js. *Node.js JavaScript runtime*. Nodejs.org, 2024. [En línia] Disponible a: <https://nodejs.org/> [Últim accés: 02 Febrer 2025]
- [10] Vue.js. *The Progressive JavaScript Framework*. Vuejs.org, 2024. [En línia] Disponible a: <https://vuejs.org/> [Últim accés: 02 Febrer 2025]
- [11] Express.js. *Fast, unopinionated, minimalist web framework for Node.js*. Expressjs.com, 2024. [En línia] Disponible a: <https://expressjs.com/> [Últim accés: 02 Febrer 2025]
- [12] Amazon Web Services (AWS). *Amazon EC2 / Virtual Server Hosting*. Aws.amazon.com, 2024. [En línia] Disponible a: <https://aws.amazon.com/ec2/> [Últim accés: 02 Febrer 2025]
- [13] Amazon Web Services (AWS). *Amazon DocumentDB / Managed NoSQL Database*. Aws.amazon.com, 2024. [En línia] Disponible a: <https://aws.amazon.com/documentdb/> [Últim accés: 02 Febrer 2025]
- [14] Vuetify. *Material Design Component Framework*. Vuetifyjs.com, 2024. [En línia] Disponible a: <https://vuetifyjs.com/> [Últim accés: 02 Febrer 2025]
- [15] Cloudflare. *Cloudflare / DDoS Protection and Web Security*. Cloudflare.com, 2024. [En línia] Disponible a: <https://www.cloudflare.com/> [Últim accés: 02 Febrer 2025]

Apèndix

1.3 Demostració de funcionament

S'ha creat un usuari únicament per la demostració de l'aplicació. Aquest usuari és líder d'un equip, i l'única limitació que té és el canvi de contrassenya.

- Usuari: **usuaritest@fynix.pro**
- Contrassenya: **1234**

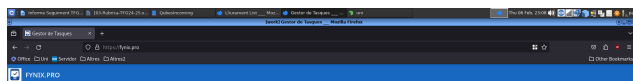


Fig. 2: Visualització de la pàgina d'inici de sessió

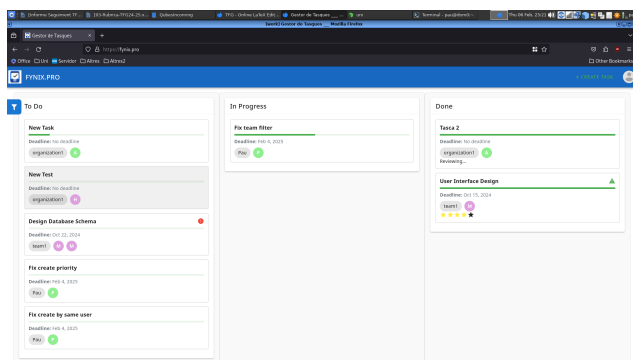


Fig. 3: Pàgina principal i petita demostració del funcionament de la taula Kanban de la pàgina inicial, en la que es poden desplaçar tasques d'un estat a un altre arrossegant entre columnes.

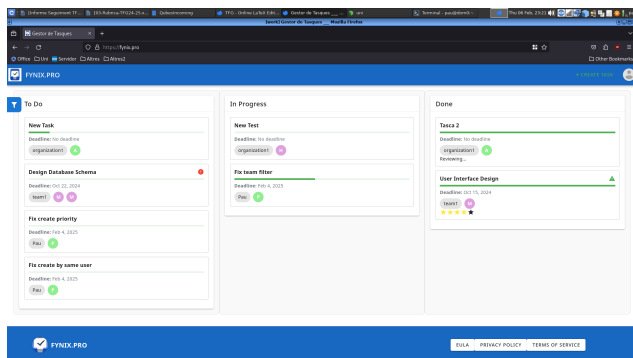


Fig. 4: Continuació de la demostració del funcionament de la taula Kanban de la pàgina inicial, en la que es poden desplaçar tasques d'un estat a un altre arrossegant entre columnes.

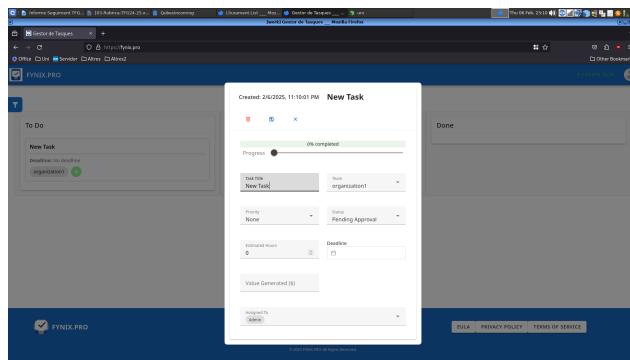


Fig. 5: Demostració de l'edició d'una tasca

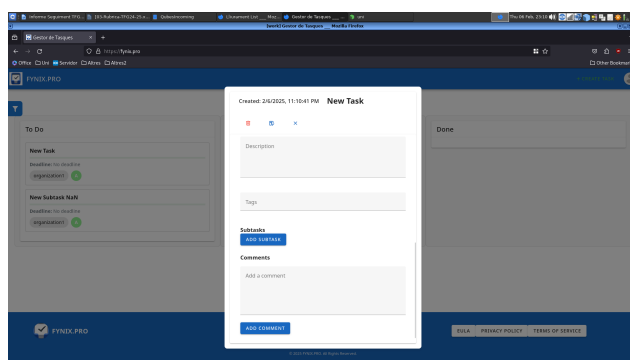


Fig. 6: Continuació de la demostració de l'edició d'una tasca

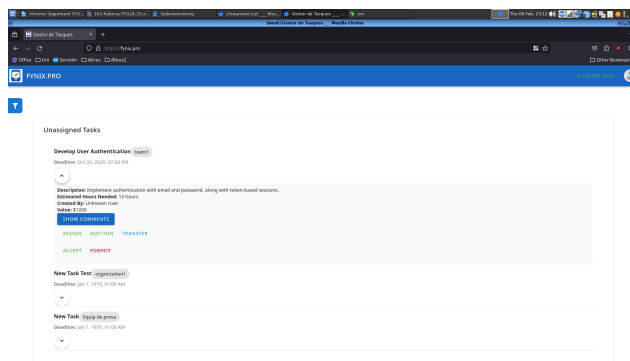


Fig. 7: Demostració de la bústia d'un usuari líder de varis equips. Secció de tasques per gestionar.

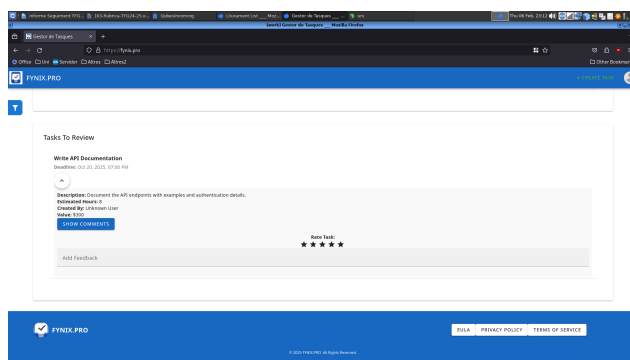
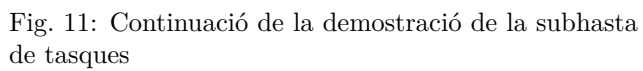


Fig. 8: Demostració de la bústia d'un usuari líder de varis equips. Secció de tasques per avaluar.



1.4 Documentació de la API del backend

Aquesta API s'executa al servidor backend de Node.js. S'encarrega de gestionar les peticions de xarxa de tots els usuaris que estan utilitzant l'aplicació web descarregada, verificar els seus permissos, i interactuar amb la base de dades quan sigui necessari.

Tota la informació s'envia en el cos de cada petició i en el format JSON, i tots els camps s'estructuren de la següent manera: {títol1:"dada", títol2:"dada"}. A més, tots els endpoints requereixen autenticació de l'usuari, excepte '/api/login' i '/api/logout'. Aquesta autenticació es produeix fent una petició a l'endpoint de login amb l'usuari i contrasenya, i es rep el JWT corresponent.

1.4.1 Autenticació i Sessions

Mètode	Endpoint	Funció
POST	/api/login	Inicia sessió amb userId i password, retorna un token JWT.
POST	/api/logout	Invalida el token de sessió de l'usuari.
POST	/api/change-password	Permet canviar la contrasenya.

1.4.2 Gestió de Tasques

Mètode	Endpoint	Funció
GET	/api/get-team-data	Retorna totes les tasques associades als equips de l'usuari autenticat.
POST	/api/get-users	Retorna el correu i nom d'usuari de tots els usuaris de tots els equips de l'usuari autenticat
POST	/api/get-history	Rep un timestamp i retorna tot l'historial dels equips de l'usuari a partir del timestamp rebut.
POST	/api/edit-task	Rep una tasca i la modifica o afegeix a la base de dades, comprovant quins camps pot modificar l'usuari segons el seus permisos.
POST	/api/add-subtask	Crea una nova subtasca i l'afegeix a la llista de subtasques del pare.
POST	/api/task-assign	Assigna una tasca a un usuari. Només els managers poden fer-ho.
POST	/api/task-forfeit	Rep una tasca, renuncia a la tasca i la retorna a l'equip de l'atribut receivedFrom o l'elimina si ja està a l'equip on s'ha creat i ho fa el manager
POST	/api/task-review	Rep una tasca valorada, i actualitza el valor de la valoració si l'usuari és manager o reviewer (de 0 a 5 estrelles).
POST	/api/task-accept	Rep una tasca, canvia l'estat d'una tasca a "To Do" i l'assigna a l'usuari.

1.4.3 Gestió d'Equips

Mètode	Endpoint	Funció
POST	/api/add-team	Rep un equip, crea el nou equip i l'afegeix com a subequip si té un pare.
POST	/api/update-team	Rep un equip i modifica l'equip si ho fa l'administrador.
POST	/api/add-user	Afegeix un usuari a un equip. Si l'usuari no existeix, se'n crea un de nou.

1.4.4 Subhastes i Licitacions

Mètode	Endpoint	Funció
POST	/api/task-auction-update	Rep una tasca, verifica si la subhasta ha acabat i assigna la tasca al millor aposta de hores necessitades.
POST	/api/task-auction	Rep una tasca i la passa a subhasta, assignant el tipus a "Auctioned".
POST	/api/place-bid	Rep una tasca i una aposta, afegeix la aposta de hores necessàries a la tasca, si encara està subhasta.

1.4.5 Transferència i Permisos d'Equips

Mètode	Endpoint	Funció
POST	/api/task-transfer	Rep una tasca i un equip, i mou la tasca al nou equip si és manager.
POST	/api/add-send-to-team	Rep un equip i un codi d'equip. Busca a quin equip correspon el codi d'equip, i l'afegeix a la llista d'equips als que es pot enviar.

1.5 Estructura de les dades a la base de dades

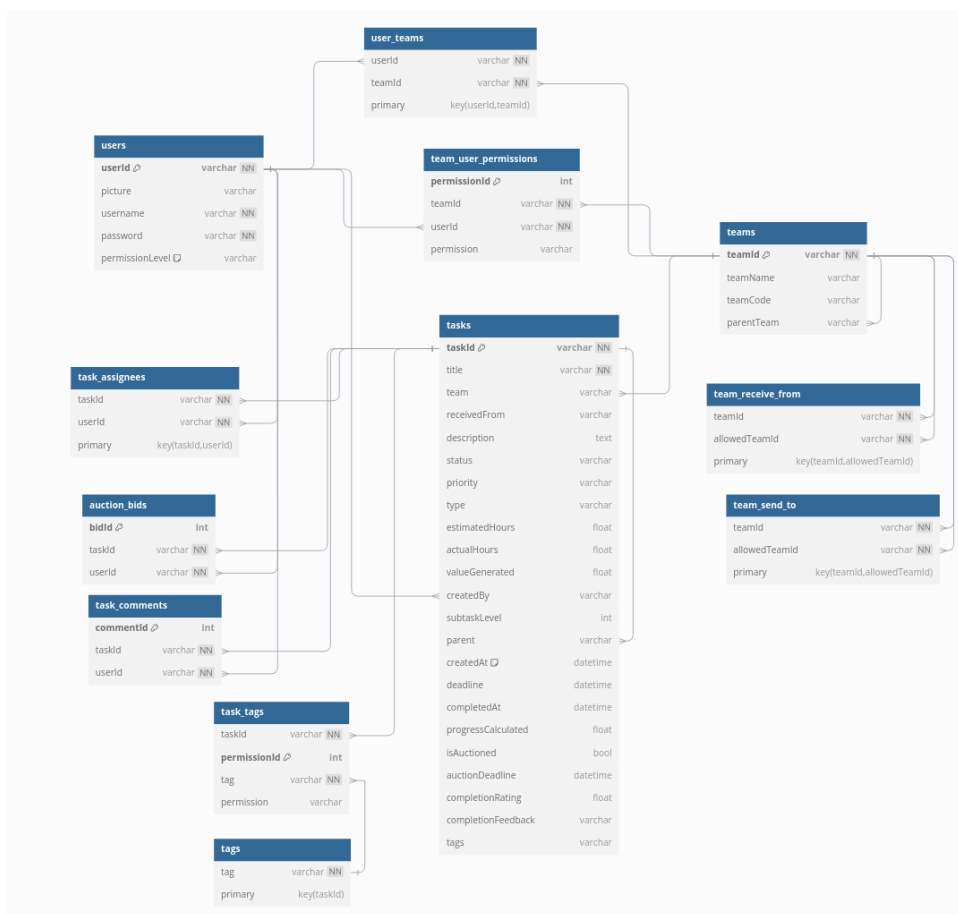


Fig. 12: Representació relacional de l'estructura de l'emmagatzematge de les dades de la base de dades. És la traducció més propera a estructura real, però no exacta ja que s'ha utilitzat una base de dades no relacional.

1.6 Diagrama de Gantt del projecte

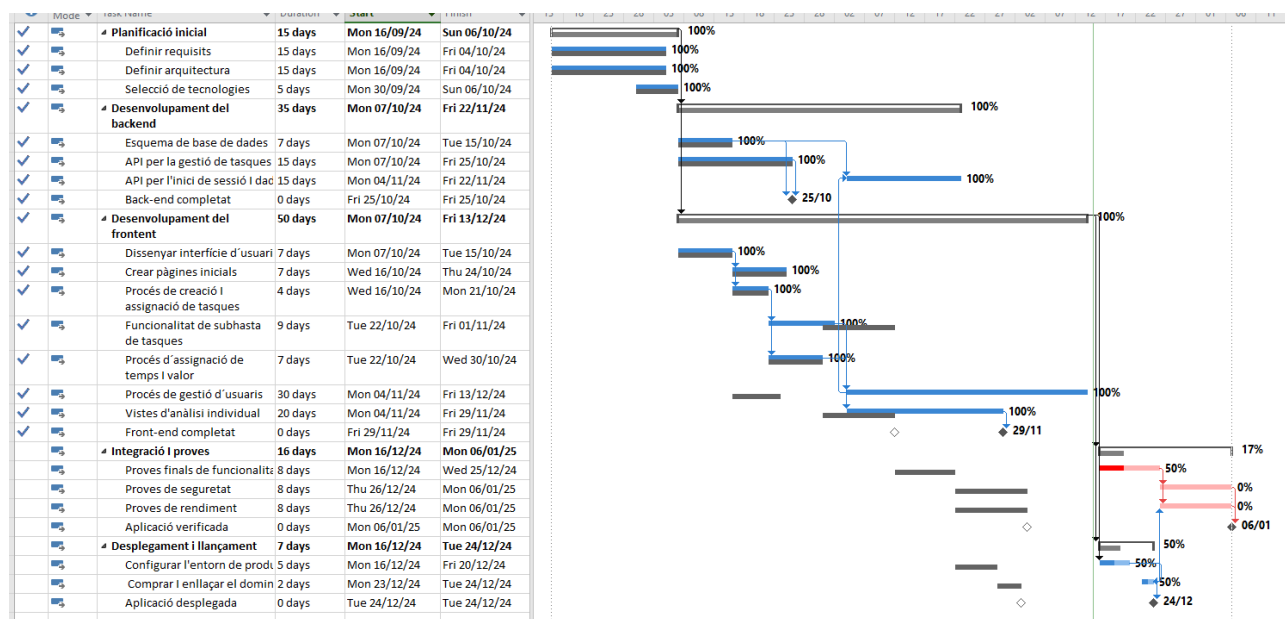


Fig. 13: Visualització del progrés del treball i els retards acumulats a data del segon informe de seguiment.