

ECOMENTOR

PROJECTE D'ENGINYERIA DEL SOFTWARE, Q2 2024-2025
MEMORIA
GRUP 22

Víctor Díez Serrano, Back-end Developer, Scrum Master Sprint2
Dídac Dalmases Valcàrcel, Scrum Master 1 & Front-end Developer
David Mas Escudé, Scrum Master 2 & Back-end Developer
Rubén Palà Vacas, Front-end Developer, Architectural designer & Scrum Master Sprint1
David Sanz Martínez, Back-end Developer & Scrum Master Sprint3
Neptune Christoper Lumayag Cartalla, Front-end Developer



ECOMENTOR	1
1. Requirements	4
1.1. General conception of the project	4
1.1.1 La aplicació	4
1.2. “NOT” list	5
1.3. Conceptual model	6
1.4. Taiga	6
1.5. Summary of product backlog in the document	7
1.6. Non-functional requirements	11
1.6.1 Performance Requirements	11
1.6.2 Security Requirements	11
1.6.3 Usability Requirements	12
1.6.4 Reliability & Availability Requirements	12
1.6.5 Scalability Requirements	12
1.6.6 Maintainability & Extensibility Requirements	12
1.6.7 Compliance & Legal Requirements	12
1.7. Treatment of transversal aspects	13
1.8. Third-party services	14
1.9. Real Stakeholders	15
1.9.1 Resultats enquesta i millores aplicades	16
1.10. Before and after Mock-ups	1
2. Methodology	1
2.1. Project management	1
2.2. Repository management	1
2.3. Communication within the team	1
2.4. Quality management	1
2.5. Testing strategy	1
2.6. Management of configurations	1
2.7. Interaction with colleagues	1
2.8. Bug management	1
2.9. NFRs treatment	1
2.10. Coding Assistant	1
3. Technical description	1
3.1. Overall conception of the architecture	1
3.1.1. Physical architecture	1
3.1.2. Architectural pattern(s) applied	1
3.2. Domain layer	1
3.2.1. Domain model diagram (optional)	1
3.2.2. Design patterns applied	1
3.3. Database diagram (UML)	1
3.4. Instrumentation and list of technologies	1
3.5. APIs	1

3.5.1 API nostre	1
3.5.2 APIs externes que utilitzem	1
3.5.3 Service Consumption	1
3.6. Data treatment	1
3.6.1. Recommendation algorithm	1
3.6.2. Càlcul dels certificats no oficials	1
3.7. Development tools and working environment	1

1. Requirements

1.1. General conception of the project

1.1.1 La aplicació

El nostre projecte té l'objectiu de desenvolupar una app per **millorar l'eficiència energètica dels edificis a Catalunya**, amb el propòsit de promocionar un estil de vida més **sostenible i ecològic**.

L'aplicació permet a tots els usuaris visualitzar de manera dinàmica en un **mapa** la informació energètica dels edificis del territori català -proporcionada pels certificats energètics-. Es podran veure diferents dades i també comparar dos edificis amb certificat energètic.

ECOMENTOR també permet als usuaris que no tinguin un certificat energètic la possibilitat d'obtenir un *pseudo-certificat digital* a través d'un petit **qüestionari**, on es valoraran superficialment certes característiques de l'edifici i es retornarà a l'usuari un resultat. A més a més, l'aplicació oferirà ajuda per obtenir el certificat.

Per als usuaris que ja tenen un certificat, es podran fer consultes a un **xatbot** de com millorar el seu immoble en termes de **sostenibilitat i consum**. Els usuaris també tindran a la seva disposició una **calculadora** on podran veure de forma **quantitativa** quin seria el benefici o el perjudici de canviar certs aspectes de l'immoble.

1.1.2 Dades rellevants

Per substanciar aquesta observació hem consultat diverses fonts oficials com l'Informe de l'Estat dels Certificats Energètics el 2023 i les dades parlen per elles mateixes. Veiem que més del 80% dels edificis nous obtenen les qualificacions A, B i C mentre que les edificacions existents només menys del 16% d'edificis registrats superen el nivell E.

També hem vist interessant cercar quants edificis no tenen un certificat energètic. Segons l'informe,

El nombre total de certificats registrats als corresponents registres autonòmics ascendeix a 5.978.358, dels quals els edificis existents representen el 97,58%. (Pàg. 2 de l'Estat de Certificats Energètics dels Edificis)

Com que la majoria d'edificis tenen certificat, també busquem el nombre

d'habitatges amb certificat. Encara que no hi hagi números oficials, hem trobat a CoHispania, la tasadora homologada oficial del Banc d'Espanya, que el 56% dels habitatges oferts a Espanya no té publicat el certificat energètic, sigui perquè no en disposen o perquè tenen una mala qualificació.

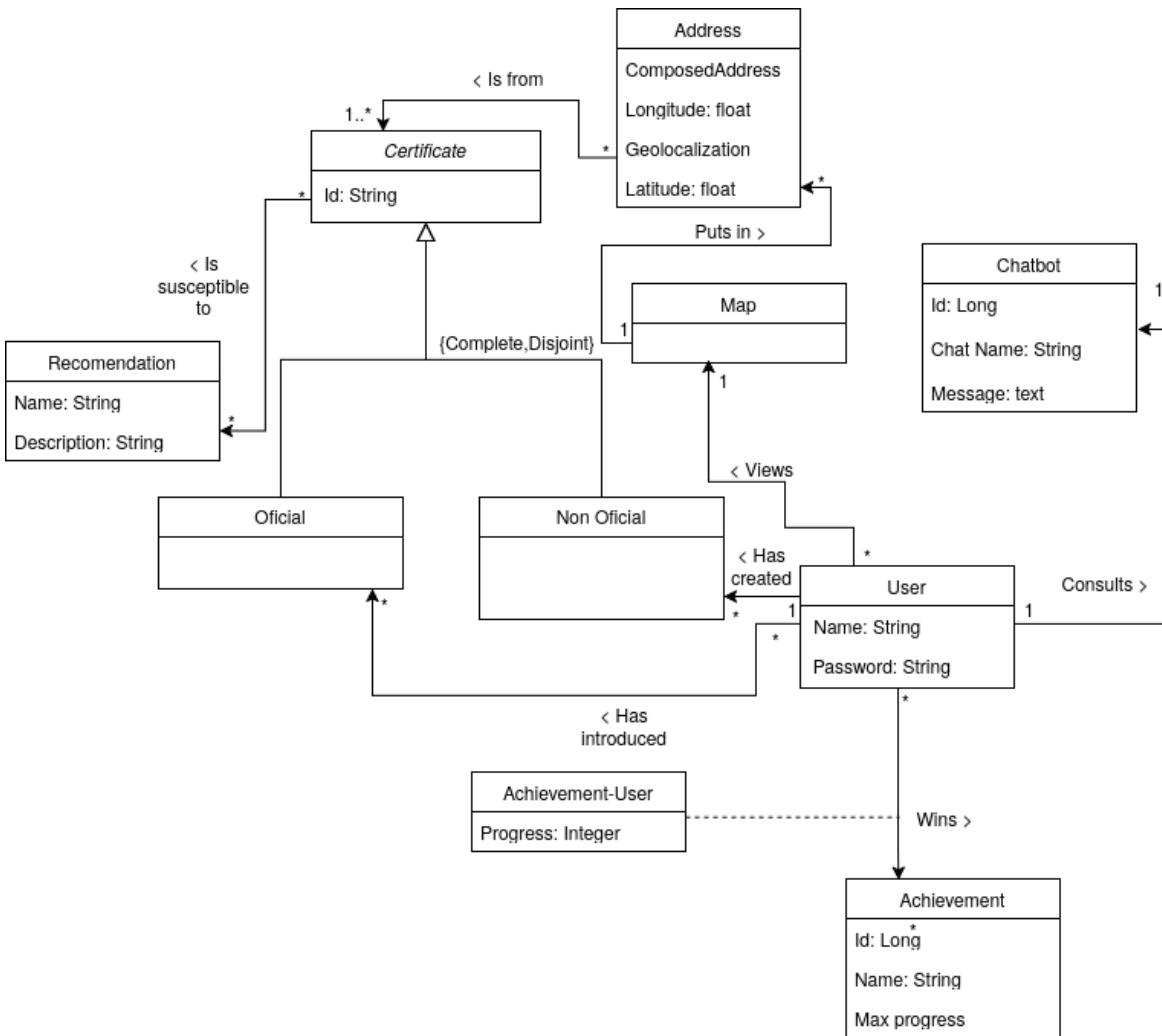
Amb totes aquestes dades, hem vist una oportunitat per fer una aplicació que pugui resoldre aquests problemes.

1.2. “NOT” list

Per acabar de clarificar les característiques del projecte, s'ha elaborat una *NOT LIST* per clarificar quina és la **identitat** de la nostra aplicació.

SÍ	NO
Mapa interactiu on veure els certificats elèctrics que tenen els edificis	Abast del mapa global (només a catalunya, Restricció Data set)
Mapa amb filtres per poder filtrar segons consum, nota del certificat, emissions	Actualització de les dades en temps real
Oferim un sistema per poder obtenir certificats electrics no oficials.	Oferim on comprar les millores esmentades en les recomanacions només diem el preu estimat que pot tenir
Oferim un sistema de recomanacions per poder millorar la nota del certificat/Consum elèctric indicant el preu estimat que suposa aplicar la millora	Oferim un sistema per obtenir certificats elèctrics oficials, que siguin reconeguts de manera oficial
Oferim un chatbot especialitzat per poder demanar recomanacions encara més personalitzades	Calendari per avisar quan renovar un certificat
Comparació energetica entre edificis	
Visualització de dades històriques sobre edificis amb certificats elèctrics a catalunya	
Sistema de gamificació	

1.3. Conceptual model



Disseny conceptual UML

1.4. Taiga

Amb l'objectiu de poder organitzar millor l'equip i assegurar-nos que tothom sàpiga clarament què ha de fer, hem decidit utilitzar **Taiga**, una eina de gestió de projectes àgil i intuïtiva. Aquesta plataforma ens permet assignar tasques, fer-ne el seguiment, establir prioritats i mantenir una visió global de l'estat del projecte en tot moment.

Taiga ens facilita treballar de manera col·laborativa, ja que tots els membres de l'equip poden veure l'evolució del projecte, actualitzar l'estat de les seves tasques i comunicar-se dins de la pròpia eina. A més, el seu enfocament en metodologies àgils, com Scrum o

Kanban, s'adapta perfectament a la nostra manera de treballar, afavorint una millor planificació i flexibilitat davant possibles canvis.

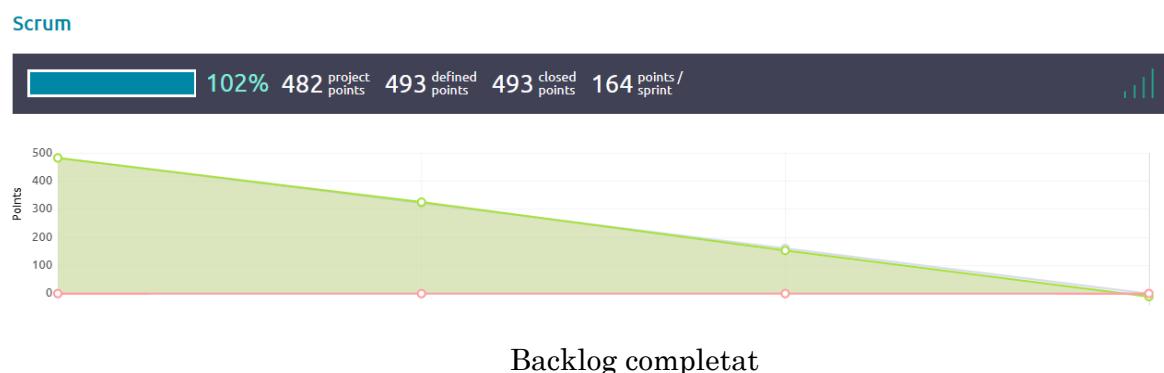
En definitiva, l'ús de Taiga ens ajudarà a ser més eficients, a millorar la comunicació interna i a garantir que el projecte avanci de manera ordenada i transparent per a tothom.

1.5. Summary of product backlog in the document

Hem dividit les èpies del nostre projecte en funcionalitats principals de l'aplicació, com és el mapa, els usuaris, els certificats oficials dels edificis, els certificats no oficials que oferim, el Xatbot, els trofeus per a la gamificació i la web admin per gestionar l'aplicació internament. Hem fet aquesta divisió per poder atomitzar les tasques per funcionalitats i d'aquesta manera poder separar millor les implementacions de les característiques de l'aplicació en els diferents sprints. Amb aquesta distribució ens assegurem que hi hagi una bona repartició de feina i cap sprint es quedi coix respecte a un altre.

Amb el nom de la tasca podem saber si es tracta d'una del front o del back, a més cada una disposa de la seva DoD en la descripció amb comentaris que reforcen el desenvolupament de cada tasca. Pel que fa a les Històries d'Usuari, també compten amb la seva frase <Com a [rol o tipus d'usuari], vull [acció o funcionalitat] per a [benefici o raó]>, junt amb el criteri d'acceptació per tal de mantenir un estàndard en la finalització d'aquestes.

Totes les tasques han estat completades amb èxit



NAME	PROJECT	SPRINT
☰ ▾ ● #1 Gestionar Mapa		
#6 Visualizar mapa ●		Sprint 1
#8 Filtrar Mapa ●		Sprint 1
#16 Buscar localització ●		Sprint 1
#290 MultiFiltratge ●		Sprint 2
Taiga Èpica Gestionar Mapa		

☰ ▾ ● #4 Gestió d'Usuaris
#22 Registre d'usuari ●
#65 Iniciar sessió ●
#67 Actualitzar perfil ●
#69 Eliminar compte ●
#231 Escollir preferència d'idioma ●
#235 Navegació de l'app ●
#238 Refutació ●
#338 Gestió de notificacions ●
#339 OAuth de tercers ●

Taiga Èpica Gestió d'usuaris

☰ ▼ ● #2 Gestionar Edificis

#10 Conseguir y mostrar info edifici(s) ●

#13 Comparar info edificis ●

#20 Consultar dades històriques ●

#285 Creació API per a tercers ●

Taiga Èpica Gestionar Edificis

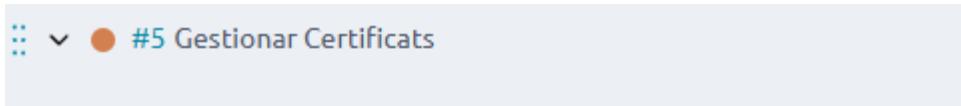
▼ ● #18 Gestionar Chatbot

#147 Utilitzar xatbot ●



Sprint 2

Taiga Èpica Gestionar Chatbot



#19 Visualitzar certificat

#24 Assistir en certificat (calculadora)

#112 Vincluar certificat oficial

#17 Obtindre recomanacions

#328 Millorar Calcular certificat no oficial

#334 Integrar calculadora en recomanacions

Taiga Èpica Gestionar Certificats



#190 Desbloquejar Trofeu

#284 Seguiment trofeu

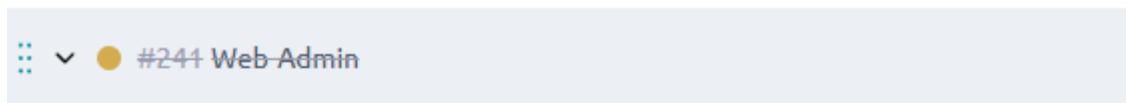
Taiga Èpica Gestió Trofeu



#228 Visualitzar esdeveniments culturals

#286 Integració API tercers

Taiga Èpica Gestió de Calendari



#242 Creació entitats

#244 Set up deployment

#275 Creació de la web admin

#329 Augmentar seguretat aplicació

Taiga Èpica Web-Admin

1.6. Non-functional requirements

1.6.1 Performance Requirements

- El sistema ha de recuperar i mostrar les dades de certificats energètics en un termini de vuit segons com a màxim en condicions normals de la xarxa.
- La interície del mapa ha de respondre a les interaccions de l'usuari (zoom, panoràmica, filtre) en cinc segons com a màxim.
- L'algoritme de recomanació per millorar l'eficiència energètica hauria de retornar suggeriments en un termini de deu segons com a màxim.

1.6.2 Security Requirements

- L'autenticació d'usuari ha de ser compatible amb OAuth 2.0 i la validació de testimoni basada en JWT.
- Totes les dades delicades dels usuaris s'han de xifrar en repòs i en trànsit mitjançant el passwordencoder de spring framework.
- El sistema hauria d'implementar el control d'accés basat en rols (RBAC) per restringir les modificacions de dades.

1.6.3 Usability Requirements

- L'aplicació hauria de tenir una interfície d'usuari intuïtiva amb un tauler de control fàcil d'utilitzar i coherència estètica entre pantalles.
- Els usuaris haurien de poder completar les tasques bàsiques (p. ex., comparar edificis) en cinc passos o menys.

1.6.4 Reliability & Availability Requirements

- El sistema hauria de mantenir un temps d'activitat del 85%.

1.6.5 Scalability Requirements

- El sistema hauria de gestionar 20 usuaris simultàniament sense una degradació significativa del rendiment.
- El sistema hauria d'emmagatzemar i recuperar de manera eficient les dades històriques dels certificats energètics durant almenys cinc anys.

1.6.6 Maintainability & Extensibility Requirements

- La base de codi ha de seguir principis d'arquitectura neta mitjançant Linter i estar ben documentada amb Swagger.
- Les proves unitàries haurien de cobrir almenys el 75% de les funcionalitats crítiques.

1.6.7 Compliance & Legal Requirements

- El sistema hauria de registrar les interaccions i garantir la llei orgànica de protecció de dades.

1.7. Treatment of transversal aspects

Aspect	Description	Current status
Geolocalització	Mostrem un mapa amb diferents punts que mostren els certificats energètics dels edificis.	Implementat en l'sprint 1.
Xarxes socials	Login. Poder compartir trofeus a les xarxes socials con Instagram, Facebook, X...	Implementat en l'sprint1, falta implementar les opciones de login amb google. Es farà en l'sprint 3.
Xat	La nostra aplicació tindrà un xatbot per poder fer consultes sobre certificats elèctrics.	Implementat a l'sprint 2.
Gamificació	Trofeus per a l'usuari quan completa tasques relacionades amb els certificats.	Implementat en l'sprint 3.
Stakeholders reals	Farem una reunió amb possibles stakeholders en la qual els hi ensenyarem la app i els hi demanarem recomanacions	Hem recollit feedback per part dels alumnes de 2n de batxillerat d'una escola a través d'una demostració i enquesta
Refutació	Bloquejar comptes que envien molts missatges al xat o que utilitzen paraules malonants	Implementat en l'sprint 2.
Calendari	La idea és tindre un calendari que mostri activitats relacionades amb el medi ambient i l'aprofitament de l'energia aprofitant la api de un altre grup	Implementat en l'sprint 3 perquè depèn de la API de 3ers
Web-app admin	Tindrem un apartat on només podrà accedir l'admin per fer CRUD d'entitats del sistema i poder bloquejar usuaris.	Implementat en l'sprint 2
Multiidioma	Possibilitat de canviar l'idioma de l'app entre. català, castellà i anglès	Implementat en l'sprint 1

1.8. Third-party services

S'ha realitzat un acord on EcoMentor, nosaltres, consumirà el servei ofert per Agendados i oferirà el seu servei a Ventus. Com es pot veure a les següents imatges, s'ha realitzat un contracte per arribar a un acord i mantenir-lo durant el desenvolupament, on ha estat firmat per ambdues parts

 **Dídac Dalmases Valcárcel** <didac.dalmases@estudiantat.upc.edu>
per a Jashanpreet ▾

Buenas Jashan,
Adjunto el contrato de la API que proporcionaremos, firmado por mi, representante de EcoMentor.
Muchas gracias,
Dídac.

Missatge de Jashanpreet Singh Kaur <jashanpreet.singh@estudiantat.upc.edu> del dia dv., 28 de març 2025 a les 13:10:

Un fitxer adjunt • Escanejat per Gmail ⓘ



Correu firmant l'acord de la nostra API per a Ventus.



David Comino Muriano
per a mi ▾

Buenas Dídac,
te devuelvo el contrato firmado.
Saludos,
David Comino Muriano

Missatge de Dídac Dalmases Valcárcel <didac.dalmases@estudiantat.upc.edu> del dia dj., 27 de març 2025 a les 23:11:

Un fitxer adjunt • Escanejat per Gmail ⓘ



Correu de Agendados firmant l'acord de la seva API per a EcoMentor.

Els contractes es poden trobar a la carpeta ContractesAPI de l'entrega del sprint 1. En aquests, s'acorda que de la nostra part, oferirem un endpoint per a Ventus on donades unes coordenades (o una altra mesura d'àrea, no especificat al contracte), retornarem els edificis amb millor qualificació que es trobin en tal zona. Degut a que podem tornar tant com informació basada en la localització com dades ja sobre l'energia i emissions, no serà un problema adaptar-nos.

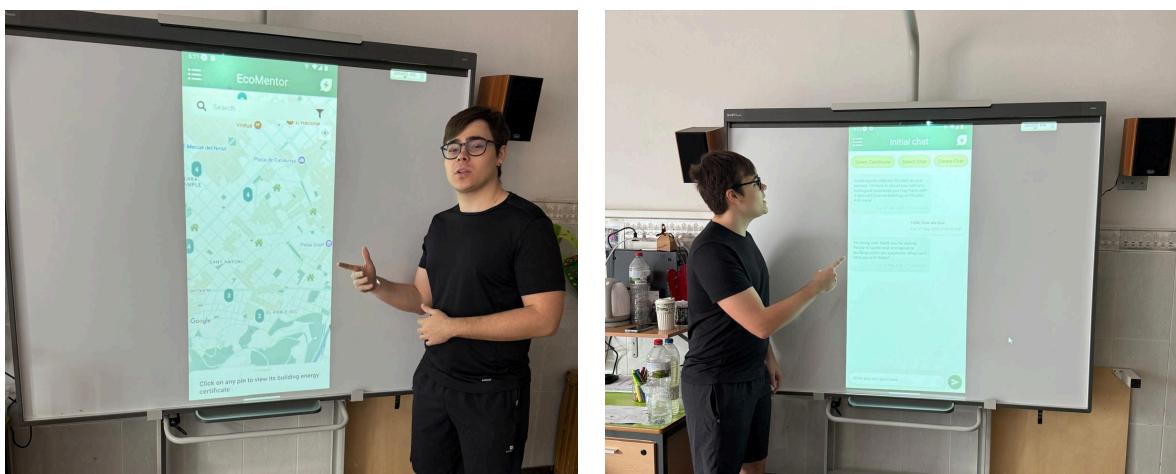
Per la part de l'API que consumirem, hem acordat amb Agendados que la seva API, donat una àrea definida per una latitud i longitud mínimes i màximes, obtenir events

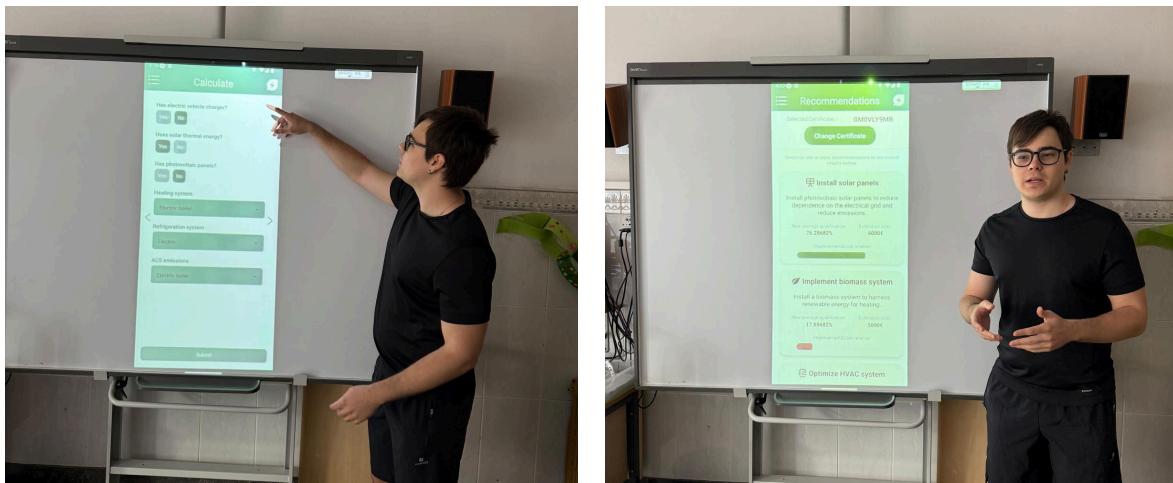
que passaran en aquella zona i la seva informació, d'aquesta manera poder mostrar la informació al nostre mapa o als edificis d'aquest.

1.9. Real Stakeholders

Per poder rebre feedback per millorar la nostra aplicació vam pensar que seria un salt de qualitat poder presentar el nostre projecte a algun sector que tinguéssim accés i que la proposta pogués ser interessant. Com que teníem contacte amb l'escola on alguns membres de l'equip havien cursat l'ESO i batxillerat, vam veure que seria una bona idea plantejar als professors i als alumnes la nostra idea i d'aquesta manera conscienciar sobre medi ambient i el consum energètic alhora que rebem opinions i propostes per a la nostra aplicació.

Tot seguit vam contactar amb el centre Maristes Rubí, i vam posar data un dia per poder fer una presentació. Durant, l'exposició vam explicar una petita introducció del nostre projecte i de la situació actual energètica, a continuació, vam ensenyar a l'alumnat els una petita demo per tal que poguessin apreciar la navegabilitat dels menús i algunes funcions bàsiques. Per acabar, vam passar un mòbil amb una build del programa perquè poguessin provar l'aplicació. Finalment, vam deixar-los una petita enquesta on podien deixar suggeriments, millores i la seva opinió.





Presentació de l'aplicació a Stakeholders Reals

Tota la informació recollida a través del feedback rebut serà de gran utilitat per introduir millors en el tercer sprint del projecte. Tindrem molt en compte les opinions i suggeriments dels participants, especialment pel que fa a canvis en com es veu l'aplicació i en les funcionalitats principals, amb l'objectiu de fer-la més intuïtiva, útil i adaptada a les necessitats reals dels usuaris.

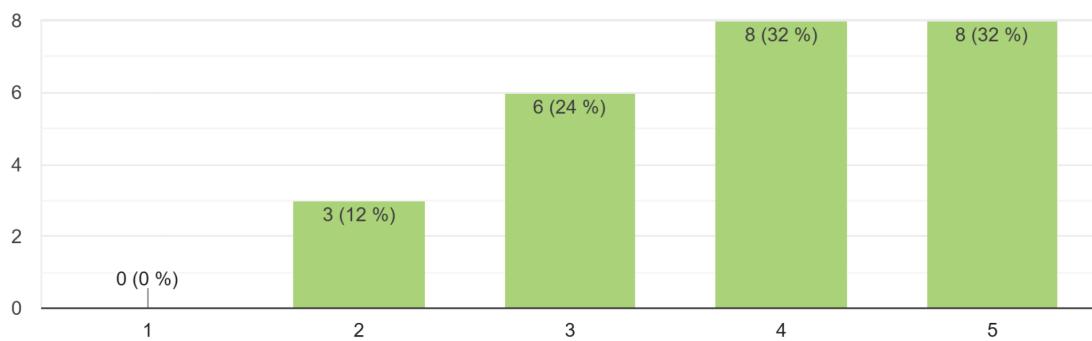
Link de l'enquesta: <https://forms.gle/7zFKMm5cNi9zHCqMA>

1.9.1 Resultats enquesta i millores aplicades

L'enquesta es va passar als alumnes d'una classe de batxillerat i a alguns professors del centre, aquests són els resultats de l'enquesta i el que podem aprofitar.

Quin és el teu nivell de satisfacció amb la pantalla d'inici de sessió?

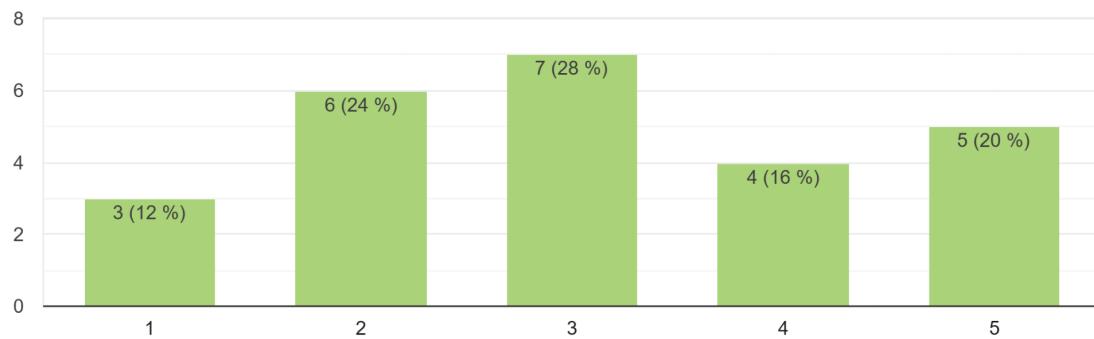
25 respuestas



Estadístiques login

Veiem que en general estan força satisfets amb la manera d'iniciar sessió a l'aplicació. Ens comenten que és simple i completa, i cap altre comentari destacable.

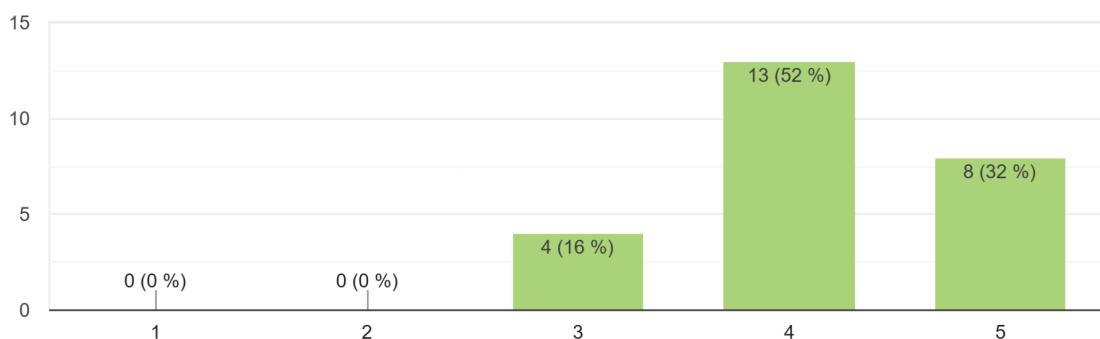
Quin és el teu nivell de satisfacció amb la visualització del mapa i els certificats energètics?
25 respuestas



Estadístiques mapa

Aquí observem que hi ha opinions diverses i que el com es veu el mapa no és del grat de tothom. Tenim comentaris de l'estil “Massa cases alhora”, “Potser es veuen massa edificis a la vegada”, “Els pins del mapa estan molt saturats”, “Veig molts certificats i em costa clicar”.

Quina ha estat la teva experiència en la secció de perfil?
25 respuestas

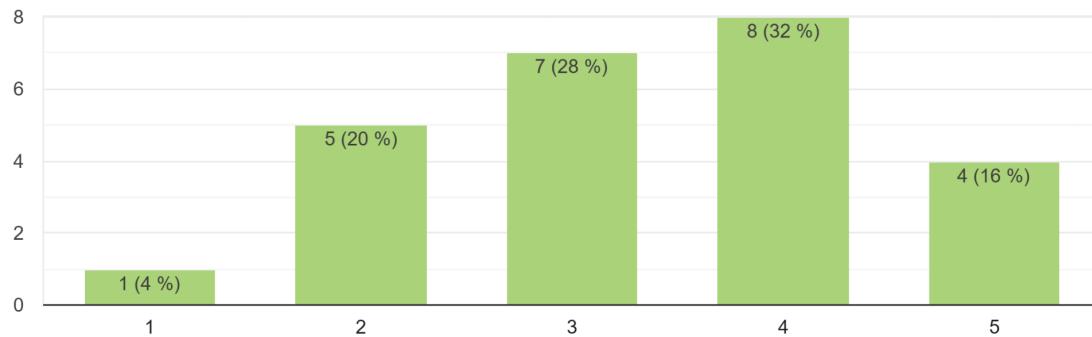


Estadístiques perfil

Res destacable a la secció de perfil, els ha agratjat en general, tot i això poques persones han provat de vincular un certificat, encara que els que ho han aconseguit els ha estat senzill.

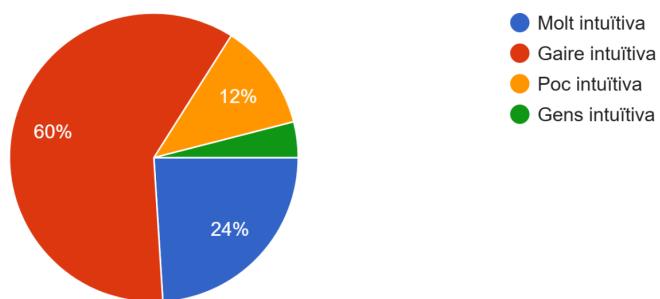
Quin ha estat el teu nivell de satisfacció amb la navegabilitat entre els menús de l'aplicació?

25 respuestas



Et sembla la interfície prou intuïtiva?

25 respuestas

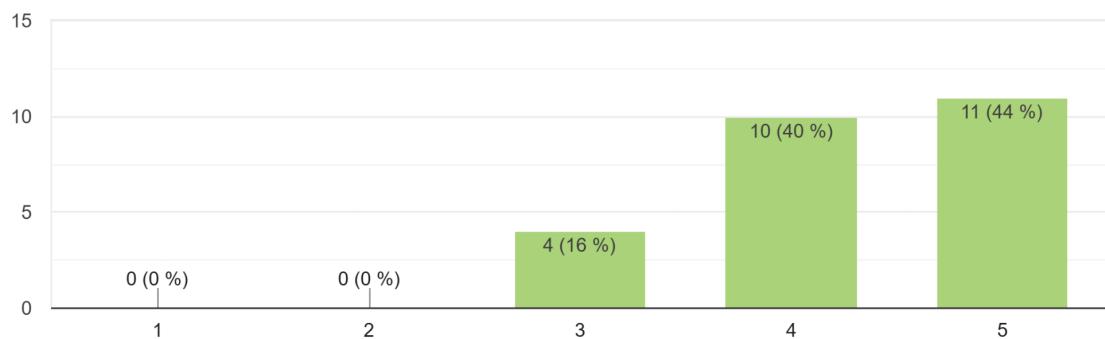


Estadístiques interfície

Podem observar que la gran part de l'alumnat i els professors els ha agratjat la manera de moure's per l'aplicació i la gran majoria concorden que l'aplicació és gaire intuïtiva.

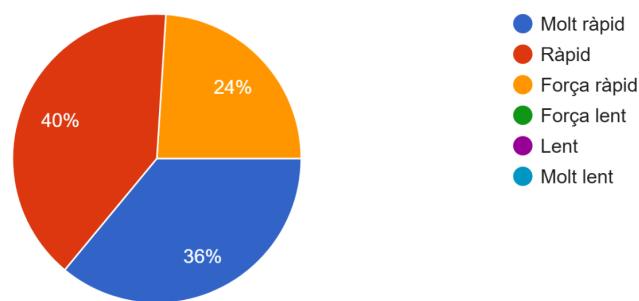
Quin ha estat el teu nivell de satisfacció amb la calculadora de certificats?

25 respuestas



Què et sembla el temps de resposta de la calculadora?

25 respuestas

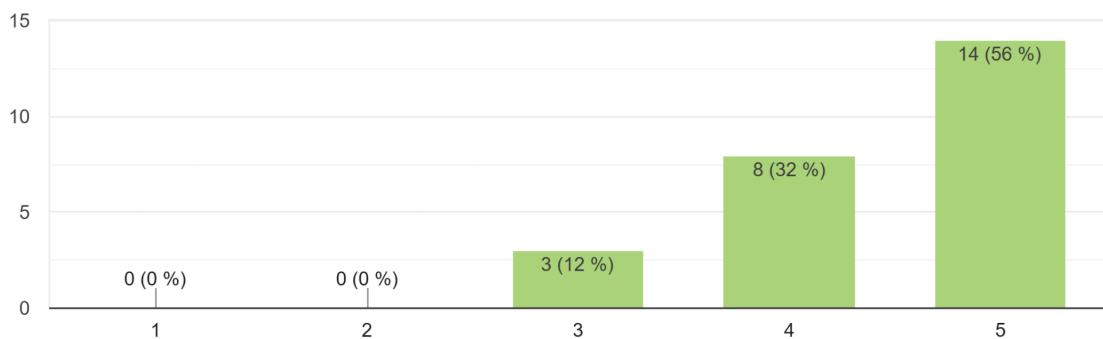


Estadístiques calculadora

Aquí observem que la calculadora en general agrada i que el temps de resposta compleix les expectatives. Cap comentari rellevant.

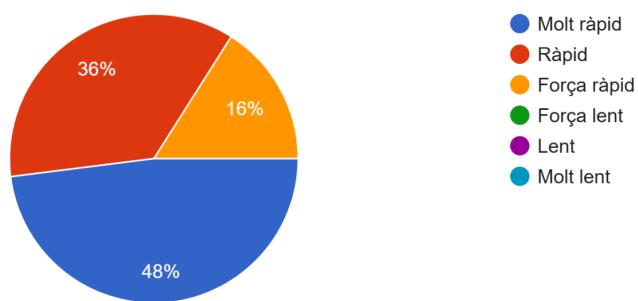
Quin ha estat el teu nivell de satisfacció amb les recomanacions de certificats?

25 respuestas



Què et sembla el temps de resposta de les recomanacions?

25 respuestas

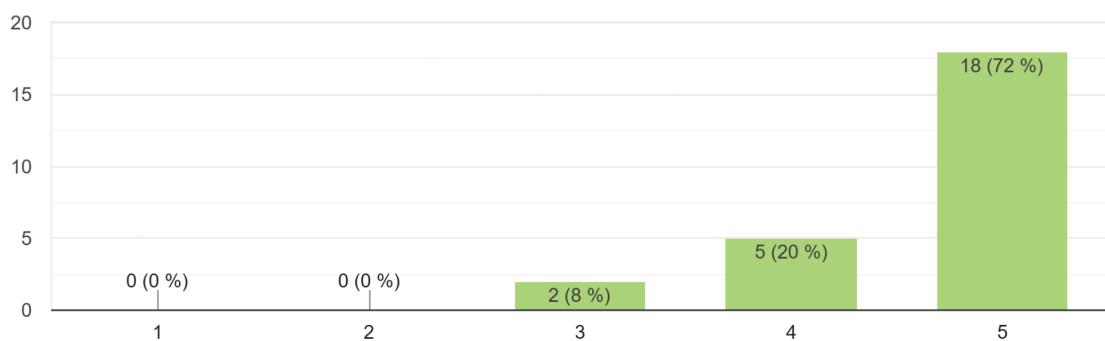


Estadístiques recomanacions

En general, també estan molt contents amb les recomanacions i el temps de resposta és l'esperat. Cap comentari rellevant.

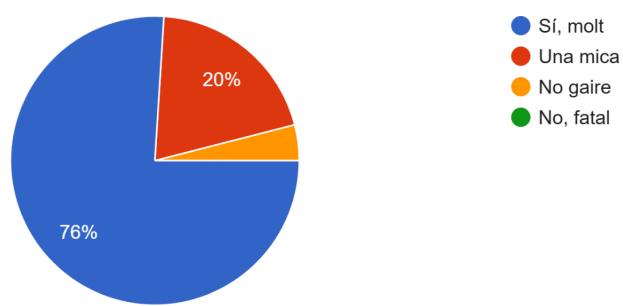
Quin ha estat el teu nivell de satisfacció amb EcoMentor ChatBot?

25 respuestas



T'han resultat útils les seves suggerències de millora?

25 respuestas

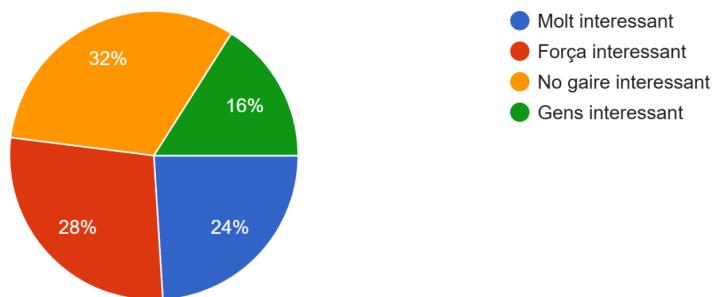


Estadístiques Xatbot

El Xatbot és una funcionalitat que li ha encantat a tothom i que concorden que els ha ajudat bastant. Cosa que és normal perquè en general els alumnes tot el que està relacionat amb intel·ligència artificial els apassiona i els interessa. Comentaris positius.

T'ha semblat interessant la secció històrica dels certificats?

25 respuestas

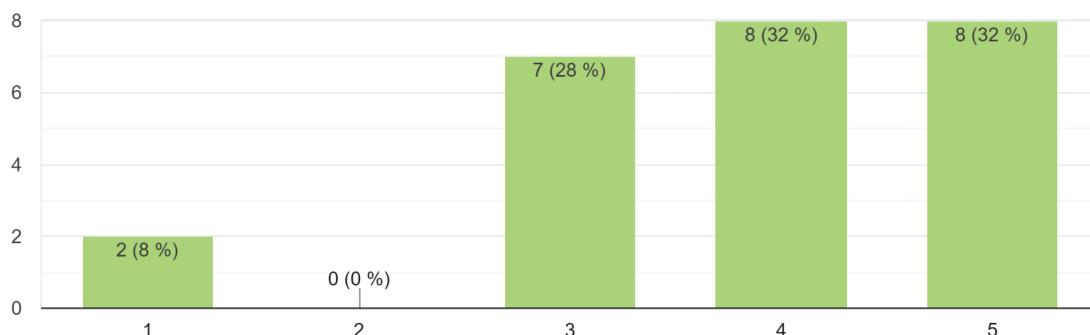


Estadístiques històric

Pel que fa a l'apartat d'històric, en general no acaba d'agradar i els sembla poc interessant. Hi ha comentaris de l'estil “No aporta res”, “No em diu res rellevant”.

Quina ha estat la teva experiència amb el filtratge i la comparació de certificats?

25 respuestas



Estadístiques filtratge i comparacions

Per acabar, tenim l'apartat de filtratge i comparacions que gran part dels enquestats els ha semblat un bon afegit. Cap comentari rellevant.

En resum, l'enquesta ens confirma que la base de l'aplicació (login, perfil, navegabilitat, calculadora, recomanacions i xatbot) funciona molt bé i és ben valorada per usuaris i professors. Els comentaris més rellevants han estat sobre el mapa (massa elements visibles alhora) i l'històric (informació sense valor percebut).

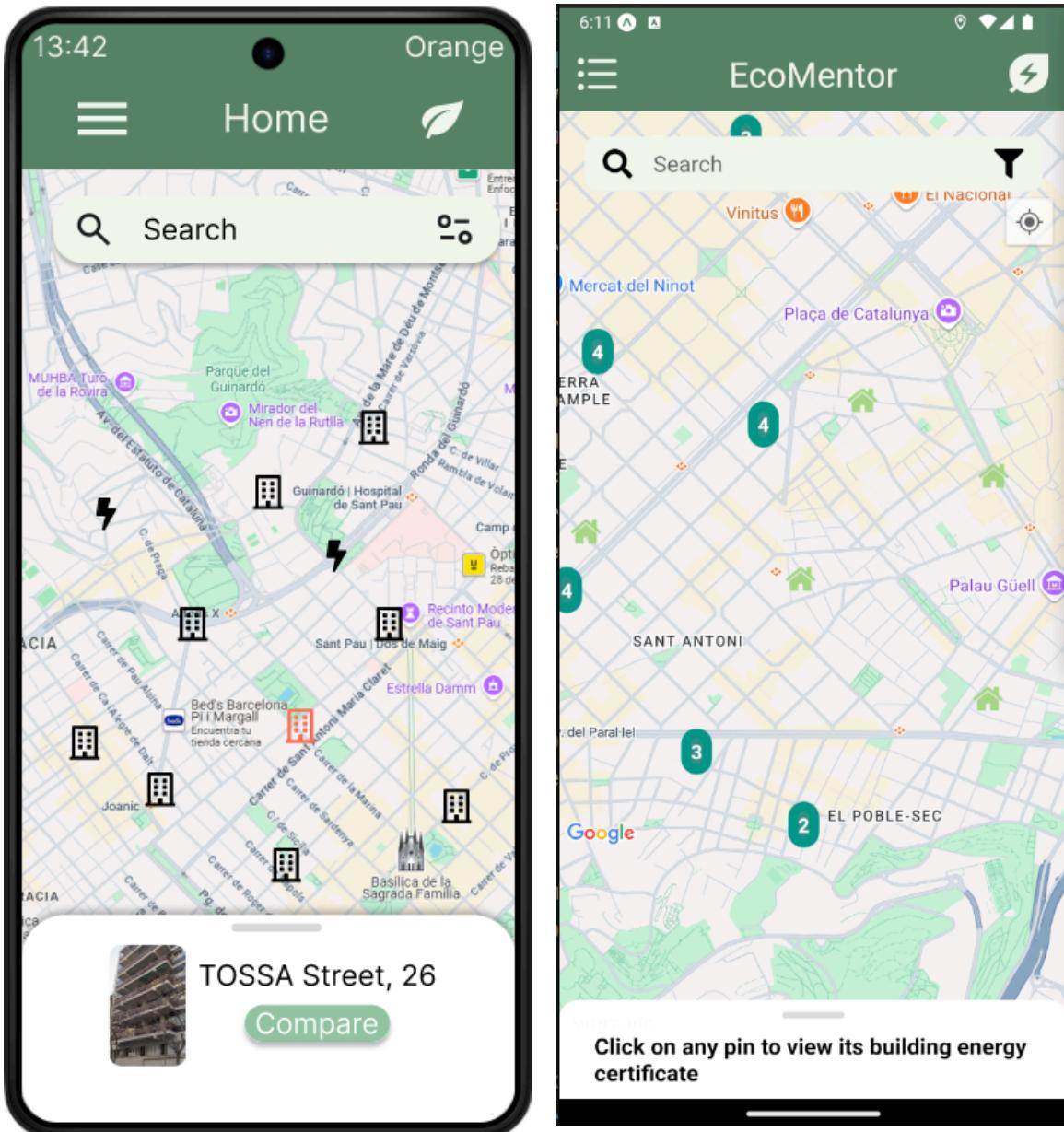
Gràcies al feedback rebut, ens va ajudar a adonar-nos del problema del mapa i d'aquesta manera millorar la visualització introduint **clustering**: ara els edificis s'agrupen per zones, reduint el soroll visual i facilitant el clic i el desplaçament. Pel que fa a l'històric, no veiem la necessitat de canviar-ho ja que es un apartat que suma a l'aplicació però és més un afegit secundari que no afecta a l'experiència de l'usuari.

En canvi, mantenim sense canvis les parts que ja funcionen bé (interfície general, xatbot, calculadora, recomanacions) i només afegirem petits ajustos preventius (un tooltip perquè més usuaris vinculin el certificat al perfil, seguiment de rendiment al xatbot, etc.).

A grans trets, l'experiència de compartir la versió pilot ha estat molt útil: hem pogut validar les decisions de disseny que ja eren encertades i detectar punts de millora que ara estem implementant. Continuarem recollint feedback i mesurant objectivament l'ús de cada funcionalitat (analítica interna) per assegurar que les properes iteracions aportin encara més valor als usuaris.

1.10. Before and after Mock-ups

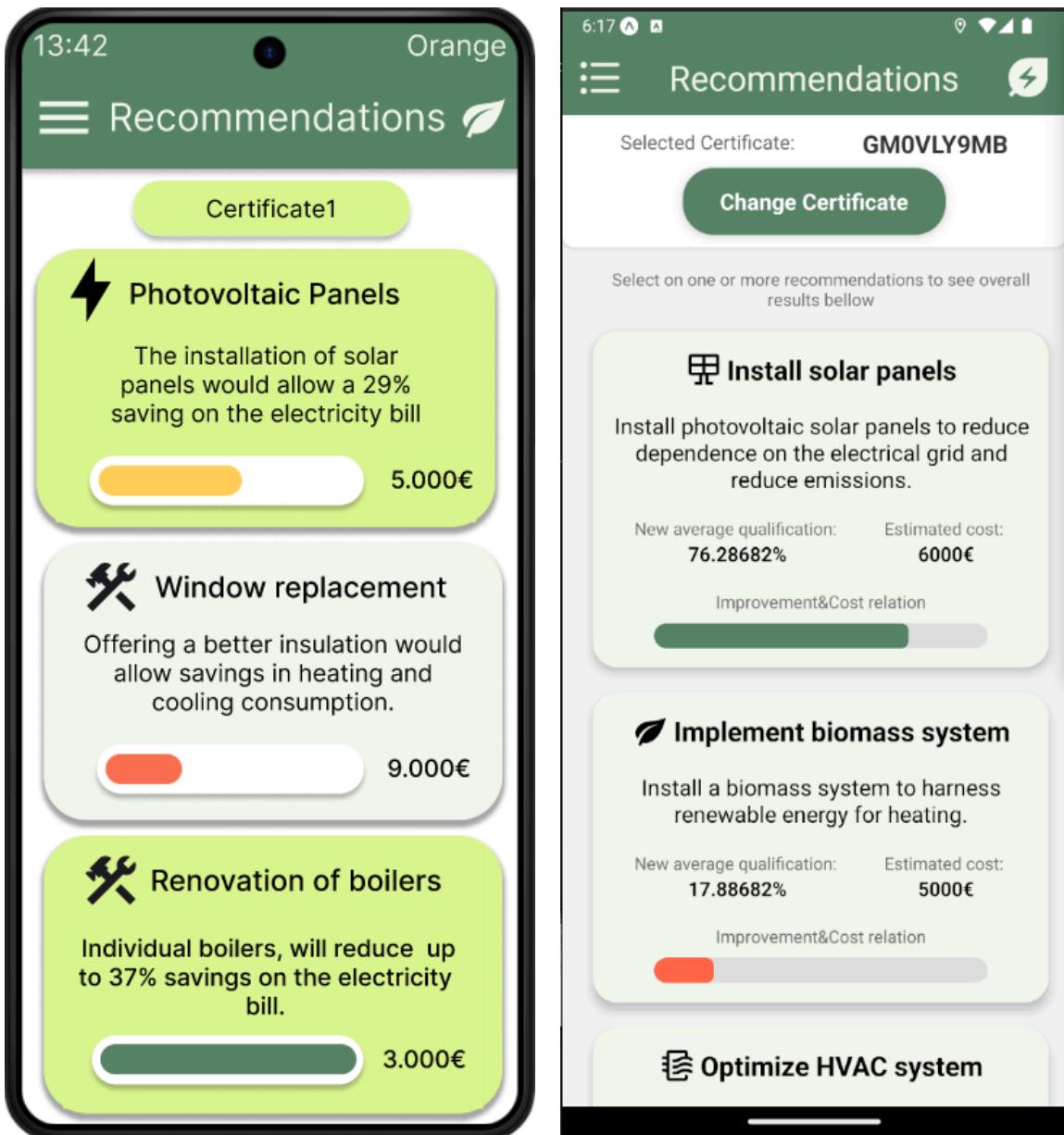
A continuació veurem com havíem plantejat inicialment el disseny de l'aplicació mitjançant els mockups i com ha acabat sent finalment després del desenvolupament. Aquesta comparativa ens permetrà observar els canvis, ajustos i millors que s'han realitzat al llarg del projecte, tant per motius tècnics com arran del feedback rebut.



Mapa abans i després

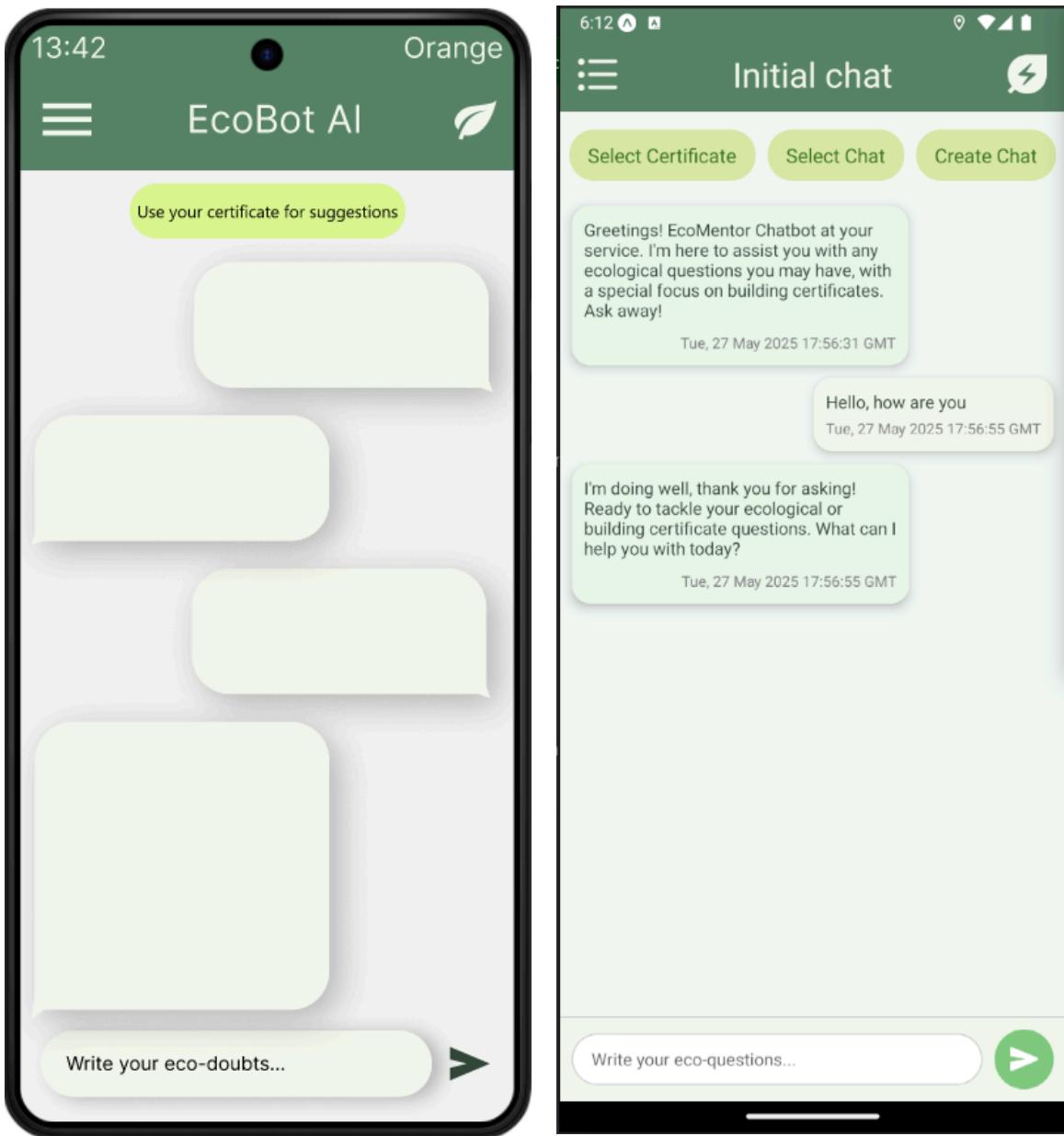
Com es pot veure a la imatge, a la versió final de l'aplicació hem implementat clustering per a mostrar d'una manera més sútil els punts al mapa. Això es deu a la llarga quantitat de certificats a tractar, millorant així l'eficiència i l'UX.

Un altre canvi ha estat el descartar l'ús de la imatge per a cada certificat, ja que l'obtenció de cada imatge en temps real podria ser un procés costós i no aporta informació rellevant a l'aplicació.



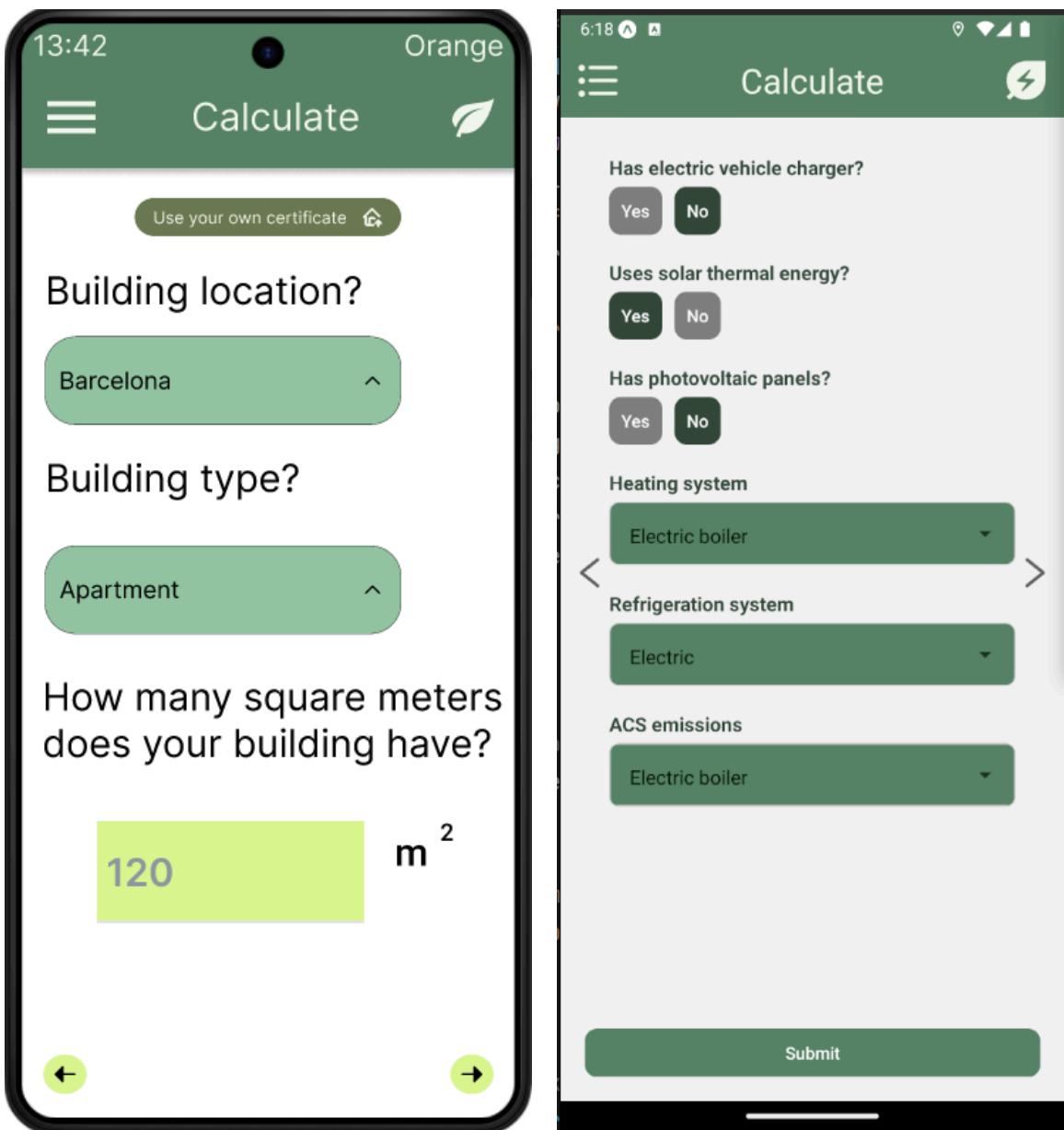
Recomanacions abans i després

Com es pot observar, la pantalla de recomanacions ha quedat practicament igual a excepció de petits retocs dels estils. A més, s'ha afegit la secció superior per poder veure el certificat actual, donant així més informació a l'usuari.



ChatBot abans i després

Al chatbot s'ha afegit l'opció de seleccionar un chat o crear-ne un, d'aquesta manera aportem més finestres de context a l'usuari i reduïm el consum dels tokens utilitzats per a la API de Gemini, evitant passar més informació segons cada missatge.



Calculadora abans i després

Per la calculadora, els paràmetres s'han mantingut bastant similars, els canvis més rellevants han estat els dels estils, ajustant els colors lleugerament i la posició de les fletxes.

2. Methodology

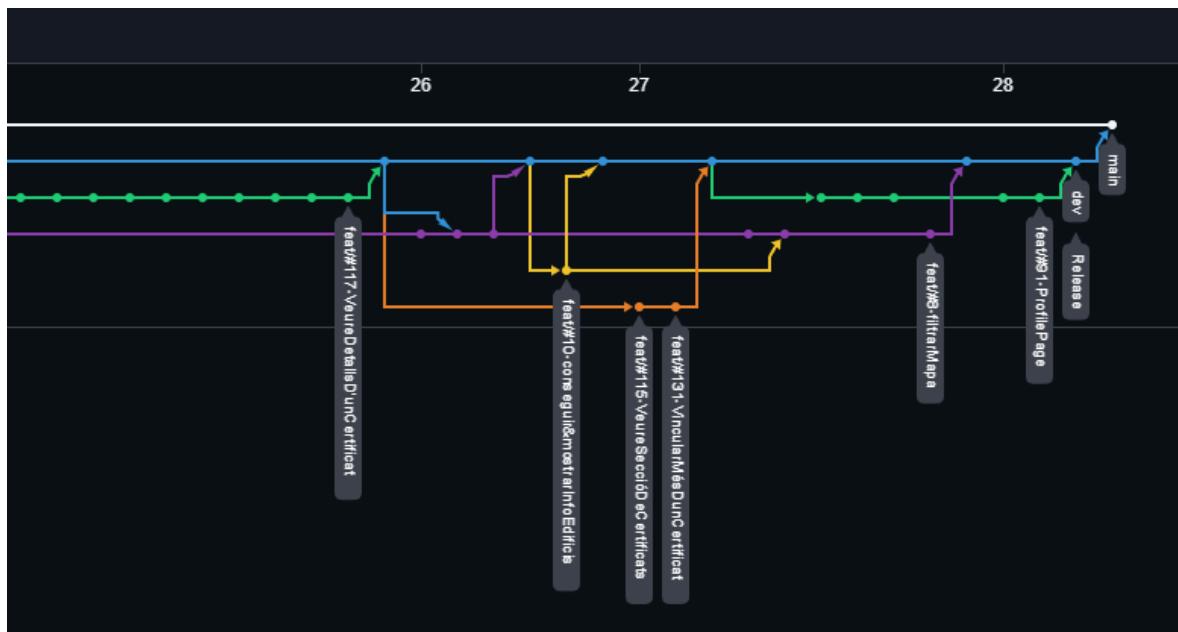
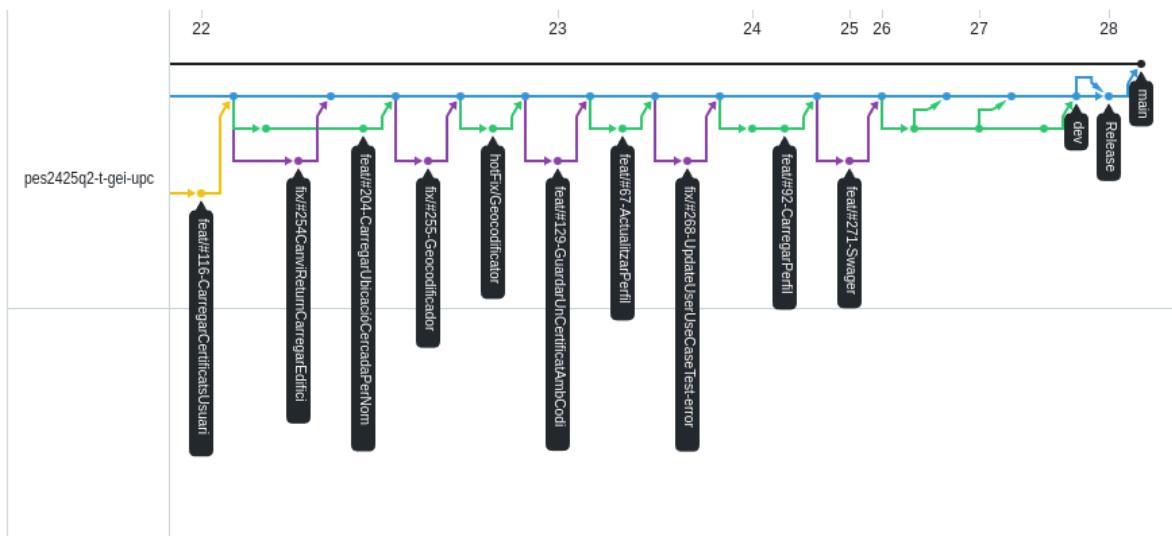
Durant el desenvolupament del projecte, hem seguit un enfocament àgil, utilitzant Scrum com a metodologia principal per organitzar el treball i gestionar els lliuraments de manera iterativa. Aquest enfocament ens ha permès adaptar-nos als canvis, validar progressivament les funcionalitats i mantenir una comunicació fluida dins de l'equip.

2.1. Project management

- Fem servir **Taiga** per gestionar el backlog i fer seguiment de les tasques amb **story points** per prioritzar i estimar l'esforç de cada història d'usuari.
- Les tasques al Taiga procuren tenir suficients dades per ser implementades, a més un cop completades, el temps que es va portar en realitzar-les.
- Els sprints es defineixen d'acord amb els objectius clau i les tasques es tanquen mitjançant revisió col·laborativa.
- Les reunions de revisió permeten detectar bloquejos i ajustar el roadmap.
- Tenim un document amb una guia pel nous desenvolupadors que s'incorporen al backend amb diverses guies com per l'estruccura dels tests, execució. En el archivo **guidelines.md**

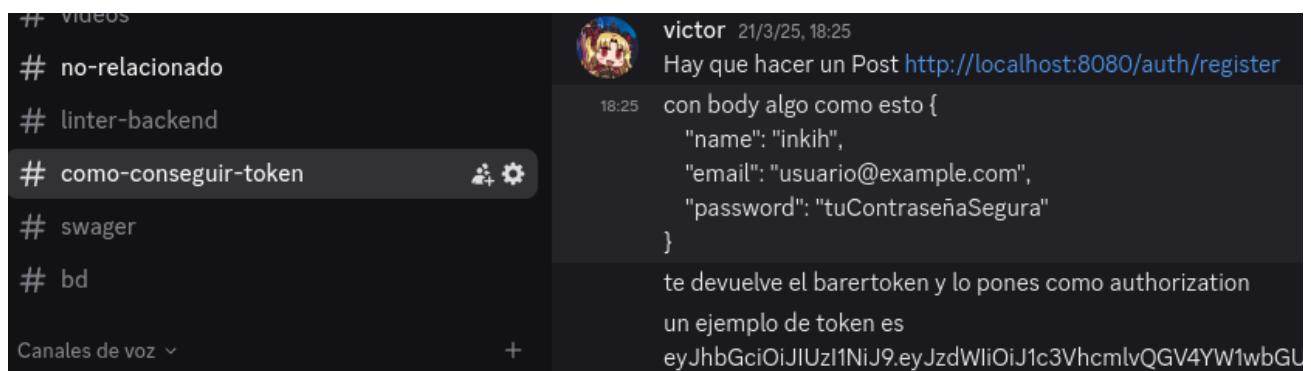
2.2. Repository management

- Implementem **GitHub** com a sistema de control de versions, seguint un model **Git Flow** amb branques separades per funcionalitats, desenvolupament i producció.
- Cada canvi passa per un procés de revisió via **pull requests** abans de ser fusionat a la branca principal.
- Hem decidit cada cop que desenvolupem una funcionalitat de l'aplicació començar una nova branca des de dev amb el nom **feat/#<nombreTaiga>-<nomTasca>** per tal de mantenir una coherència durant tot el desenvolupament.
- Un cop completa la tasca hem decidit que l'estruccura dels commits ha de ser **Task #<numTaiga> descripció de la tasca** per tal de saber el context del desenvolupament d'aquella funcionalitat.
- Si dona resulta que hi ha hagut una petita errada sobre una nova branca anomenada **bugFix/<IssueDelTaiga>**.
- Es fan commits freqüents per assegurar la traçabilitat dels canvis.



2.3. Communication within the team

- Per a la comunicació diària, fem servir un **grup de WhatsApp** per missatges ràpids i un **servidor de Discord** per reunions més extenses i diferents canals que es van creant cada cop que necessitem una consulta amb l'equip per a alguna determinada tasca o alguns errors que es produeixen en el desenvolupament a tall de fòrum, com a tutorials o explicant les crides a l'API això ens permet tindre sempre disponible i de manera endreçada els missatges relacionats amb un tema.



Exemple de tutorial al discord de Ecomentor

- Es realitzen **dailies informals** per sincronitzar l'equip i resoldre possibles bloquejos. Es realitzen cada cop que tenim classe presencial i ajuda a saber què és el que hem de fer d'ara en avanç i valorar la feina que s'ha anat realitzant.
- Taiga facilita la documentació i assignació clara de responsabilitats, de manera que cada tasca conté la seva DoD adjunta al seu objectiu, el seu requisit i les branques del Github afectades, per tal de saber quan s'ha de donar la tasca com finalitzada.

2.4. Quality management

- Hem establert **estàndards de codi** per garantir coherència i mantenibilitat.
- Les revisions de codi són una part essencial del procés, assegurant bones pràctiques i identificant possibles errors abans del merge.
- Integrem un linter i proves unitàries al nostre pipeline de GitHub per evitar pujar codi que no compili ni falli en executar-se. El linter (Tant el linter del back-end com el del front-end està definit a la pàgina 24-25) verifica errors d'estil i sintaxi, mentre que els tests unitaris validen el correcte funcionament del codi.

2.5. Testing strategy

- Utilitzem proves unitàries i integració contínua per validar les funcionalitats abans del desplegament, hem fet servir Mockito a l'hora de fer els tests unitaris en el back-end.
- Es farà servir **Jest** per a les proves unitàries del front-end validant la funcionalitat dels components i mòduls abans del desplegament.
- Tant en front com back es compleix que almenys un 75% per cent del codi estigui testejat per les nostres estratègies proposades.

2.6. Management of configurations

- Les configuracions es gestionen mitjançant **Docker** i variables per adaptar el projecte a diferents entorns sense comprometre la seguretat.
- Hem fet servir Docker per tal que no hi hagués problemes amb el projecte en múltiples sistemes i entorns de treball fent servir volums per emmagatzemar la base de dades amb diferent tamanys de dades per tal de fer diferents proves.
- La base de dades en la qual basem el nostre projecte és un arxiu CSV amb més d'un milió de files i seixanta-nou columnes, el qual suposa una càrrega de dades molt gran, a més moltes de les columnes era informació no rellevant o que no ens aporta cap mena d'informació en el que tenim plantejat per a la nostra aplicació. Per això, hem seguit unes certes mesures per tal de filtrar la informació que ens és important i netejar les taules per tal d'optimitzar la base de dades i evitar problemes amb alguns valors nuls.
- Tenim dos volums, ja que són els que tenim pensats. Cada volum seria pràcticament igual, però a l'apartat de certificats, un en tindrà molts més (més files) per testejar temps de càrrega de certes views i l'altre en tindrà menys per poder testejar i desenvolupar altres funcionalitats de forma més fluida. Aquest procés si va bé només cal fer-ho un cop i després és anar escollint el volum desitjat.

2.7. Interaction with colleagues

- Per interactuar amb els altres grups realitzem reunions presencials durant les hores de classe també, hem creat un grup de WhatsApp per poder parlar amb ells fora de l'horari escolar.
- També hem fet un contracte amb els altres dos grups per poder especificar el que ha de fer cadascun i no hi hagi problemes en el futur.

2.8. Bug management

- Els errors es registren a Taiga com a **incidències**, prioritzant-se segons l'impacte en el projecte.
- Obrim incidències al Taiga per tal de notificar el bug a l'equip i creem branques noves anomenades **bugFix/<IssueDelTaiga>** per tal de corregir-lo sense afectar la resta del projecte.

Issues

	subject or reference <input type="text"/>	<input checked="" type="checkbox"/> Tags	
TYPE	SEVERITY	PRIORITY	ISSUE
			#272 Arreglar filters
			#268 UpdateUserUseCaseTest fail
			#267 Fix insert in data base
			#256 Fix bottomsheet
			#255 fixGeocodificador

Alguns dels issues del taiga

2.9. NFRs treatment

Per cada apartat dels NFR mencionat en la secció [1.5](#) del document, aquesta és l'estrategia que estem seguint per complir amb l'establert i mantenir la integritat d'EcoMentor:

2.9.1 Performance requirements

- Avaluació de UI: Es definiran temps específics per a la resposta del mapa interactiu i l'algorisme de recomanació, amb proves d'usabilitat per verificar-ne el compliment.
-

2.9.2 Security Requirements

- Autenticació i autorització: S'implementarà OAuth 2.0 i JWT per validar usuaris, amb proves de penetració per identificar vulnerabilitats.

- Encriptació de dades: S'assegura que totes les credencials i dades delicades estiguin xifrades tant en trànsit com en repòs mitjançant Spring Security i PasswordEncoder.
- Control d'accés: S'aplica RBAC amb validacions automàtiques per garantir que només els usuaris autoritzats puguin modificar dades.

2.9.3 Usability Requirements

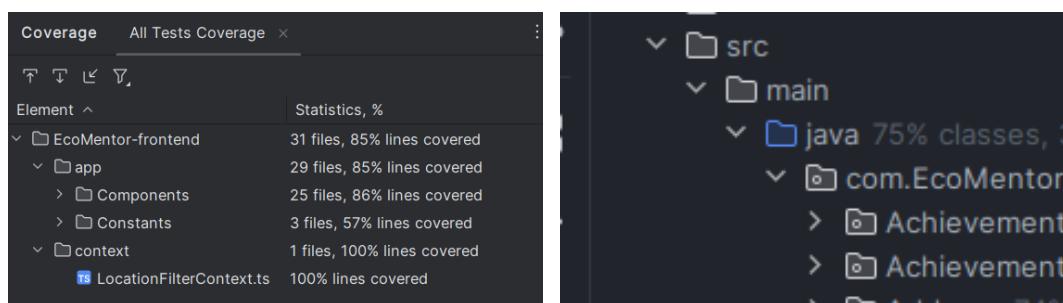
- Interfície intuïtiva: Es duran a terme proves d'usuari per avaluar la facilitat d'ús del dashboard i la coherència visual a les pantalles.
- Reducció de passos: Es mesuraran les tasques clau (comparar edificis, consultar recomanacions) per assegurar que es puguin completar en un màxim de cinc pasos.

2.9.5 Scalability Requirements

- Proves de càrrega: Se simularan 20 usuaris concurrents amb eines com ara Gatling per garantir un rendiment estable.
- Optimització d'emmagatzematge: Es dissenyarà un model de dades eficient per manejar 5 anys de registres de certificats energètics sense afectar-ne el rendiment.

2.9.6 Maintainability & Extensibility Requirements

- Codi net i documentat: S'aplicarà Linter per verificar la qualitat del codi i es documentaran les APIs amb Swagger.
- Cobertura de proves: Es mantindrà un 75% de cobertura en funcionalitats crítiques amb proves unitàries en back amb Mockito i en front amb Jest. Em comprovat amb el test coverage que facilita els IDEs de JetBrains.



Test coverage de més del 75%

2.10. Coding Assistant

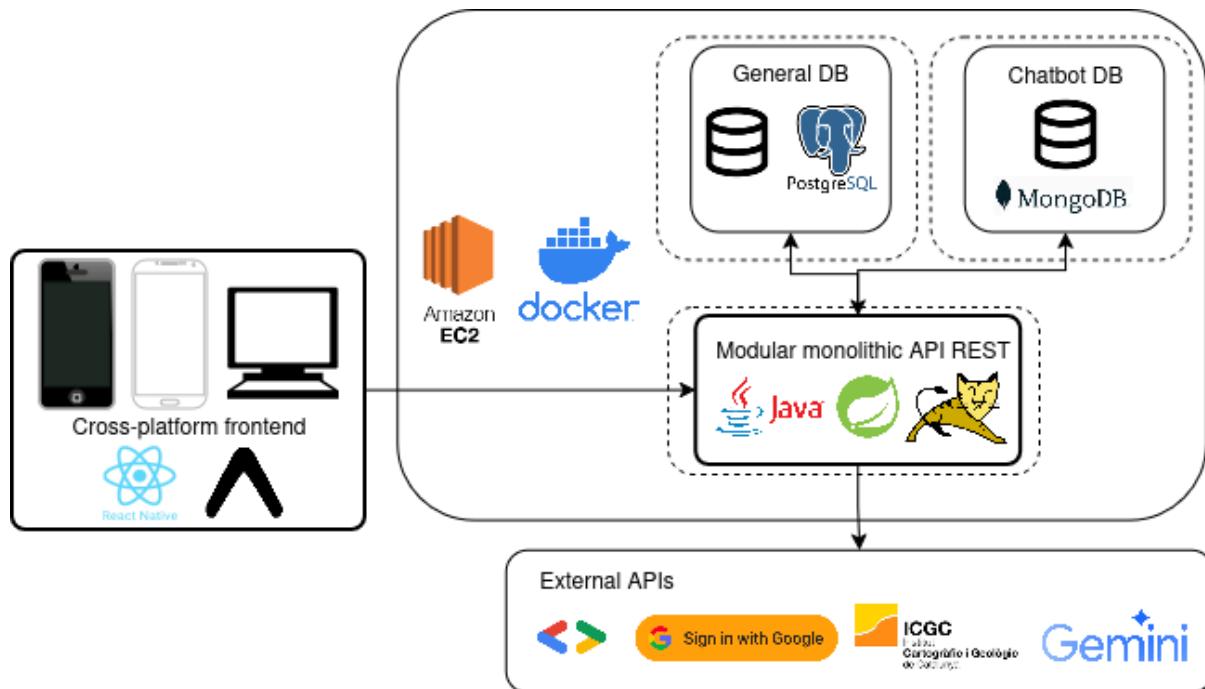
- Hem fet ús d'eines d'ajuda al desenvolupament, com **linters** per garantir la qualitat del codi i assistents d'IA per suggeriments i optimitzacions.
- Per tal de tenir una programació eficient, utilitzem GitHub Copilot amb l'IntelliJ que ens ajuda a fer de manera ràpida els tests unitaris, corregir-nos si tenim algun error de codi i resoldre dubtes.

3. Technical description

3.1. Overall conception of the architecture

3.1.1. Physical architecture

L'arquitectura de l'aplicació es veu representada en el següent diagrama.



Aspectes claus i justificació:

El front-end està desenvolupat amb **React Native** i **Expo**. L'ús d'aquesta tecnologia ens permet desenvolupar una aplicació multiplataforma amb un afegit de codi mínim. S'ha escollit aquesta opció per davant d'altres similars com **IONIC** per el coneixement previ dels membres de l'equip en **React**. Hem considerat que utilitzar una tecnologia coneguda seria un benefici important. **Flutter** es va considerar, ja que presentava avantatges en **llibreria de components** i **optimització**, però la diferència no era suficient com per justificar l'aprenentatge de **Dart**.

Pel que fa el back-end, es s'ha desenvolupat una aplicació en **Java Spring Boot**, seguint la arquitectura de software **modular monolithic** conjuntament amb la filosofia de **clean code**. L'aplicació s'executarà dins d'un contenidor **Docker**. L'arquitectura modular monolithic ens permetrà desenvolupar codi d'una manera més divisible, clara i neta. Per més detall veure [Design patterns applied](#).

La principal justificació per escollir Java Spring Boot és: la seva gran compatibilitat amb **React Native**; la llibreria **ORM** de **Spring Boot**, **Spring Boot JPA**, que permet un disseny de base de dades àgnostic del **SGBD**; els mòduls de testing de **Java**, com **Mockito**, que proporcionen un entorn de testing molt robust. Es van contemplar altres opcions com **Django**, però donat el coneixement previ dels membres en **Java** es van

sacrificar els possibles avantatges (middleware més potent, codi més net i gran integració) en favor de **una tecnologia coneguda**.

L'SGBD escollit per la base de dades general és **Postgres**, principalment per la seva gran compatibilitat amb **SpringBoot JPA** i per la seva utilització en projectes previs. El fet de que sigui codi obert també ha sigut un punt a favor. Pel que fa el Chatbot, hem escollit una base de dades no relacional per la seva velocitat de lectura/escriptura i la seva eficiència per a emmagatzemar dades de tipus document (missatges). L'SGBD per aquesta BD no-relacional és **MongoDB**, principalment per la seva bona integració, la documentació existent i la seva **dashboard** de configuració.

L'aplicació utilitzarà com a **APIs** externes (a nivell d'arquitectura física) les APIs de Google de autenticació, mail i **Gemini**. Hem escollit les APIs de Google per la seva àmplia integració, la bona documentació existent i per tenir els millors *tiers* gratuïts.

Finalment, pel deployment hem escollit **Amazon Web Services**, concretament un EC2 amb la següent configuració:

- Tipus d'instància: **t2.large**
- Memòria: 8GB
- Procesador: 2vCPUS

S'ha escollit AWS per dos principals raons:

- Àmplia documentació, tutorials i facilitat d'integració amb **Docker**
- Crèdit de 300\$ per poder fer un deployment realista.

3.1.2. Architectural pattern(s) applied

Pel que fa **backend**, donades les funcionalitats de la nostra aplicació, diversos patrons arquitectònics es presenten com a candidats. No obstant, d'entre totes les possibilitats, vam considerar que la més òptima era una aplicació **monolítica**.

La principal justificació és el tamany de l'equip i la complexitat afegida que suposen les altres alternatives. Una aplicació basada en **microserveis**, per exemple, seria una alternativa molt més robusta en quant a la fiabilitat i escalabilitat de la nostra aplicació. No obstant, ens hagués suposat un esforç molt gran de configuració, coordinació i instrumentació que no es veia compensat pels beneficis d'aquest patró arquitectònic.

Malgrat haver escollit una arquitectura monolítica, s'ha intentat recollir alguns conceptes de la filosofia de microserveis i és per això que la nostra aplicació no només és monolítica, sino que és **modular** monolítica -en la línia de la filosofia de **clean code architecture** que s'explica més endavant-.

Això vol dir que el nostre codi està dividit en **mòduls** o **features**, de manera que en una fase futura de l'aplicació, un -o més- **mòduls** es podríen aïllar i convertir en microserveis independents de l'aplicació principal amb un cost petit o mínim.

A més, cada mòdul està dividit en **tres capes**:

- Capa d'**entitat**: capa de domini com a tal, on estan les classes que representen elements de l'UML.
- Capa d'**infraestructura**: agrupa totes les classes encarregades de la comunicació entre capes (repositoris i controladors dels *endpoints*).
- Capa de **useCases**: lògica de negoci de l'aplicació.

3.2. Domain layer

3.2.1. Domain model diagram (optional)

3.2.2. Design patterns applied

Clean code architecture

És un enfocament al disseny software que defineix diferents capes que ens permet que el codi sigui fàcil de modificar i mantenible a causa de les capes. De les diferents capes tenim primer els interface adapters, en el nostre cas utilitzem controladors (i aquí tindrem també repositoris per a mètodes) que ens permeten fer la traducció per a les entitats externes de la que usen a les entities i use cases. Després tenim les entities que en el nostre cas són les classes que guardarem a la base de dades, o sigui, que seran els objectes del sistema. En última instància tenim els diferents use cases que implementem. Normalment, els controladors cridaran als use cases que controlaran les entities.

Hem optat per aquesta opció per una sèrie de motius. Un dels més importants era que buscavem treballar en un ambient que poguéssim aprendre i entendre ràpidament per poder posar-nos a fer el codi l'abans possible i sense perdre gaire temps i aquest enfocament ens ho permet. A més a més, podem anar treballant de forma més o menys paral·lela en les nostres branques sense trepitjar-nos gaire. Aquesta estructura també ens permet fer tests de forma senzilla i en seguir-la si necessitem mirar codi que no hem fet és bastant fàcil d'entendre i agradable de llegir.

DTO

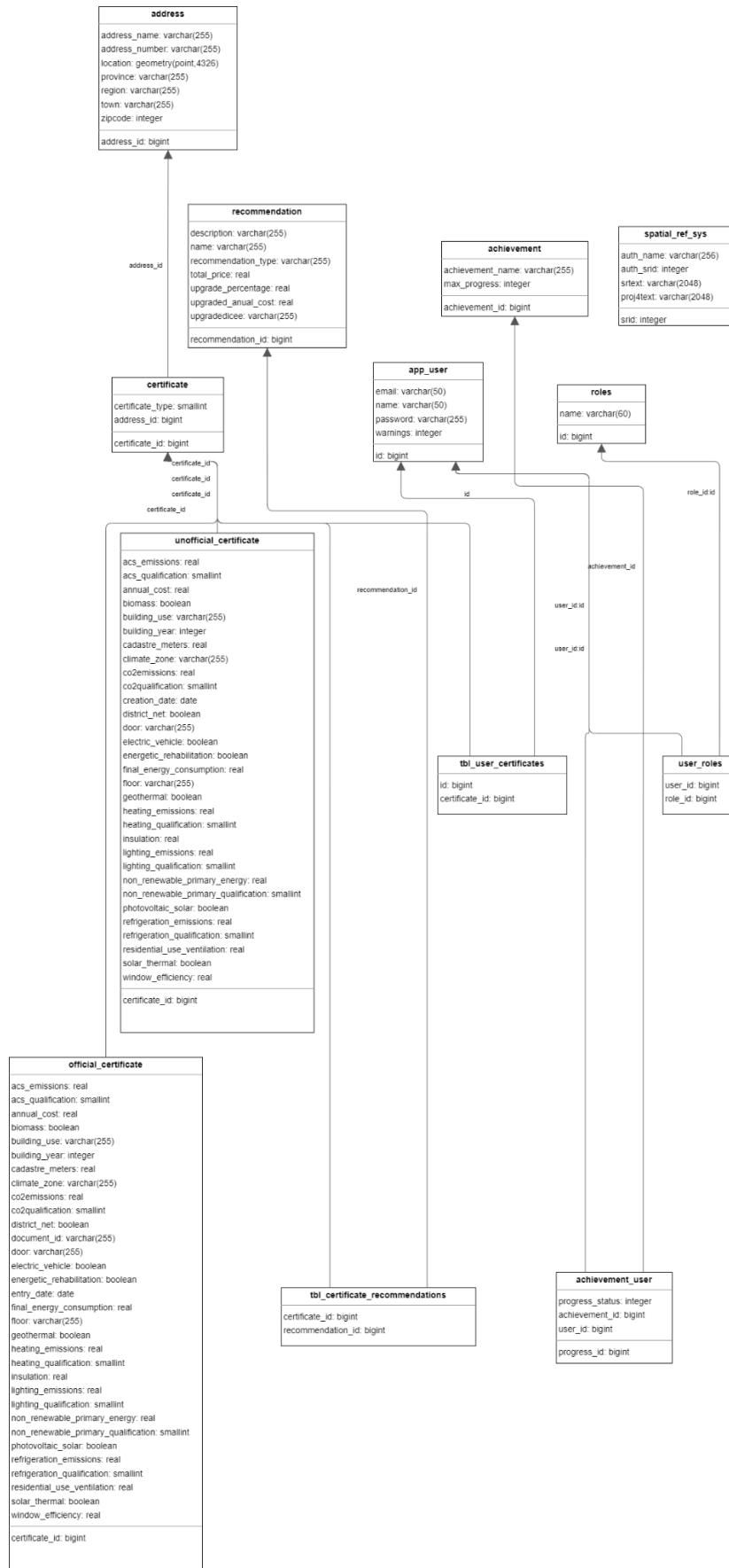
Utilitzem DTO (Data Transfer Object) per enviar la informació necessària al front-end o per crear files a la base de dades. Hem considerat aquest format perquè a la comunicació front-end-back-end fem servir text JSON, ja que és bastant llegible i està estandarditzat, i perquè considerem que és una forma neta d'enviar la informació.

Adaptador

Utilitzem el patró *adaptador* per introduir una capa d'abstracció entre les crides realitzades des del *frontend* a la nostra API i les APIs externes que consumim, com ara les d'Agendados i ICGC.

Aquest patró ens permet desacoblar el nostre sistema de la implementació específica de serveis externs. Així, podem mantenir una interfície estable i coherent cap al *frontend*, fins i tot si les APIs de tercers canvien. A més, facilita les proves, la reutilització del codi i la substitució de dependències externes en el futur.

3.3. Database diagram (UML)



El diagrama actualitzat de la base de dades d'Ecomentor reflecteix la implementació del patró Table per Class per a la jerarquia de certificats, amb una classe pare certificate i dues subclasses (official_certificate i unofficial_certificate), complint així una relació Disjoint i Complete. Els certificats no oficials, que podran ser creats pels usuaris, ja estan incorporats al model. S'han definit taules associatives (tbl_user_certificates, tbl_certificate_recommendations, user_roles, achievement_user) que contenen únicament claus foranes per representar relacions entre entitats. Les recomanacions ja compten amb la taula recommendation.. L'entitat address s'ha integrat amb suport geoespatial mitjançant un camp geometry(point,4326) i s'associa amb els certificats. Els usuaris (app_user) poden tenir rols (rols), certificats i èxits (achievement) vinculats mitjançant taules associatives, incloent seguiment del progrés en els èxits. Aquest model consolida una estructura relacional clara, extensible i preparada per a futures funcionalitats.

Link:

https://drive.google.com/file/d/1vdBzAD7Td6ciK0U72RK_snSLehNj5EwC/view?usp=sharing

3.4. Instrumentation and list of technologies

A continuació un llistat sobre les tecnologies de les que hem parlat en l'arquitectura física. La justificació del seu ús es troba en l'apartat d'arquitectura física.

Llenguatges de programació i frameworks:

- **React Native + Expo (JavaScript):** Utilitzat en la part **front-end** de l'aplicació.
- **SpringBoot (Java):** Extensió de l'*Spring Framework* que facilita molt la gestió i creació d'applications **API REST**, a més de la llibreria Spring Data JPA que permet crear una aplicació amb back-end **ORM**.

Servidors:

- **Amazon EC2 + Docker:** S'ha utilitzat un servidor Amazon EC2 per allotjar el nostre back-end. Per el funcionament de l'aplicació s'ha fet ús de contenidors **Docker**.

Base de dades:

- **PostgreSQL:** Solució amb suport i documentació àmplia de bases de dades relational.
- **MongoDB:** Utilitzem aquesta base de dades no relational per millorar l'eficiència en l'escriptura i lectura dels xats

Altres eines :

- **Postman:** Utilitzem Postman per poder fer crides a la API i poder testejar-la
- **Swagger:** La nostra aplicació és documentada en un swagger que a més també permet executar les crides a l'api
- **Google Cloud Console:** Per poder integrar el SignIn amb un compte de google.

3.5. APIs

3.5.1 API nostre

La nostra api està en el document **API_ECOMENTOR** que es troba en la carpeta del sprint 1 i del 2 és la conversió a pdf del nostre Swagger, on es mostren els diferents endpoints i les seves característiques.

3.5.2 APIs externes que utilitzem

Utilitzem quatre APIs de Google:

- **Google Gmail:** Integrada al nostre backend per utilitzar el servei SMTP i poder notificar els usuaris quan sigui necessari. Hem hagut de configurar els paràmetres a la nostra aplicació:

```
#Mail service
spring.mail.host=smtp.gmail.com
spring.mail.port=587
spring.mail.username=ecomentorCAT@gmail.com
spring.mail.password=${SPRING_MAIL_PASSWORD}
spring.mail.properties.mail.smtp.auth=true
spring.mail.properties.mail.smtp.starttls.enable=true
google.oauth.clientId=${GOOGLE_WEB_CLIENT_ID}
```

I després crear una sèrie de **templates** HTML per a cadascun dels correus.

```
<body>
<div class="email-container">
    <h1>Welcome to Ecomentor!</h1>
    <p>Thank you for signing up. We're excited to have you on board.</p>
    <p>If you have any questions, feel free to reply to this email.</p>
    <div class="footer">
        
        <span>© 2025 ECOMENTOR</span>
    </div>
</div>
</body>
</html>
```

Que quan arriben a l'usuari es veuen de la següent manera:

Welcome to Ecomentor!

Thank you for signing up. We're excited to have you on board.

If you have any questions, feel free to reply to this email.



© 2025 ECOMENTOR

- **Gemini:** Per integrar-nos amb l'API de Gemini, a banda de l'obtenció de l'*API KEY*, hem hagut de fer algunes configuracions:

```
application.properties
# Gemini
gemini.api-key=${GEMINI_API_KEY:fake-key}
gemini.api-endpoint=https://generativelanguage.googleapis.com/v1beta
```

I després la part més important és el nostre **Gemini Service**, encarregat de comunicar-se amb l'endpoint i **seleccionar el model** - en el nostre cas, `gemini-2.0-flash`.

```
GeminiService.java
public GeminiService(RestTemplate restTemplate, @Value("${GEMINI_API_KEY:fake-key}") String apiKey,
                     @Value("https://generativelanguage.googleapis.com/v1beta") String endpoint) {
    this.restTemplate = restTemplate;
    this.apiKey = apiKey;
    this.endpoint = endpoint.endsWith("/")
        ? endpoint.substring(0, endpoint.length() - 1)
        : endpoint;
    this.model = "gemini-2.0-flash";
}
```

- **Google Oauth2:** La integració amb l'API d'autenticació de Google ha requerit configuració a frontend i a backend, totes sobre la mateixa **clientId**.

A frontend, configurem el client i cridem al nostre backend amb el token de Google:

```
Register.tsx
const [ request, response, promptAsync ] = Google.useAuthRequest({
    androidClientId: 'clientId',
});
```

```
● ● ● AuthContext.tsx
const res : AxiosResponse<AuthResponse<any>> = await axios.post<AuthResponse>(`${API_URL}/auth/google`, {
    idToken
});
```

I a backend, recollim el token, el validem i després apliquem la nostra lògica de negoci -creació o no de l'usuari i obtenció del JWT d'ECOMENTOR-.

```
● ● ● GoogleLoginUseCase.java
GoogleIdTokenVerifier verifier = verifierBuilder
    .setAudience(Collections.singletonList(googleConfig.getClientId()))
    .build();
```

- **Google Maps:** La única configuració necessària per integrar-nos amb l'api de Maps ha estat la obtenció d'una apiKey, requerida per l'*app.json* de l'aplicació.

```
● ● ● app.json
{
  "config": {
    "googleMaps": {
      "apiKey": ""
    }
  }
}
```

A més, utilitzem l'api d'[icgc.cat](https://eines.icgc.cat) per poder fer cerques com Barcelona i ens retorna la latitud i longitud.

```
https://eines.icgc.cat/geocodificador/cerca?text={Nom de ciutat}
```

Utilitzem aquesta API ja que les dades dels certificats que disposem són de Catalunya, i d'aquesta manera donem la possibilitat a l'usuari de buscar un municipi ràpidament i centrar-se en ell.

3.5.3 Service Consumption

L'informació sobre el consum i servei de l'API està explicada a l'apartat 1.7. L'implementació de l'API que oferirem a Ventus resulta senzilla si es porta a terme adequadament, ja que actualment disposem de mecanismes per obtenir dades en una certa àrea, només caldrà filtrar i ordenar aquests resultats.

Per part de l'ús de l'API que consumirem, es trucrà depenent de la regió de l'usuari, obtenint així les dades en la seva zona per actualitzar aquells edificis més propers a events.

Sobre les dades obertes, un cop descarregades de la pàgina, vam optar per fer un filtratge exhaustiu per tal de netejar columnes que no ens eren d'utilitat i formatejar-la perquè encaixés en el nostre sistema, tot seguit vam inserir-les amb un runner amb

trucades des de Postman, d'aquesta manera ens era més senzill realitzar proves de càrrega i tenir una millor gestió de les dades. Tot i així, en un futur volem plantejar canviar aquest sistema de manera que puguem fer actualitzacions constants a la nostra base de dades fent crides a l'API de la Generalitat.

3.6. Data treatment

La base de dades sobre la qual es fonamenta la nostra aplicació prové d'un arxiu CSV amb més d'1.300.000 files i 69 columnes, fet que representa una càrrega de dades molt considerable tant a nivell d'emmagatzematge com de rendiment. A més, una part significativa d'aquestes columnes contenen informació irrelevat per als objectius de la nostra aplicació o presentaven un percentatge elevat de valors buits. Per aquest motiu, vam decidir aplicar un procés rigorós de tractament de dades amb l'objectiu d'optimitzar la seva gestió i assegurar-ne la qualitat.

El primer pas va ser l'eliminació de columnes innecessàries mitjançant un **script en Python**, mantenint únicament aquelles que aportaven valor directe a les funcionalitats de l'aplicació (com el consum energètic, les emissions, la superfície útil, etc.). Aquest pas no només va reduir la mida dels fitxers, sinó que també va facilitar la manipulació i càrrega dels mateixos a la base de dades PostgreSQL.

Per tal de facilitar el desenvolupament i les proves de rendiment, vam generar **tres volums de dades** diferenciats: un reduït amb 1.000 files per a tests ràpids, un intermedi amb 10.000, un més ampli amb 100.000, i un complet amb totes les files disponibles. Aquesta estratègia ens ha permès simular escenaris amb diferents càrregues i ajustar el comportament del sistema abans de desplegar-lo amb el conjunt de dades complet.

En una fase posterior, vam intentar filtrar només aquelles files que no contenien cap valor nul per evitar possibles errors en temps d'execució. Tot i així, vam observar que pràcticament totes les entrades tenien algun camp buit, la qual cosa feia inviable aquesta opció sense perdre massa informació útil. En conseqüència, vam optar per gestionar els valors nuls de forma interna en el codi backend, aplicant valors per defecte o estratègies de substitució segons el context del camp.

A més, vam dur a terme una **normalització dels valors** de determinades columnes per garantir la coherència semàntica dins del sistema. Per exemple, en camps com el tipus d'energia principal o la descripció de l'ús de l'edifici, vam unificar els valors possibles per evitar problemes derivats de la inconsistència en la nomenclatura, l'ús de majúscules/minúscules o el format de les dates.

Aquest tractament de dades no només ha millorat l'eficiència i la robustesa del sistema, sinó que també ha assegurat que les funcionalitats com la comparació d'edificis, les visualitzacions o el sistema de recomanacions treballin amb **dades netes, rellevants i**

consistents, fet que incrementa la qualitat de l'experiència de l'usuari final i la fiabilitat de les conclusions extretes.

3.6.1. Recommendation algorithm

L'algorisme de recomanacions energètiques s'ha desenvolupat com una eina automatitzada per analitzar dades reals de certificats oficials d'eficiència energètica i generar suggeriments de millora energètica personalitzats.

El sistema que hem fet és una mica complicat perquè ha d'unir moltes coses diferents: models energètics, dades de consum i informació tècnica d'edificis molt diferents entre ells.

L'algorisme que hem programat mira les dades reals del certificat energètic (com l'energia que es consumeix, les emissions o l'aïllament) i les compara amb uns valors mitjans. Si veu que hi ha diferències importants, decideix automàticament quines millores es poden fer.

Algunes de les recomanacions que pot donar són: posar plaques solars, millorar l'aïllament de les parets, sostres i terres, canviar les finestres per unes de més eficients, o posar calefacció amb biomassa. Aquestes recomanacions es decideixen seguint unes regles que hem programat, tenint en compte les dades reals de cada edifici.

Dades i valors de referència

El sistema pren com a entrada els valors actuals obtinguts del certificat energètic oficial: consums energètics anuals per tipus d'energia (calefacció, refrigeració, aigua calenta sanitària, etc.), emissions de CO₂, qualificació energètica global i altres paràmetres tèrmics de l'envolupant (transmissivitat de murs, coberta i finestres). Aquests indicadors reals es comparen amb uns valors de referència mitjans, que poden ser genèrics o adaptats segons la zona climàtica i l'ús de l'edifici (habitatge, oficina, comerç, etc.). Aquests valors mitjans s'obtenen d'una calculadora energètica interna del projecte, construïda sobre models normatius i dades estadístiques. D'aquesta manera es defineixen llindars de comparació raonables que permeten mesurar objectivament l'estat energètic actual de cada edifici en relació amb un patró de referència.

Generació dinàmica de recomanacions

El procés de generació de recomanacions s'executa automàticament aplicant regles basades en la comparació entre els valors reals del certificat i els valors de referència. Concretament, per a cada mesura energètica s'estableixen condicions d'aplicabilitat que cal verificar. Per exemple:

- **Instal·lació de panells solars fotovoltaics:** Si el consum elèctric de l'edifici és molt superior a la mitjana de la zona i les condicions d'insolació són adequades, el sistema recomana implantar energia solar fotovoltaica per generar electricitat in situ.
- **Millora de l'aïllament tèrmic:** Si el consum de calefacció o refrigeració és significativament més alt que el valor de referència per l'aïllament existent (pareds, coberta, terres), el sistema proposa reforçar la capa aïllant o renovar els materials aillants.
- **Substitució de finestres:** Si les finestres actuals tenen baixa eficiència (vidre simple o antic) i això es reflecteix en un consum elevat de climatització, es recomana reemplaçar-les per finestres de doble o triple vidre amb marcs i perfils d'alta eficiència.
- **Sistemes de biomassa:** Si la calefacció actual es duu a terme amb fonts fòssils (per exemple, gas-oil o gas natural) i l'edifici es troba en una zona amb disponibilitat de biomassa, el sistema suggereix instal·lar una caldera o estufa de biomassa per produir calor renovable.

Aquests criteris s'activen quan la diferència entre els paràmetres reals i els de referència supera un llindar definit, assegurant així que només es proposin millors energètiques amb un benefici significatiu per al cas concret. En definitiva, l'algorithm construeix un conjunt de recomanacions personalitzades combinant la informació del certificat amb la lògica modular, i el resultat és un conjunt de millors a mida per a cada edifici.

Factors de millora i calibració

Per estimar l'impacte energètic i les reduccions d'emissions associades a cada mesura, l'algorithm aplica **factors de correcció** als valors inicials. Aquests factors (per exemple, multiplicadors de 0,85 o 0,90 sobre el consum o les emissions) provenen d'estudis interns i models energètics desenvolupats en el projecte. Concretament, cada mesura té assignat un factor específic que redueix o ajusta el consum inicial segons el tipus d'actuació. Per exemple, s'ha calibrat que reforçar l'aïllament de sostres o parets podria reduir un 10–15% el consum de calefacció (aplicant un factor de 0,85–0,90), mentre que instal·lar panells solars podria abastir una part substancial de la demanda elèctrica (per exemple, aplicant un factor de 0,70 per a l'energia solar generada). Aquests valors s'han obtingut analitzant dades empíriques i estudis tècnics previs, garantint estimacions tant realistes com fàcils d'entendre per a usuaris no tècnics. A més, aquests factors són paràmetres configurables que es poden actualitzar amb noves investigacions o canvis reglamentaris, la qual cosa permet mantenir el sistema ajustat a l'evolució del sector energètic.

Càlcul d'estalvis, cost total i nova qualificació

Per a cada mesura aplicable, el sistema calcula automàticament els **estalvis anuals** d'energia i emissions, el **cost total** d'implementació i la **nova qualificació energètica** estimada. Concretament, la calculadora energètica estima:

- **Estalvis anuals:** Diferència de consum energètic (kWh) i emissions de CO₂ (kg) entre l'estat actual i l'escenari amb la millora aplicada.
- **Cost total:** Preu estimat de la intervenció, calculat multiplicant les quantitats necessàries (per exemple, metres quadrats d'aïllament o unitats de panells) pels preus unitaris definits en la calculadora energètica.
- **Nova qualificació energètica:** Nova etiqueta energètica (per exemple, de classe E a classe D) que s'obté recalculant el consum global de l'edifici amb la millora aplicada.

Aquest procés integral assegura que cada recomanació s'acompanyi d'una avaluació econòmica i energètica completa, facilitant la presa de decisions amb dades objectivables.

3.6.2. Càlcul dels certificats no oficials

Per poder calcular les qualificacions (A,B,C,D,E,F,G) que trobem als certificats energètics oficials per **valors de consum d'energia primària no renovable i emissions de CO₂**, s'utilitzen les següents fórmules

Calificación	Índice
A	$C_1 < 0,15$
B	$0,15 \leq C_1 < 0,50$
C	$0,50 \leq C_1 < 1,00$
D	$1,00 \leq C_1 < 1,75$
E	$1,75 \leq C_1 < 1,00$
F	$1,75 \leq C_1$
	$1,00 \leq C_2 < 1,50$
G	$1,75 \leq C_1$
	$1,50 \leq C_2$

$$C_1 = \frac{(R \cdot I_o / \bar{I}_r) - 1}{2(R - 1)} + 0,6$$

$$C_2 = \frac{(R' \cdot I_o / \bar{I}_s) - 1}{2(R' - 1)} + 0,5$$

Aquesta fórmula s'aplica a immobles amb **ús residencial privat**, per altres edificis s'utilitza una altre fórmula. A la fórmula es té un valor **I_o** que es el valor del indicador analitzat de l'edifici que s'està certificant, com per exemple emissions anuals de CO₂ en calefacció, aquest valor al principi demanavem una aproximació i ara el calculem, més

endavant explicarem això en detall. També tenim les variables de \bar{I}_r i \bar{I}_s que són els valors mitjans del indicador del parc de referència d'edificis d'ús residencial privat (vivenda), el primer és d'edificis nous i el segon d'edificis “existents”, no els nous. També tenim un valor R o R' que es el rati entre el valor de \bar{I}_r o \bar{I}_s depenent de si es R o R' , i el valor del indicador corresponent al percentil del 10% del parc de referència d'edificis existents d'ús residencial privat. Amb això **podem obtenir els indicadors C_1 i C_2** que ens indiquen la lletra de certificació del nostre edifici per emissions CO2 totals, consum d'energia primària, emissions CO2 de refrigeració, etc. C_1 ens és útil si C_2 és més gran que 1 ja que ens ajuda a determinar si és una bona certificació energética.

Per **edificis d'altres usos** s'utilitza la següent taula:

Calificación	Índice
A	$C < 0,40$
B	$0,40 \leq C < 0,65$
C	$0,65 \leq C < 1,00$
D	$1,00 \leq C < 1,30$
E	$1,30 \leq C < 1,60$
F	$1,60 \leq C < 2,00$
G	$2,00 \leq C$

On C , l'índex de qualificació, és el quocient entre valor de l'indicador per l'edifici a certificar i el valor del indicador per l'edifici de referència.

El document on es pot trobar tota aquesta informació és **Calificación de la eficiencia energética de los edificios, enllaç al document:**

<https://www.miteco.gob.es/content/dam/miteco/es/energia/files-1/Eficiencia/CertificacionEnergetica/DocumentosReconocidos/documentos-reconocidos/normativamodelosutilizacion/20151123-Calificacion-eficiencia-energetica-edificios.pdf>

Al mateix document podem trobar molts del valors de \bar{I}_r , \bar{I}_s , R i R' per diversos indicadors. Depenen de la **zona climàtica** i del **tipus d'edifici** residencial privat (vivenda) i en **bloc** i residencial privat (vivenda) i tipus **unfamiliar**. La cosa és que en aquest document **no es troben** els valors de referència **per totes les zones climàtiques** que tenim a catalunya (normalment perquè són valors de refrigeració a zones fredes que són ínfims i poc comuns), per alguns dels valors que utilitzem i el més important, no tenim el valors de referència pels edificis que no són residencials privats, per tant vam haver de buscar una solució a això per tal de ser fidels al càlcul que es fa per les certificacions oficials.

La manera de solucionar això va ser partir dels certificats oficials catalans que tenim disponibles, que utilitzem a la nostre aplicació i que estan carregats a la base de dades. En aquest cas vam agafar els **csv** (Els vam dividir en tres segons el tipus d'edifici perquè 1 sencer era bastant gran i així es pot aprofitar per si només es vol calcular per un tipus d'edifici algú valor). Vam fer **scripts de Python** on s'analitzaven els csv i s'extreien **mitjanes**, que és essencialment el que són els valors oficials que ens donen però trets a partir de mitges, o medianes si era una mostra més reduïda però això va ser molt infreqüent. Amb això ja **podem calcular les diverses qualificacions** del certificat igual que al càlcul definit oficialment amb alguns valors que han hagut d'estar aproximats, però necessitem el valor del **Io**.

A l'**sprint 2** vam desenvolupar la **calculadora** perquè l'usuari, segons el que creia, l'usuari donava una **aproximació** de les emissions de CO2 que generava i l'energia primària no renovable que consumia. Clarament això **no** és **ideal** i és molt **desgradable** per l'usuari però al sprint 2 no vam poder completar la calculadora com ens agradaria. A l'**últim sprint** vam fer aquest canvi que va passar a ser a partir del teu **consum en kWh anuals** de calefacció, refrigeració, ACS i il·luminació, **i el tipus de sistemes** que tenia: caldera de gas, plaques solars, sistema de geotermia, etc. A partir dels sistemes, hi han predefinits a un altre document del govern un factor que serveix per aconseguir el **Io**, ja sigui emissions de CO2 o energia primària no renovable. El **Io** és el **producte dels kWh anuals i el factor corresponent**, per els indicadors totals simplement es realitza una suma dels indicadors calefacció, refrigeració, ACS i il·luminació que és efectivament, el consum total d'electricitat. Els factors estan definits en aquest document: **Factores de emisión de co2 y coeficientes de paso a energía primaria de diferentes fuentes de energía final consumidas en el sector de edificios en España**:

https://www.miteco.gob.es/content/dam/miteco/es/energia/files-1/Eficiencia/RITE/documentosreconocidosrite/Otros%20documentos/Factores_emision_CO2.pdf

L'**antic càlcul** per aproximació encara **s'utilitza per l'aïllament** de la façana i les finestres, **i la ventilació** ja que són valors que s'han de mesurar però l'usuari si pot tenir una idea de si té bon aïllament o no per exemple, hi ha 7 graus com qualificacions hi ha. **Per calcular la millora** que té aplicar les **recomanacions** també s'utilitza aquesta aproximació enllloc de consums en kWh anuals perquè el certificat només guarda els valors de Io i no els de consum (però sí el consum total que no ens serveix perquè no està desglossat). Aquests valors aproximats en 7 rangs també han estat trets a partir de mitges (o medianes) de tots, o casi tots, els certificats oficials que teníem disponibles.

3.7. Development tools and working environment

Per al back-end, hem utilitzat **Spring Boot** com a framework principal perquè proporciona una base robusta i escalable per a la nostra aplicació, juntament amb **JPA** per a la persistència de dades, la qual facilita la interacció amb la base de dades de manera eficient i estructurada. Utilitzem **Maven** per a la gestió de dependències i **Docker** per a la contenedorització, amb la intenció que el nostre projecte pugui funcionar en tots els entorns de desenvolupament, cosa que garanteix un entorn consistent en les diferents etapes de desenvolupament i desplegament. A més, utilitzem **Mockito** per a les proves unitàries, assegurant la fiabilitat del codi.

També, el nostre repositori de back-end compta amb un sistema d'**Integració Contínua (CI)** que imposa l'execució obligatòria de totes les proves automatitzades abans de poder realitzar un pull request. La nostra intenció és que, en el següent sprint, s'afegeixi al CI l'obligatorietat de passar el linter, encara que ja estem fent que abans de fer els pull requests amb dev, cada programador s'asseguri que el seu codi passa el linter que hem establert, el qual és una versió modificada de google checks en la qual hem augmentat la mida mínima de la línia i hem suprimit l'obligatorietat de documentar amb javadoc.

Seguint amb el repositori back-end, s'implementa (a partir de l'sprint 2) un sistema de **Continuous Deployment**, on s'utilitza GitHub Actions per automatitzar el procés de creació de les imatges de Docker de l'aplicació (l'aplicació Java, el SGBD de Postgres i el de Mongo) i la seva pujada i posada en marxa a l'EC2 d'AWS. Gràcies a la *pipeline* del CI ens assegurem que la imatge és **vàlida**, i per tant el **CI+CD** ens assegura tenir **en tot moment** una versió **actualitzada i correcta**.

El deployment, per tant, té lloc a **AWS**, plataforma que ens proporciona moltes eines que permeten fer el procés de desplegament de l'aplicació més fàcil. L'ús que fem de la plataforma de gestió d'EC2 es limita a iniciar i aturar la nostra instància i poder monitoritzar el seu estat. Qualsevol altra gestió la farem accedint al propi EC2 mitjançant una terminal i **SSH**.

Pel que fa el front-end, utilitzem **React Native** i **Expo** com a eines de desenvolupament. Expo ens ofereix la possibilitat de utilitzar molts components pre-carregats al projecte, a més de facilitat en aspectes de l'aplicació com **routing**. Fem ús de **i18n** per implementar la internacionalització de l'aplicació.

Al repositori front-end també hi ha integrada una *pipeline* de **CI** semblant a la de back-end. En aquest cas fem ús del linter **eslint**, el qual a més de comprobar aspectes estilístics també comproba errors en el codi.

Pel que fa al deployment, finalment no s'ha implementat una *pipeline* de CD sòlida. Això es deu als diversos problemes que s'han tingut durant el 3r Sprint amb la plataforma de **EAS** (expo application service), la qual permet generar una APK a partir d'un projecte fet amb expo. El principal impediment ha estat el **temps d'espera obligatori** per a realitzar cada *build*, ja que un cop passats els 3 primers de prova, a hores laborals el temps no baixa de 200 minuts. Veient aquest coll d'ampolla, es va haver de buscar una alternativa: Generar l'APK de forma local. Per assolir-ho, cal:

Executar `eas build --platform android --local`, que ens genera un .aab, amb **bundletool** obtenim un .apks a partir del .aab. Aquest .apks cal canviar-lo de nom- a .zip i extreure'l, obtenint així l'arxiu de l'apk, que sortirà amb el nom **universal.apk**.

Un altre motiu que ens va decantar per no crear el CD va ser el **número limitat de builds** que ofereix expo per cada projecte, ja que només n'ofereix 30, que no són suficients si volem provar de manera adequada els problemes en producció o resoldre'ls.

Finalment, hem optat per carregar a l'EC2 una base de dades amb 1.000 entrades de certificats per mantenir els costos sota control i garantir un rendiment òptim durant la demo. Aquesta quantitat de registres és més que suficient per demostrar totes les funcionalitats principals de l'aplicació i permet treballar amb una mostra representativa sense haver d'assumir despeses massa elevades. Si haguéssim carregat la totalitat dels certificats disponibles, els recursos necessaris haurien disparat els costos i hauríem acostat el sistema a límits de rendiment frustrants per a una presentació per a inversors o clients. Amb 1.000 files, podem mostrar el comportament real de les cerques, filtres i recomanacions sense comprometre ni la qualitat de la demo ni el pressupost disponible.