
ADJUDI-CAT

PES

Grau en Enginyeria Informàtica

Memòria



2023-2024 Q2

Facultat d'Informàtica de Barcelona
Universitat Politècnica de Catalunya

TAULA DE CONTINGUTS

| | |
|---|-----------|
| 1. Requirements | 4 |
| 1.1. General conception of the project | 4 |
| 1.2. "NOT" list | 4 |
| 1.3. Conceptual model | 6 |
| 1.4. Summary of product backlog in the document | 7 |
| 1.4.1. Gestió d'usuaris | 8 |
| 1.4.2. Gestió d'ofertes | 11 |
| 1.4.3. Gestió de subhastes | 14 |
| 1.4.4. Gestió de valoracions | 16 |
| 1.4.5. Gestió de serveis | 17 |
| 1.4.6. Gestió de menús | 20 |
| 1.4.7. Gestió d'administració | 23 |
| 1.5. Non-functional requirements | 25 |
| 1.5.1. Requisits de capacitat d'ús i humanitat | 25 |
| 1.5.2. Requisits de rendiment | 27 |
| 1.5.3. Requisits de seguretat | 27 |
| 1.5.4. Requisits de preservació i suport | 29 |
| 1.6. Treatment of transversal aspects | 30 |
| 1.7. Third-party services | 33 |
| 2. Methodology | 34 |
| 2.1. Project management | 34 |
| 2.1.1. Sprint planning | 35 |
| 2.1.2. Retrospective | 35 |
| 2.1.3. Dailys | 36 |
| 2.1.4. Sprint review | 36 |
| 2.1.5. Refinement | 36 |
| 2.2. Repository management | 37 |
| 2.2.1. Gestió de branques | 37 |
| 2.2.2. Flux de treball | 37 |
| 2.3. Communication within the team | 39 |
| 2.4. Quality management | 40 |
| 2.5. Testing strategy | 40 |
| 2.6. Management of configurations | 41 |
| 2.7. Interaction with colleagues | 42 |
| 2.8. Bug management | 42 |
| 2.9. NFRs treatment | 43 |
| 2.10. IDEs | 44 |
| 2.10.1. Plugins utilitzats i coding assistants | 44 |
| 2.11 Deploy de l'aplicació | 45 |
| 3. Technical description | 47 |

| | |
|--|----|
| 3.1. Overall conception of the architecture | 47 |
| 3.1.1. Physical architecture | 47 |
| 3.1.2. Architectural pattern(s) applied | 51 |
| 3.2. Domain layer | 51 |
| 3.2.1. Domain model diagram (optional) | 51 |
| 3.2.2. Design patterns applied | 52 |
| 3.2.3. S.O.L.I.D. | 60 |
| 3.3. Database diagram (UML) | 61 |
| 3.4. Instrumentation and list of technologies | 61 |
| 3.5. APIs | 62 |
| 3.6. Development tools and working environment | 63 |

1. Requirements

1.1. General conception of the project

El nostre projecte consisteix en un portal d'ofertes de licitacions i concursos públics amb l'objectiu de connectar les empreses o entitats públiques que ofereixen la plaça i aquelles que desitgen presentar-se a aquesta. La funcionalitat principal serà un cercador d'ofertes, on es mostraran totes les disponibles fins al moment, amb la possibilitat de filtrar per proximitat a la ubicació de l'usuari, la data d'inici o final, o pressupost inicial proposat per a la plaça. Cada oferta inclourà els detalls d'aquesta, així com una secció dedicada a valoracions de l'empresa que ha publicat l'oferta. Adjudi-Cat permetrà seguir de prop tot el procés, rebent notificacions quan se superi l'oferta suggerida, quan s'apropi el final del termini o quan aquest acabi, proporcionant connectivitat amb les webs oficials de la Generalitat durant tot el procés.

1.2. “NOT” list

In scope

1. Integració amb *Google Calendar* per a mantenir un seguiment de les dates importants relacionades amb les licitacions.
2. Mapa interactiu que mostra la ubicació de les empreses que han estat adjudicatàries de licitacions, possibilitat d'afegir filtres de cerca.
3. Sistema de gestió d'usuaris amb tres tipus de comptes diferenciats: òrgans públics, empreses privades i administradors.
4. Canal de comunicació directa entre empreses per a facilitar la resolució de dubtes i la negociació.
5. Permet que les empreses litigants puguin avaluar els organismes públics als quals s'han presentat per a un càrrec o han tingut tracte amb ells. Proporcionen detalls sobre el tracte rebut i la facilitat de la interacció, entre altres aspectes, per garantir la transparència i honestetat dels organismes que sol·liciten les posicions.
6. Cada oferta publicada inclou informació detallada com les dates clau i les tasques a realitzar, així com el temps restant fins a la finalització de l'oferta i un històric de les ofertes presentades.
7. La plataforma estarà disponible en català, castellà i anglès.
8. Notificacions sobre esdeveniments importants del calendari, com el final d'una licitació o si algú ha sobrepassat la teva oferta.
9. Mecanisme de subhastes per a la presentació i gestió d'ofertes de manera dinàmica i competitiva. Cada anunci tindrà un camp per augmentar l'oferta.
10. Marcar ofertes d'interès com a favorites i visualitzar els detalls d'aquestes directament des de la seva *home page*.

11. Implementar un xatbot que funcioni com a wiki o buscador d'articles d'ajuda, proporcionant respostes ràpides a les preguntes freqüents.
12. Mecanismes de seguretat addicionals, com el bloqueig de comptes.

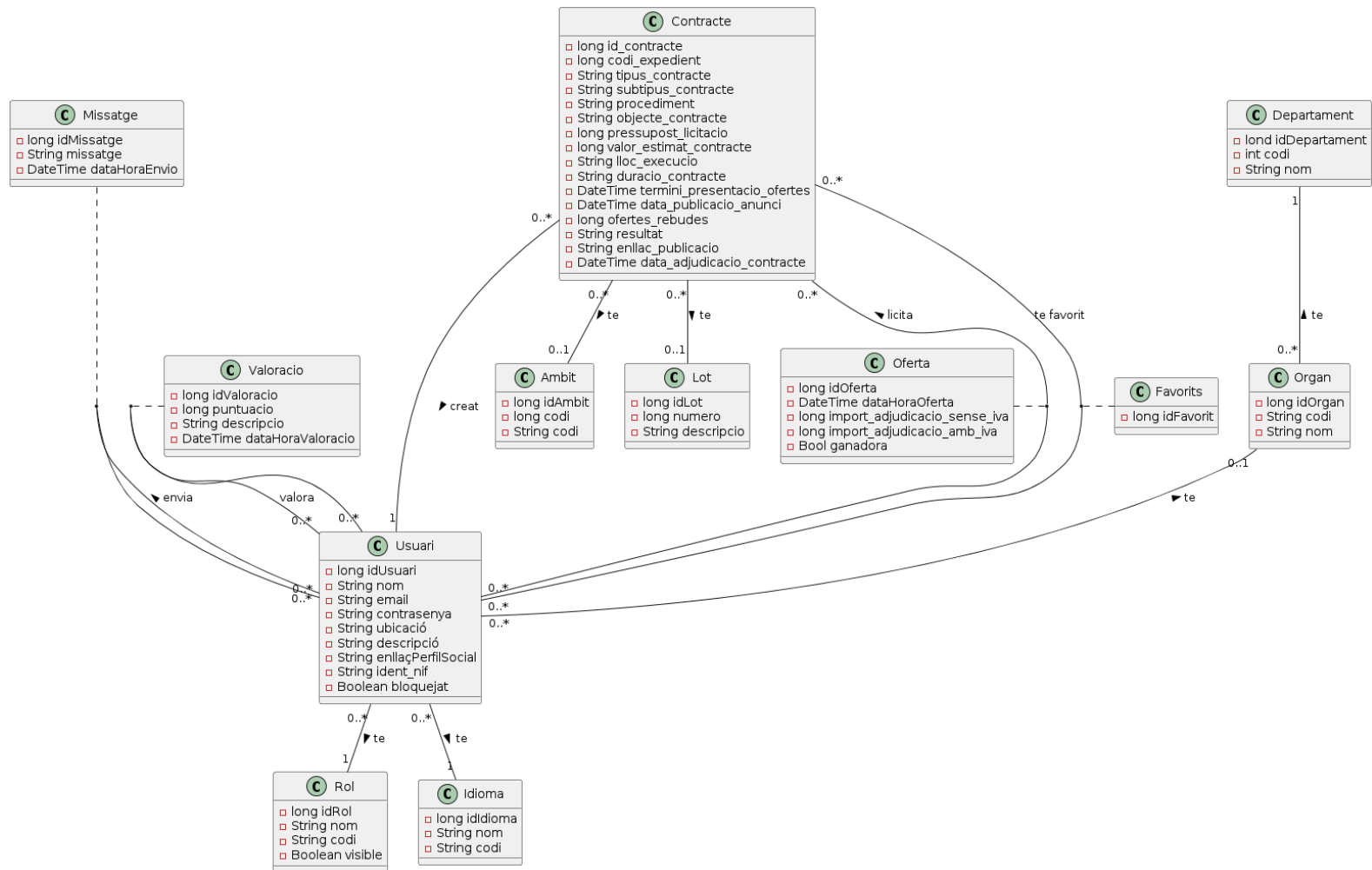
Maybe

1. Oferir l'opció d'utilitzar les credencials de *Google* per a iniciar sessió.
2. Funcionalitats perquè les empreses puguin afegir enllaços als seus perfils de xarxes socials dins de la plataforma.
3. Permetre als usuaris exportar els detalls de les licitacions en formats com CSV, Excel o JSON, simplificant així l'anàlisi de dades i la integració amb altres sistemes.
4. Ofereix l'opció als usuaris de generar un PDF de l'anunci i que el comparteixin per diferents vies a altres persones.

Out of scope

1. La transferència i gestió de documentació oficial entre empreses es deriva a la pàgina web de la conselleria corresponent.
2. Servei de predicció sobre futures licitacions basant-nos en tendències de mercat o dades històriques.
3. Processar pagaments directament entre les parts. En cas necessari, redirigir als usuaris a la web de la Generalitat o altres entitats oficials per a completar aquestes transaccions.
4. Incorporar una funcionalitat similar a un "Muro de Twitter" per a publicacions suggerides dins de la nostra plataforma.
5. Oferir opcions de crear llistes d'amics ni seguir a altres empreses dins de la plataforma.
6. Dur a terme anàlisis addicionals sobre els hàbits d'ús de la plataforma. L'únic seguiment que realitzarem serà si una empresa s'ha presentat a una licitació.
7. Oferir integració automàtica amb sistemes de planificació de recursos empresarials (ERP) o altres sistemes de gestió interns de les empreses.
8. Recomanació de futurs llocs.

1.3. Conceptual model



Restricció textual:

- **R.T.1:** Claus externes:

Usuari: idUsuari,

Departament: idDepartament,

Contracte: idContracte,

Rol: idRol,

Idioma: idIdioma,

Ambit: idAmbit,

Lot: idLot,

UsuariLoginFallido: idUsuariLoginFallido,

Organ: idOrgan

- **R.T.2:** id_emissor i id_receptor no poden ser el mateix usuari d'un missatge.
- **R.T.3:** No poden existir dues ofertes amb el mateix contracte i usuari.
- **R.T.4:** No poden existir dues valoracions amb el mateix usuari de tipus empresa privada i el mateix usuari de tipus òrgan públic.
- **R.T.5:** Un contracte és creat per un usuari de tipus òrgan públic.
- **R.T.6:** No poden existir dos missatges amb el mateix usuari receptor i usuari emissor per un contracte només 1 oferta guanyadora.
- **R.T.7:** Només pot existir una oferta guanyadora per cada contracte.

1.4. Summary of product backlog in the document

En aquest apartat es mostra les taules de cada èpica amb les seves històries d'usuari corresponents, cada un té assignada una puntuació i la prioritat que hem considerat oportuna. El sistema de puntuació està directament relacionat amb el planning poker que s'ha decidit íntegrament amb tot l'equip i que es mostra detalladament al final d'aquest punt.

Per realitzar el planning poker, primer vam acordar que cada hora de treball equivaldria a un punt. A continuació, vam utilitzar el lloc web <https://planningpokeronline.com/> per facilitar la puntuació de les històries d'usuari. En aquesta plataforma, cada membre de l'equip, de forma anònima, va seleccionar el nombre de punts que creia que representava el temps necessari per completar cada tasca, les cartes de punts a triar eren l'escala fibonacci. Després de revelar les puntuacions, discutíem aquelles històries on hi havia una gran disparitat de punts, amb l'objectiu d'arribar a un consens i aclarir els criteris d'acceptació en cas que fos necessari.

D'altra banda, per la prioritat de la història d'usuari, vam utilitzar un sistema de dividit entre baixa, mitjana o alta, comparant-ho amb les funcionalitats més importants de l'aplicació.

1.4.1. Gestió d'usuari

Aquesta èpica se centra en tot el relacionat amb la creació, gestió i eliminació de comptes d'usuari. Inclou funcionalitats com el registre d'usuari, l'autenticació, la gestió de perfils, la recuperació de contrasenyes i permisos d'usuari entre altres.

| Èpica | Històries d'usuari | Punts | Prioritat |
|---------------------|-----------------------|-------|-----------|
| Gestió d'usuari 1.1 | Iniciar sessió | 13 | Alta |
| Gestió d'usuari 1.2 | Tancar sessió | 3 | Alta |
| Gestió d'usuari 1.3 | Registrar compte | 21 | Alta |
| Gestió d'usuari 1.4 | Canviar d'idioma | 3 | Mitja |
| Gestió d'usuari 1.5 | Modificar perfil | 3 | Alta |
| Gestió d'usuari 1.6 | Consultar perfil | 5 | Alta |
| Gestió d'usuari 1.7 | Canviar contrasenya | 3 | Alta |
| Gestió d'usuari 1.8 | Recuperar contrasenya | 21 | Baixa |
| Gestió d'usuari 1.9 | Mantenir actualitzats | 21 | Alta |

US-1.1 Iniciar sessió

Com a qualsevol usuari registrat a l'aplicació,

Vull poder iniciar la sessió del meu compte,

Per a poder realitzar les tasques que m'interessen utilitzant algunes funcionalitats internes de l'aplicació.

Criteris d'acceptació:

- L'usuari ha de poder iniciar sessió sempre que obri la app sense estar prèviament loguejat, o després de tancar sessió.
- El sistema ha de mostrar un error en cas de que la contrasenya sigui incorrecta. En cas de que s'introdugués de forma incorrecta varies vegades, el sistema bloquejarà el compte temporalment. Si es donen més errors, el compte quedarà bloquejat a l'espera de la intervenció d'un administrador.
- La pantalla ha de mostrar un botó per recuperar la contrasenya.
- El sistema ha de mantenir la sessió iniciada si l'usuari i la contrasenya són correctes.

US-1.2 Tancar sessió

Com a qualsevol usuari registrat a l'aplicació,

Vull poder sortir de la sessió que havia iniciat per entrar a l'aplicació,

Per a poder tancar la sessió oberta i/o iniciar amb una altra compte des del mateix aparell (una vegada tancada la sessió oberta).

Criteris d'acceptació:

- L'usuari ha de poder tancar sessió en qualsevol moment emprant aquesta opció, sempre que estigui loguejat.
- No ha de quedar rastre de la sessió prèviament iniciada per l'usuari un cop s'hagi tancat.
- Després de tancar sessió el sistema ha de mostrar la pantalla d'inici de sessió.

US-1.3 Registrar compte

Com a un usuari no registrat a l'aplicació,

Vull poder donar-me d'alta a un compte,

Per a iniciar sessió i després utilitzar l'aplicació.

Criteris d'acceptació:

- Ha de poder fer sign up en qualsevol moment.
- L'email emprat en registrar el compte no pot estar en us.
- El sistema ha de guardar l'usuari a la BD amb totes les dades proporcionades.
- Després de registrar-se, el sistema iniciarà sessió amb l'usuari que s'ha registrat.

US-1.4 Canviar d'idioma

Com a qualsevol usuari registrat a l'aplicació,

Vull canviar l'idioma de l'aplicació,

Per a entendre millor els textos de l'aplicació.

Criteris d'acceptació:

- El canvi d'idioma ha de ser immediat i afectar a totes les pàgines amb text de l'aplicació.
- El text en qualsevol dels tres idiomes ha de ser correcte ortogràfica i gramaticalment, així com contenir la mateixa informació en els tres idiomes de l'aplicació.

US-1.5 Modificar perfil

Com a qualsevol usuari registrat a l'aplicació,

Vull modificar les dades que havia introduït de perfil en el procés de registre a l'aplicació,

Per a tenir el perfil correcte i actualitzat.

Criteris d'acceptació:

- L'usuari ha de poder modificar el perfil en qualsevol moment.
- El sistema ha de guardar els canvis a la base de dades quan l'usuari els confirmi.

- No es té que poder tornar a l'estat anterior del perfil.

US-1.6 Consultar perfil

Com a qualsevol usuari registrat a l'aplicació,

Vull consultar les dades del meu perfil,

Per a revisar la correcció de les dades.

Criteris d'acceptació:

- L'usuari ha de poder consultar el perfil en qualsevol moment.
- La informació mostrada ha de ser correcta i actualitzada, així com ajustar-se a la pantalla.

US-1.7 Canviar contrasenya

Com a qualsevol usuari registrat a l'aplicació,

Vull canviar la meua contrasenya,

Per a actualitzar les meves credencials.

Criteris d'acceptació:

- Ha de poder canviar la contrasenya en qualsevol moment, sempre que estigui loguejat i proporcioni la contrasenya correcta.
- La contrasenya nova ha de ser diferent de la contrasenya antiga.
- El sistema ha d'actualitzar la base de dades amb la nova informació.
- El sistema ha de mostrar un missatge de confirmació al acabar el procés, o un d'error en cas de que no es compleixin els requisits.

US-1.8 Recuperar contrasenya

Com a qualsevol usuari registrat a l'aplicació,

Vull recuperar la meua contrasenya,

Per a tenir accés al meu compte.

Criteris d'acceptació:

- Quan l'usuari introdueix un correu registrat a la aplicació, el sistema detecta si existeix un usuari amb aquest correu.
 - En cas positiu, li envia instruccions per recuperar la contrasenya.
 - En cas negatiu, li informa per pantalla.

US-1.9 Mantenir actualitzat

Com a qualsevol usuari registrat a l'aplicació,

Vull poder llistar tota la informació pertinent,

Per a mantenir-me actualitzat en tot moment de la informació de l'API de la generalitat.

Criteris d'acceptació:

- Tota informació de l'aplicació ha d'estar actualitzada de la API si s'escau.

1.4.2. Gestió d'ofertes

Engloba les funcions necessàries per a crear, publicar, buscar i gestionar licitacions dins del sistema. Això inclou la definició de licitacions i ofertes, les validacions d'aquestes i el seu seguiment fins a la seva finalització o cancel·lació entre altres.

| Èpica | Històries d'usuari | Punts | Prioritat |
|-----------------------|---|-------|-----------|
| Gestió d'ofertes 2.1 | Publicar nova licitació | 21 | Alta |
| Gestió d'ofertes 2.2 | Esborrar una licitació | 3 | Alta |
| Gestió d'ofertes 2.3 | Veure licitacions disponibles | 21 | Alta |
| Gestió d'ofertes 2.4 | Ordenar licitacions disponibles | 13 | Mitja |
| Gestió d'ofertes 2.5 | Filtrar licitacions per una categoria | 21 | Mitja |
| Gestió d'ofertes 2.6 | Veure informació detallada d'una licitació disponible | 8 | Alta |
| Gestió d'ofertes 2.7 | Llicitar en un contracte | 3 | Alta |
| Gestió d'ofertes 2.8 | Buscar licitacions per Keywords | 8 | Baixa |
| Gestió d'ofertes 2.9 | Guardar una licitació a la llista de favorites | 3 | Baixa |
| Gestió d'ofertes 2.10 | Esborrar una licitació de la llista de favorites | 2 | Baixa |
| Gestió d'ofertes 2.11 | Consultar la llista de favorites | 5 | Baixa |

US-2.1 Publicar nova licitació

Com a un organisme públic,

Vull poder publicar una licitació a l'aplicació,

Per a trobar empreses que puguin cobrir la vacant.

Criteris d'acceptació:

- Ha de poder publicar una licitació en qualsevol moment.
- La licitació s'ha de guardar en la base de dades amb la informació correcta.
- El sistema ha de mostrar un missatge de confirmació a l'òrgan públic publicitant.

US-2.2 Esborrar una licitació

Com a un organisme públic,

Vull poder esborrar una licitació a l'aplicació,

Per a que no aparegui.

Criteris d'acceptació:

- L'usuari que va publicar la licitació ha de poder esborrar-la en qualsevol moment.
- S'eliminarà qualsevol rastre del contracte en la base de dades, incloent les ofertes realitzades a aquella licitació.
- Es notificarà a les empreses que havien presentat una oferta amb una notificació.

US-2.3 Veure licitacions disponibles

Com a una empresa licitadora registrada a l'aplicació,

Vull poder veure totes les licitacions disponibles,

Per a poder estar informat sobre noves licitacions que puguin sortir.

Criteris d'acceptació:

- Ha de poder veure totes les licitacions disponibles quan l'usuari entra a la pantalla Inici de l'aplicació.
- L'ubicació de l'organisme públic ha de poder ser indicada al mapa, és a dir, una ubicació existent i correcta.
- La informació ha de ser correcta i actualitzada.

US-2.4 Ordenar licitacions disponibles

Com a una empresa licitadora que busca un litigi al qual presentar-se,

Vull poder filtrar les licitacions disponibles per ubicació, puja mínima, àmbit o dates límit,

Per a poder veure primer aquelles licitacions que siguin del meu interès.

Criteris d'acceptació:

- L'usuari ha de poder ordenar les licitacions sempre que s'estigui en alguna pantalla que els mostri; per defecte, els més recents primer.
- L'ordre ha de ser el correcte i correspondre's amb el desitjat per l'usuari.

US-2.5 Filtrar licitacions per una categoria

Com a una empresa licitadora registrada a l'aplicació,

Vull filtrar les licitacions disponibles per una categoria específica,

Per a facilitar la cerca de les licitacions que m'interessen.

Criteris d'acceptació:

- S'ha de poder filtrar les licitacions en qualsevol moment.
- Després del filtre, es deuen mostrar tan sols les licitacions que compleixin tots els paràmetres establerts mantenint l'ordre.

US-2.6 Veure informació detallada d'una licitació disponible

Com a una empresa licitadora interessada en una licitació disponible,

Vull poder veure el nom de l'organisme públic, un text amb la descripció de l'organisme, l'àmbit que treballa, un mapa indicant la seva ubicació (país, regió, ciutat, carrer i número), un URL que redirecciona la pàgina web pública de la licitació, el cost, el nombre de visites, el nombre d'inscrits, la puja mínima, la data límit de l'oferta, les valoracions de l'organisme públic.

Per a poder accedir a una informació més detallada d'una *licitació* que m'interessa.

Criteris d'acceptació:

- L'usuari ha de poder accedir a aquesta informació sempre que vegi la licitació en una llista.
- La informació ha de ser correcta i actualitzada.
- Ha d'incloure un mapa que mostri la ubicació de la oferta.

US-2.7 Licitat en un contracte

Com a una empresa licitadora interessada en una licitació ,

Vull poder crear una oferta,

Per a poder quedar-me amb la *licitació* i/o crear un xat amb l'organisme públic.

Criteris d'acceptació:

- Ha de poder licitar en qualsevol moment, sempre que la oferta compleixi les condicions (major al mínim definit per la empresa, menor a la licitació més baixa actual)
- Es pot obrir un nou xat entre l'organisme públic i la empresa.
- La oferta s'ha de guardar a la base de dades mentre no es presenti una menor per la mateixa empresa o es tanqui la licitació.
- El sistema ha de mostrar confirmació a l'empresa licitant.

US-2.8 Buscar licitacions per Keywords

Com a una empresa licitadora registrada a l'aplicació,

Vull buscar una licitació usant una o més paraules relacionades amb el meu interès,

Per a facilitar la cerca de totes les licitacions relacionades llistades.

Criteris d'acceptació:

- L'usuari ha de poder buscar licitacions per qualsevol keyword en qualsevol moment.
- El sistema ha de retornar totes les licitacions que continguin la seqüència de paraules en qüestió.

US-2.9 Guardar una licitació a la llista de favorites

Com a una empresa licitadora registrada a l'aplicació,

Vull afegir una licitació d'interès a la llista de favorites,

Per a tenir en una llista totes aquelles que m'interessin.

Criteris d'acceptació:

- L'usuari ha de poder guardar qualsevol licitació.
- La licitació s'ha de mostrar en cas de que es mostri la vista de contractes guardats.
- No es pot guardar una licitació a la llista de favorites quan ja està guardat com a favorita; quan succeeix això, es pregunta a l'usuari si el vol esborrar de la llista.

US-2.10 Esborrar una licitació de la llista de favorites

Com a una empresa licitadora registrada a l'aplicació,

Vull eliminar una licitació d'interès de la llista de favorites,

Per a treure aquella que ja no m'interessa.

Criteris d'acceptació:

- S'ha de poder esborrar una licitació favorita en qualsevol moment (des de la licitació mateixa, o des de la llista de favorites).
- Després d'eliminar-la, no ha d'aparèixer a la llista de favorites llevat que es torni a afegir.

US-2.11 Consultar llista de favorits

Com a una empresa licitadora registrada a l'aplicació,

Vull consultar licitacions que estan a la llista de favorits,

Per a saber quines licitacions tinc a la llista, així com alguns dels seus detalls.

Criteris d'acceptació:

- Ha de poder consultar la llista de favorites en qualsevol moment.
- Es pot arribar a la pàgina de qualsevol dels contractes desde aquesta llista.

1.4.3. Gestió de subhastes

Aquesta èpica tracta sobre el disseny i la implementació de mecanismes per a subhastar licitacions dins de la plataforma. Se centra sobretot en el sistema de notificacions i seguiment d'aquestes.

| Èpica | Històries d'usuari | Punts | Prioritat |
|-------------------------|----------------------------|-------|-----------|
| Gestió de subhastes 3.1 | Termini de l'oferta | 21 | Alta |
| Gestió de subhastes 3.2 | Superació de l'oferta | 5 | Alta |
| Gestió de subhastes 3.3 | Finalització del període | 3 | Alta |
| Gestió d'usuaris 3.4 | Eliminació de la licitació | 3 | Alta |

US-3.1 Termini de l'oferta

Com a una empresa licitadora interessada en una licitació,

Vull rebre una notificació quan s'acosta la finalització del termini d'ofertes,

Per a trobar empreses que puguin cobrir la vacant.

Criteris d'acceptació:

- L'usuari el qual rebrà la notificació ha d'haver realitzat una oferta anteriorment per aquesta licitació.
- La data d'enviament de la notificació ha de ser anterior a la data límit de l'oberta.

US-3.2 Superació de l'oferta

Com a una empresa licitadora interessada en una licitació,

Vull rebre una notificació quan una altra empresa ha superat la meva oferta,

Per a poder fer una contraoferta.

Criteris d'acceptació:

- L'usuari el qual rebrà la notificació ha d'haver fet una oferta anteriorment per aquesta licitació.
- La data d'enviament de la notificació ha de ser anterior a la data límit de l'oberta.

US-3.3 Finalització del període

Com a una empresa licitadora interessada en una licitació,

Vull rebre una notificació quan finalitza el període d'ofertes d'una licitació per la qual n'he fet una,

Per a estar informat de la finalització.

Criteris d'acceptació:

- L'usuari el qual rebrà la notificació ha d'haver creat una oferta anteriorment per aquesta licitació.
- La data d'enviament de la notificació ha de ser posterior a la data límit de l'oberta.

US-3.4 Eliminació de la licitació

Com a una empresa licitadora interessada en una oferta,

Vull rebre una notificació quan una licitació és esborrada,

Per a estar informat de la finalització.

Criteris d'acceptació:

- L'usuari el qual rebrà la notificació ha d'haver realitzat una oferta anteriorment per aquesta licitació.

1.4.4. Gestió de valoracions

Inclou tot el necessari per permetre als usuaris valorar i comentar sobre productes, serveis o altres usuaris. Això implica la gestió de sistemes de puntuació, recopilació d'opinions entre altres.

| Èpica | Històries d'usuari | Punts | Prioritat |
|---------------------------|--|-------|-----------|
| Gestió de valoracions 4.1 | Valorar a l'organisme públic creador d'una licitació | 13 | Mitja |
| Gestió de valoracions 4.2 | Valorar a l'empresa licitadora | 5 | Mitja |
| Gestió de valoracions 4.3 | Consultar valoracions d'una empresa licitadora | 8 | Mitja |
| Gestió de valoracions 4.4 | Consultar valoracions d'un organisme públic (dins de la informació detallada d'una oferta) | 5 | Mitja |

US-4.1 Valorar a l'organisme públic creador d'una licitació

Com a una empresa licitadora que havia realitzat una oferta per a una licitació d'un organisme públic,
Vull poder valorar a l'organisme públic,

Per a poder donar suport en cas que estiguem satisfets, o en cas contrari, exposar els punts negatius que hi va haver o les millores que podrien fer.

Criteris d'acceptació:

- El sistema ha de guardar la ressenya en la base de dades.
- Aquesta ressenya ha de ser accessible pel cas d'ús "Consultar valoracions".
- La ressenya no es guardarà en cas de que s'emprin paraules obscenes o prohibides a l'aplicació.
- El valor numèric ha d'estar entre 1 i 5.

US-4.2 Valorar a l'empresa licitadora.

Com a un organisme públic que publicador de licitacions,

Vull poder valorar a l'empresa licitadora guanyadora de la subhasta,

Per a poder donar suport en cas que estiguem satisfets, o en cas contrari, exposar els punts negatius que hi va haver o les millores que podrien fer.

Criteris d'acceptació:

- El sistema ha de guardar la ressenya en la base de dades.
- Aquesta ressenya ha de ser accessible pel cas d'ús "Consultar valoracions".

- La ressenya no es guardarà en cas de que s'emprin paraules obscenes o prohibides a l'aplicació.
- El valor numèric ha d'estar entre 1 i 5.

US-4.3 Consultar valoracions d'una empresa licitadora

Com a un organisme públic que havia publicat un contracte,

Vull veure les valoracions d'una empresa licitadora que ha pujat a l'oferta,

Per a poder valorar si l'empresa licitadora és fiable o no.

Criteris d'acceptació:

- S'ha de poder consultar valoracions d'una empresa licitadora en qualsevol moment.
- S'ha de poder ordenar aquestes valoracions per nombre d'estrelles o data.

US-4.4 Consultar valoracions d'un organisme públic (dins de la informació detallada d'una oferta)

Com a una empresa licitadora registrada a l'aplicació,

Vull visualitzar les valoracions que té un organisme públic,

Per a poder valorar si l'organisme públic és fiable o no.

Criteris d'acceptació:

- S'ha de poder veure totes les valoracions realitzades en qualsevol moment.
- S'ha de poder ordenar aquestes valoracions per nombre d'estrelles o data.

1.4.5. Gestió de serveis

Se centra en l'oferiment de serveis tant interns com externs a la plataforma, i la seva utilització dins d'aquesta.

| Èpica | Històries d'usuari | Punts | Prioritat |
|-----------------------|--|-------|-----------|
| Gestió de serveis 5.1 | Utilitzar Xatbot | 34 | Mitja |
| Gestió de serveis 5.2 | Obrir mapa de la ubicació d'un organisme | 21 | Mitja |
| Gestió de serveis 5.3 | Obrir mapa de les ofertes properes | 5 | Mitja |
| Gestió de serveis 5.4 | Consultar disponibilitat horaris de locals | 13 | Mitja |
| Gestió de serveis 5.5 | Concertar esdeveniment d'obertura | 5 | Mitja |
| Gestió de serveis 5.6 | Sincronitzar notificacions amb calendari | 8 | Baixa |
| Gestió de serveis 5.7 | Utilitzar cercador de paraules | 8 | Baixa |

| | | | |
|-----------------------|--|----|-------|
| Gestió de serveis 5.8 | Enviar missatges a l'organisme públic d'una oferta pujada | 34 | Mitja |
| Gestió de serveis 5.9 | Enviar missatges a una empresa licitadora que havia pujat a l'oferta | 5 | Mitja |

US-5.1 Utilitzar XatBot

Com a qualsevol usuari registrat a l'aplicació,

Vull introduir un text al *Xatbot*,

Per a tal que m'ofereixi informacions relacionades amb el text introduït.

Criteris d'acceptació:

- L'usuari ha de poder accedir al *xatbot* en qualsevol moment.
- Els articles i respostes del *xatbot* han de ser precises i actualitzades, així com estar disponibles en els 3 idiomes de la aplicació.
- El *xatbot* ha d'oferir tots els articles que retorni la cerca, fins a un màxim de 3.
- En acabar, el *xatbot* preguntarà si s'ha resolt el dubte; en cas de resposta negativa, donarà la opció de tornar a preguntar.

US-5.2 Obrir mapa de la ubicació d'un organisme

Com a una empresa licitadora registrada a l'aplicació,

Vull obrir un mapa on indica la ubicació d'un organisme públic,

Per a poder visualitzar al mapa la ubicació d'un organisme públic.

Criteris d'acceptació:

- L'usuari ha de poder obrir el mapa en qualsevol moment.
- S'ha de poder fer zoom al mapa.
- La ubicació ha de ser precisa i coincidir amb la proporcionada a la licitació.

US-5.3 Obrir mapa dels contractes propers

Com a una empresa licitadora registrada a l'aplicació,

Vull obrir un mapa on indica els contractes properes des de la meua ubicació,

Per a poder visualitzar al mapa les ofertes properes.

Criteris d'acceptació:

- La ubicació s'ha de calcular de forma precisa.
- El rang de distància de les ofertes s'ha de poder modificar.

- El sistema mostrarà al mapa tots els contractes disponibles amb ubicació continguda entre la ubicació de l'usuari i el rang especificat.
- Cada oferta ha de poder contenir un enllaç a la pàgina d'aquesta oferta.

US-5.4 Consultar disponibilitat horaris de locals

Com a empresa litigadora que ha guanyat un concurs,

Vull consultar els horaris en els quals els locals d'esdeveniments disponibles estan sense ocupar,

Per a poder consultar si és viable realitzar un esdeveniment d'obertura en cas que així ho desitgi.

Criteris d'acceptació:

- Les dates mostrades han de ser correctes.
- S'ha de mostrar de forma precisa la franja horaria en la que cada local estarà ocupat.

US-5.5 Concertar esdeveniment d'obertura

Com a empresa litigadora que ha guanyat un concurs,

Vull poder reservar un local disponible en una data i hora determinades,

Per a poder realitzar esdeveniments d'obertura o d'inici del litigi.

Criteris d'acceptació:

- S'ha de poder concertar sempre que s'hagi tancat un litigi, i sempre que el local escollit tingui disponibilitat
- S'ha de mostrar aquest event a l'app de CulturaConnect.
- S'ha de donar la opció de sincronitzar l'event amb el calendari de Google.

US-5.6 Sincronitzar notificacions amb calendari

Com a una empresa licitadora registrada a l'aplicació,,

Vull sincronitzar notificacions amb el calendari de *Google*,

Per a tal que m'avisi altre cop del contingut de la notificació per via de calendari de *Google* en una data concreta, així com poder sincronitzar les dates clau al meu calendari personal.

Criteris d'acceptació:

- Ha de poder sincronitzar notificacions amb calendari en qualsevol moment.
- Les dates sincronitzades al calendari han de ser les correctes.
- Qualsevol canvi en les dates de la licitació ha de ser sincronitzat també al calendari.

US-5.7 Utilitzar cercador de paraules

Com a qualsevol usuari registrat a l'aplicació,

Vull introduir una paraula o grup de paraules al cercador,

Per a poder trobar missatges que continguin aquesta paraula amb més facilitat.

Criteris d'acceptació:

- S'han de mostrar tots els missatges que continguin la paraula.
- S'ha de poder buscar en qualsevol moment que estem en un xat (sigui amb el xatbot o entre empresa i organisme).
- L'usuari ha de poder anar a un missatge en concret després de la cerca.

US-5.8 Enviar missatges a l'organisme públic d'un contracte pujat

Com a una empresa licitadora registrada a l'aplicació,

Vull enviar missatges a l'organisme públic responsable d'una oferta pujada,

Per a poder contactar amb ells amb l'objectiu de profunditzar detalls/informació sobre l'oferta.

Criteris d'acceptació:

- S'ha de poder enviar missatges a qualsevol organisme que hagi publicat un concurs.
- S'ha de guardar un històric dels missatges fins que conclou el termini de l'oferta.
- El missatge s'ha d'enviar i rebre tal com l'ha emés l'empresa licitadora.
- El sistema ha d'informar a l'organisme públic de que ha rebut un missatge.

US-5.9 Enviar missatges a una empresa licitadora que havia pujat al contracte

Com a un organisme registrat a l'aplicació,

Vull enviar missatges a una empresa licitadora que havia pujat a una oferta meua,

Per a poder contactar amb ells amb l'objectiu de profunditzar detalls/informació sobre el contracte.

Criteris d'acceptació:

- Tan sols es podrà comunicar directament si l'empresa licitadora ha pujat a l'oferta.
- S'ha de guardar un històric de missatges fins que acaba el termini d'ofertes.
- El missatge s'ha d'enviar i rebre tal com s'ha emés per l'organisme públic.
- El sistema ha d'informar a l'empresa licitadora de que ha rebut un missatge.

1.4.6. Gestió de menús

Aquesta èpica s'ocupa de la configuració i personalització dels menús dins de l'aplicació entre altres.

| Èpica | Històries d'usuari | Punts | Prioritat |
|---------------------|---|-------|-----------|
| Gestió de menús 6.1 | Obrir el menú flotant | 5 | Alta |
| Gestió de menús 6.2 | Obrir la pantalla de Notificacions de l'aplicació | 3 | Mitja |
| Gestió de menús 6.3 | Obrir la pantalla de Perfil de l'aplicació | 3 | Alta |

| | | | |
|---------------------|---|---|-------|
| Gestió de menús 6.4 | Obrir el xat de l'aplicació | 3 | Mitja |
| Gestió de menús 6.5 | Obrir la pantalla d'Administració de l'aplicació | 3 | Alta |
| Gestió de menús 6.6 | Obrir la pantalla d'Inici de l'aplicació per a empreses licitadores | 3 | Alta |

US-6.1 Obrir el menú fixe

Com a qualsevol usuari registrat a l'aplicació,

Vull tenir un menú fixe que es pugui obrir en cas de voler accedir a altres llocs de l'aplicació, les opcions principals són: Inici, Xat, Notificacions i Perfil,

Per a poder navegar per diferents llocs de l'aplicació i no tapar el contingut de la pantalla quan no es fa servir.

Criteris d'acceptació:

- L'usuari ha de poder consultar aquest menú estigui on estigui de la app.
- Les opcions han de mostrar clarament a on dirigeixen.
- El menú no ha d'ocupar massa espai de la pantalla.
- El menú ha de situar-se en algun lloc on l'usuari pugui arribar fàcilment amb el polze.

US-6.2 Obrir la pantalla de Notificacions de l'aplicació

Com a qualsevol usuari registrat a l'aplicació,

Vull veure totes les notificacions que em van arribant de l'aplicació i una opció de filtrar les notificacions per tipus o data concreta (a nivell d'any, mes o dia), també, poder afegir una notificació al calendari (de Google),

Per a revisar les notificacions que m'havien arribat.

Criteris d'acceptació:

- L'usuari ha de poder obrir el xat desde qualsevol pantalla per via del menú fixe.
- S'han de mostrar totes les notificacions rebudes, ordenades de forma cronològica, així com opcions per filtrar-les i guardar-les com a favorites.
- Cada notificació ha de donar la opció de guardar-la al calendari de Google.

US-6.3 Obrir la pantalla de Perfil de l'aplicació

Com a qualsevol usuari registrat a l'aplicació,

Vull tenir un perfil on pugui veure i modificar les dades emplenades en el moment de la registració a l'aplicació, accedir a la llista de favorites i canviar idioma,

Per a accedir a les coses de preferències "personals".

Criteris d'acceptació:

- Ha de poder obrir la pantalla de perfil en qualsevol moment.
- La pantalla ha de mostrar botons per a totes las opcions disponibles, i obrir una vista per a canviar el que sigui necessari.
- Les dades mostrades han de ser correctes i actualitzats, i encaixar a la pantalla.

US-6.4 Obrir el xat de l'aplicació

Com a qualsevol usuari registrat a l'aplicació,

Vull poder obrir el xat dins de l'aplicació,

Per a comunicar-me directament amb altres usuaris o amb el suport tècnic de manera eficient.

Criteris d'acceptació:

- L'usuari ha de poder obrir el xat desde qualsevol pantalla per via del menú fixe.
- La pantalla ha de mostrar tots els xats actius, així com l'AdjudiBot (*xatbot* d'ajuda).
- S'ha de poder obrir cada xat, mostrant els missatges recents, donant l'opció de redactar i enviar nous missatges.

US-6.5 Obrir la pantalla d'Administració de l'aplicació

Com a usuari registrat a l'aplicació,

Vull poder anar al menú d'administració de l'aplicació,

Per a consultar i modificar les dades disponibles d'administració.

Criteris d'acceptació:

- L'usuari ha de poder accedir a aquesta pantalla sempre que sigui un administrador i esculli aquesta opció.
- La pantalla ha de tenir totes les opcions exclusives per a aquest tipus d'usuaris.

US-6.6 Obrir la pantalla d'Inici de l'aplicació

Com a una empresa licitadora registrada a l'aplicació o organisme públic,

Vull poder anar a l'inici de l'aplicació, on es poden veure totes les ofertes disponibles, un filtre d'ofertes, un xat d'ajuda i un cercador,

Per a consultar ofertes disponibles.

Criteris d'acceptació:

- L'usuari ha de poder accedir a aquesta pantalla en qualsevol moment, mitjançant el menú fixe.

- La pantalla ha de mostrar totes les ofertes disponibles, per defecte en ordre cronològic, així com l'opció de reordenar-les i filtrar-les, i mostrar informació ampliada de cadascuna.

1.4.7. Gestió d'administració

Engloba les tasques relacionades amb la gestió interna de la plataforma, incloent-hi la configuració del sistema, el monitoratge i anàlisi de l'ús de l'aplicació i la gestió dels usuaris entre altres.

| Èpica | Històries d'usuari | Punts | Prioritat |
|----------------------------|-------------------------|-------|-----------|
| Gestió d'administració 7.1 | Desbloquejar un compte | 3 | Mitja |
| Gestió d'administració 7.2 | Bloquejar un compte | 3 | Mitja |
| Gestió d'administració 7.3 | Enviar notificacions | 8 | Mitja |
| Gestió d'administració 7.4 | Administració d'ofertes | 5 | Alta |

US-7.1 Desbloquejar un compte

Com a un administrador registrat a l'aplicació,

Vull poder desbloquejar un compte que ha sigut bloquejat,

Per a que el propietari del compte pugui accedir-hi.

Criteris d'acceptació:

- L'usuari ha de poder accedir al seu compte immediatament després del desbloqueig.
- L'usuari ha de poder accedir al seu compte amb les mateixes credencials que abans, i no ha de perdre cap informació en el procés.

US-7.2 Bloquejar un compte

Com a un administrador registrat a l'aplicació,

Vull poder bloquejar un compte que no està bloquejat,

Per a que el propietari del compte no pugui accedir-hi.

Criteris d'acceptació:

- No s'ha d'eliminar cap informació personal o licitació en curs de l'usuari en el període de temps on el compte estava bloquejat.
- Si l'usuari bloquejat intenta iniciar sessió, una pantalla l'informarà de que el seu compte està bloquejat, i li mostrarà formes d'actuar.
- L'administrador ha d'introduir un nom d'usuari i confirmar per bloquejar un compte. En cas de que el compte que s'intenta bloquejar no compleix els requisits per a rebre un bloqueig

(es un altre administrador, no es detecten problemes) el sistema no permetrà a l'admin bloquejar-la i se li informarà de per què.

US-7.3 Enviar notificacions

Com a un administrador registrat a l'aplicació,

Vull poder enviar una notificació, escrita per mi, a un altre usuari o grup d'usuaris,

Per a avisar d'una informació en concret.

Criteris d'acceptació:

- La notificació ha d'arribar a tots els usuaris afectats per aquell canal que s'hagi escollit prèviament.
- El text de la notificació ha d'arribar als usuaris tal i com l'administrador l'hagi redactat.

US-7.4 Administració de contractes

Com a un administrador registrat a l'aplicació,

Vull poder administrar un contracte,

Per a poder modificar-la o esborrar-la en qualsevol moment.

Criteris d'acceptació:

- Qualsevol rastre de l'oferta ha de desaparèixer, així com totes les licitacions presentades fins el moment de l'esborrat.
- S'ha d'informar a tots els usuaris afectats (organisme públic, empreses licitants)

1.5. Non-functional requirements

A continuació es presenten els diferents requisits no funcionals que considerem essencials per l'aplicació. Hem utilitzat la plantilla Volere per definir-los

1.5.1. Requisits de capacitat d'ús i humanitat

| | |
|------------------------------|---|
| ID | NFR-001 |
| Requisit Volere | 11a. Requisits de Facilitat d'Utilització |
| Descripció | L'aplicació ha de ser fàcil d'utilitzar per a tots els tipus d'usuaris. Això inclou una interfície d'usuari clara i intuïtiva, navegació senzilla entre les diferents funcionalitats i ajuda accessible per als usuaris novells. |
| Justificació | La facilitat d'ús és essencial per aconseguir una adopció àmplia de l'aplicació i per garantir una interacció eficient amb el sistema. |
| Criteris d'acceptació | <ul style="list-style-type: none"> Tot usuari ha de poder utilitzar la nostra aplicació i navegar per qualsevol pantalla sense conèixer com s'ha d'utilitzar l'aplicatiu. Tot usuari que tingui qualsevol dubte ha de poder adreçar-se al xatbot de manera intuïtiva i rebre una resposta precisa a la seva pregunta. |

| | |
|------------------------------|---|
| ID | NFR-002 |
| Requisit Volere | 11b. Requisits de Personalització i Internacionalització |
| Descripció | L'aplicació ha de permetre als usuaris modificar i configurar l'aplicatiu segons les seves preferències personals i convencions culturals. Per no haver d'acceptar ajustos i preferències que els empitjori l'experiència d'usuari. |
| Justificació | La personalització i la internacionalització milloren l'experiència de l'usuari i augmenten la usabilitat del sistema en diferents contextos culturals i lingüístics. |
| Criteris d'acceptació | <ul style="list-style-type: none"> A l'hora de registrar un nou usuari se li ofereix la possibilitat d'escollir l'idioma segons la seva preferència. S'ofereix en la personalització del perfil seleccionar l'idioma de |

| | |
|--|---|
| | <p>preferència entre el català, castellà i anglès.</p> <ul style="list-style-type: none"> • Tot usuari pot modificar el seu perfil en qualsevol moment i editar les seves dades com el nom, descripció, lloc web, dades de contacte, etc. • L'usuari pot en qualsevol moment personalitzar les seves notificacions dins l'aplicatiu, entre visualitzar-les, rebre-les o ignorar-les. • En la pantalla d'inici pot realitzar cerques personalitzades per visualitzar el contingut i contractes que més s'ajustin als seus interessos. |
|--|---|

| | |
|------------------------------|---|
| ID | NFR-003 |
| Requisit Volere | 11c. Requisits d'Aprenentatge |
| Descripció | L'aplicació ha de ser fàcil d'aprendre per a nous usuaris, incloent-hi empreses privades, organismes públics i administradors. Això inclou una interfície d'usuari intuïtiva amb instruccions clares. |
| Justificació | La facilitat d'aprenentatge és essencial per aconseguir una ràpida adopció del sistema i per minimitzar el temps de formació per als nous usuaris. |
| Criteris d'acceptació | <ul style="list-style-type: none"> • Tota empresa privada que utilitzi la nostra aplicació ha de ser capaç de poder navegar per l'aplicació i desenvolupar les seves principals activitats com veure contractes i licitar en aquests sense problemes ni amb formacions especials. • Tot organisme públic ha de poder desenvolupar les seves tasques principals de visualitzar, publicar i eliminar contractes sense formació prèvia. • La funcionalitat de publicar contractes ha de ser de fàcil navegació, des del menú principal i amb botons explicatius. El formulari ha de ser senzill d'omplir i completar. • La funcionalitat de licitar en un contracte ha de tenir una navegació intuïtiva, des de les pantalles d'informació del contracte, i el seu funcionament ha de ser clar, amb botons per licitar i observar pressupostos i altres ofertes per al mateix contracte. |

| | |
|--|--|
| | <ul style="list-style-type: none"> • S'ha de poder accedir al xatbot en qualsevol cas i més si l'usuari necessita ajuda des de gairebé qualsevol pantalla on puguin aparèixer dubtes. |
|--|--|

1.5.2. Requisits de rendiment

| | |
|------------------------------|---|
| ID | NFR-004 |
| Requisit Volere | 12h. Requisits de Longevitat |
| Descripció | L'aplicació ha de ser fiable i duradora, amb un funcionament i desenvolupament fins a una durada marcada. |
| Justificació | La longevitat del sistema és crucial per garantir la continuïtat del servei i la inversió realitzada en el desenvolupament de l'aplicació. |
| Criteris d'acceptació | <ul style="list-style-type: none"> • L'aplicació ha de mantenir-se en funcionament fins a acabar el projecte de l'assignatura de PES. • L'aplicació ha d'estar en constant desenvolupament i millora fins a finalitzar el desplegament de l'aplicació o la finalització de l'assignatura de PES, període previst d'un quadrimestre des del seu començament. |

1.5.3. Requisits de seguretat

| | |
|------------------------|---|
| ID | NFR-005 |
| Requisit Volere | 15a. Requisits d'Accés |
| Descripció | L'accés a l'aplicatiu està restringit segons el tipus d'usuari que intenta accedir. En tots els àmbits, tant pel que fa a dades com de funcionalitats del sistema. |
| Justificació | És essencial garantir la segregació de funcions per protegir la informació delicada i assegurar que cada tipus d'usuari tingui accés només a les funcionalitats que li corresponguin. |

| | |
|------------------------------|--|
| Criteris d'acceptació | <ul style="list-style-type: none"> • En tot moment es registra l'inici de sessió a l'aplicatiu mitjançant tokens d'identificació, amb la seva informació com l'identificador, rol d'usuari i el seu corresponent token. • Les contrasenyes s'encripten en tot moment per no vulnerar la seguretat i a l'hora de comunicar dades entre components i capes. • Tot usuari loggejat amb el seu rol només pot accedir a les funcions proporcionades per a aquell rol. • En tot intent d'entrar a l'aplicació o realitzar qualsevol operació, és essencial o bé iniciar sessió o registrar-se amb unes credencials correctes i que es validen tant en el front-end com en el back-end. |
|------------------------------|--|

| | |
|------------------------------|--|
| ID | NFR-006 |
| Requisit Volere | 15b. Requisits d'Integritat |
| Descripció | L'aplicació implementarà validadors en els camps de dades i registres per garantir la integritat de les dades emmagatzemades. A més, s'aplicaran validadors en la creació i eliminació d'ofertes i contractes per assegurar que siguin coherents i compleixin amb les polítiques establertes. |
| Justificació | La integritat de les dades és essencial per garantir la fiabilitat de l'aplicació i prevenir errors o manipulacions no autoritzades. |
| Criteris d'acceptació | <ul style="list-style-type: none"> • Tot inici de sessió i/o registre de compte compta amb validadors de dades per introduir correctament les dades que es demanen, tant en el front-end com en el back-end, com validadors d'email, dates, etc. • Tota publicació de contracte disposa de validadors i camps obligatoris per omplir el formulari i no crear contractes incomplets o erronis. • Tota nova licitació disposa de validadors de pressupost per no realitzar ofertes menors d'un pressupost mínim ni majors del pressupost ofert. |

| | |
|------------------------------|---|
| ID | NFR-007 |
| Requisit Volere | 15c. Requisits de Privacitat |
| Descripció | L'aplicació garantirà la privacitat de les dades personals dels usuaris, complint amb les regulacions de protecció de dades vigents. Es mantindrà la confidencialitat de la informació i es prendran mesures per protegir-la contra accions no autoritzades. |
| Justificació | La privacitat de les dades és un dret fonamental dels usuaris i una responsabilitat legal de l'organització que gestiona l'aplicació. |
| Criteris d'acceptació | <ul style="list-style-type: none"> • Es revisarà la política de privacitat de l'aplicació i es proporcionarà formació als usuaris sobre com gestionar i protegir la seva informació personal dins del sistema. • No es revelarà informació privada dels usuaris a tercers de cap manera establerts per la llei. • Tota informació delicada s'encrpta mitjançant funcions criptogràfiques de <i>hash</i>. • Utilització de token per a validar les credencials pertinents. |

1.5.4. Requisits de preservació i suport

| | |
|------------------------------|--|
| ID | NFR-008 |
| Requisit Volere | 14a. Requisits de Manteniment |
| Descripció | L'aplicació es mantindrà amb regularitat, amb actualitzacions programades que inclouran la migració de dades i la resolució d'errors coneguts. |
| Justificació | El manteniment regular és essencial per garantir el bon funcionament i la seguretat de l'aplicació a llarg termini. |
| Criteris d'acceptació | <ul style="list-style-type: none"> • Les migracions i actualitzacions de les bases de dades es realitzen cada dilluns a les 5 del matí per minimitzar l'impacte en els usuaris. |

| | |
|------------------------------|--|
| ID | NFR-009 |
| Requisit Volere | 14b. Requisits de Suport |
| Descripció | L'aplicació s'ha de dissenyar per ser compatible amb múltiples plataformes, incloent-hi sistemes operatius i dispositius diversos. |
| Justificació | La diversitat de plataformes utilitzades pels usuaris requereix que l'aplicació sigui flexible i s'adapti a diferents entorns tecnològics. |
| Criteris d'acceptació | <ul style="list-style-type: none"> • L'aplicació està desenvolupada per poder-se desenvolupar sobre qualsevol plataforma, tant Android com iOS i plataformes web. |

| | |
|------------------------------|--|
| ID | NFR-010 |
| Requisit Volere | 14c. Requisits de Adaptabilitat |
| Descripció | L'aplicació ha de ser adaptable per respondre a canvis en els requisits del negoci, les tecnologies emergents i les demandes dels usuaris. |
| Justificació | L'adaptabilitat és clau per mantenir la rellevància i la competitivitat de l'aplicació en un entorn empresarial canviant. |
| Criteris d'acceptació | <ul style="list-style-type: none"> • La metodologia de desenvolupament d'aquest projecte està dissenyada per ser àgil i adaptable, ja que segueix les pràctiques d'Scrum. • L'aplicació està dissenyada amb una arquitectura Model Vista Controlador , fet que facilita la incorporació de noves funcionalitats sense afectar el conjunt de l'aplicació. |

1.6. Treatment of transversal aspects

| Aspect | Description (final target) | Current status |
|--------|----------------------------|----------------|
|--------|----------------------------|----------------|

| | | |
|-----------------|---|--|
| Geolocalització | Implementació de funcionalitats basades en la ubicació geogràfica dels usuaris, com ara la cerca de serveis o l'oferta de contingut localitzat. Incorporació d'un mapa amb les diferents ubicacions dels contractes. | Durant aquest sprint s'ha integrat l'API de Google Maps al complet amb el servei de mapes i geolocalització. S'ha implementat una pantalla amb el mapa i la geolocalització de l'usuari instantània. |
| Xarxes socials | <p>Possibilitat d'oferir l'opció d'utilitzar les credencials de Google per a iniciar sessió.</p> <p>Funcionalitats perquè les empreses puguin afegir enllaços als seus perfils de xarxes socials dins de la plataforma.</p> | <p>S'ha implementat amb èxit la integració dels serveis de google tant per iniciar com per tancar sessió amb Google OAuth, brindant una seguretat més robusta de l'aplicació i una manera més senzilla d'iniciar sessió.</p> <p>Pel que fa a les funcionalitats addicionals sobre perfils, està plantejat decidir-ho i concretar-ho pel següent sprint</p> |
| Xat | <p>Implementació d'un sistema de xat en temps real per permetre la comunicació entre usuaris dins de l'aplicació.</p> <p>Implementació d'un xatbot d'ajuda als usuaris basat en unes preguntes i respostes freqüents. Ambdós amb l'històric de missatges.</p> | Implementació completa tant al back-end com al front-end del xat funcional entre usuaris, empreses privades i organismes públics, i del xat-bot d'ajuda. A més com a servei a oferir està enllestit pel seu funcionament. |
| Gamificació | Implementació de valoracions basada en puntuació d'estrelles (1-5) i en comentaris entre diferents | S'ha implementat a nivell visual les targetes d'informació sobre perfils amb les estrelles i el possible |

| | | |
|--------------------|---|--|
| | <p>usuaris de l'aplicació.</p> <p>Com a mesura de fiabilitat i compromís entre empreses i òrgans públics.</p> | <p>comentari. Però no s'ha implementat la seva funcionalitat al complet, està plantejat implementar-ho al darrer sprint.</p> <p>Arquitectura inicialitzada.</p> |
| Stakeholders reals | <p>Possibilitat d'oferir el sistema a possibles stakeholders com la Generalitat de Catalunya.</p> <p>Creació de perfils per a stakeholders reals, com empreses i organitzacions, per permetre'ls gestionar i participar en processos específics de l'aplicació.</p> | <p>Implementació de les dues variants de perfil respecte a principals usuaris de l'aplicació: Empreses privades i Òrgans públics.</p> |
| Refutació | <p>Implementació d'un sistema per gestionar i rebatre refutacions de contractes o ofertes presentades a l'aplicació.</p> <p>Implementació de bloqueig de comptes i notificacions de tota mena.</p> | <p>S'ha començat a dissenyar i implementar el bloqueig de comptes per reincidència de <i>Log In</i> fallit.</p> <p>Arquitectura inicialitzada.</p> |
| Calendari | <p>Integració d'un calendari amb <i>Google Calendar</i> per programar esdeveniments, reunions o altres activitats relacionades amb els contractes o ofertes.</p> | <p>En aquest sprint no s'ha pogut implementar degut a problemes amb els serveis de Google però estava plantejat desenvolupar-lo. En el següent sprint s'acabara d'implementar o refactoritzar el seu ús.</p> |

| | | |
|---------------|--|--|
| Web-app admin | Desenvolupament d'una interfície d'administració web per permetre la gestió de dades i funcionalitats de l'aplicació per part dels administradors. | No s'ha implementat cap funcionalitat sobre web-app admin. Possible plantejament per a sprints finals. |
| Multiidioma | Suport per a múltiples idiomes per permetre als usuaris utilitzar l'aplicació en català, castellà o anglès. | S'ha implementat a nivell de front-end l'elecció de l'idioma de preferència tant al registrar compte com a l'hora de modificar el perfil. Arquitectura inicialitzada. |

1.7. Third-party services

Per saber a qui i de qui proveirem serveis, hem parlat a classe amb els membres d'altres equips.

En la primera fase de la Inception 1 ens van informar sobre la necessitat de col·laborar amb dos dels altres grups, programant un servei per un d'ells i rebent un servei d'un altre grup. Després de fer una pluja d'idees, vam veure que el grup amb l'API que ens resultaria de major utilitat era CulturaConnect, encarregats de la gestió d'activitats culturals. El servei que els hi vam sol·licitar va ser el de rebre des de la seva aplicació d'horaris i disponibilitat de certs locals en una data determinada, per saber si un local en concret estaria disponible en tancar un litigi per poder realitzar un esdeveniment de clausura d'aquest. CulturaConnect ha acceptat la nostra proposta.

En definitiva, la funció que necessitem és: **Donat un local i una data/hora concreta, saber si està disponible el local per realitzar un esdeveniment, o si està ocupat.**

En una classe presencial posterior, l'equip de Farm2Table va venir a parlar amb nosaltres per demanar-nos una extensió del nostre xatbot d'ajuda. La seva idea era crear un servei semblant al nostre, on els usuaris puguin plantejar dubtes i el sistema detecti el seu problema i el redirigeixi a un article d'ajuda. En poc temps vam acordar ampliar el nostre xatbot d'ajuda perquè pugui utilitzar-se també en la seva aplicació, i ho vam acceptar.

En resum, la funció que crearem serà: **Donada una relació entre paraules clau i articles proporcionada prèviament, crear un sistema que busqui aquestes paraules clau en una consulta i redirigeixi als articles d'ajuda corresponent.**

2. Methodology

2.1. Project management

En el desenvolupament del nostre projecte, hem optat per seguir la metodologia àgil Scrum, utilitzant Taiga com a eina principal per a la gestió i seguiment de les tasques. Aquesta plataforma ens facilita la creació i assignació de tasques als membres del grup, així com la classificació de les històries d'usuari dins del backlog, organitzades per èpiques. Ens permet també establir una puntuació per a cada història basada en la càrrega de treball prevista, dividida entre els components clau del projecte com són el front-end i el back-end.

A Taiga, tenim un tauler personalitzat amb columnes que representen les diferents etapes del cicle de vida d'una història d'usuari, des de 'New' fins a 'Closed', passant per 'In Progress' i 'Ready for Test', cada una amb el seu color corresponent en el diagrama d'estats, també tenim la columna de "Needs info" per casos bloquejants. Això ens ajuda a tenir una visibilitat clara i immediata de l'estat de cada tasca, facilitant una gestió àgil i adaptable del nostre flux de treball. El procés de canvi d'estat de les històries es realitza de manera intuïtiva: les històries s'arrosseguen d'una columna a una altra segons avancen en el seu desenvolupament. Per exemple, una història passa de 'New' a 'In Progress' simplement arrossegant-la a la columna corresponent. Quan una història està preparada per ser testejada, se la mou a 'Ready for Test', i una vegada passats els tests i completades totes les tasques segons la DoD, es mou a 'Closed'.

Per determinar la càrrega de treball de cada història d'usuari, fem servir la tècnica de planning poker, assignant punts que equivalen a hores de treball estimades. Aquest procés ens permet acordar el pes de cada tasca de manera consensuada, assegurant una estimació equitativa i realista, està explicat amb més detall a l'apartat d'històries d'usuari.

En relació amb la Definition of Done (DoD), està definit tant per a les històries com per als sprints. Per a les històries, inclou la finalització de totes les tasques relacionades, el pas satisfactori de tots els tests segons els criteris d'acceptació i un codi lliure d'errors. Pel que fa als sprints, nosaltres considerem completat un sprint quan totes les històries incloses s'han desenvolupat correctament, no hi ha errors ni bugs, els tests d'integració són satisfactoris i la documentació necessària està completa. Abans de començar un nou sprint, revisem tots els detalls pendents o aspectes a millorar com hem comentat a la reunió retrospectiva, així com qualsevol error comès, per garantir un inici àgil i eficient del següent sprint. Aquest enfocament metòdic i estructurat, recolzat per l'ús eficaç de Taiga i els principis de Scrum, ens permet abordar de manera eficient el desenvolupament de la nostra aplicació mòbil, assegurant una comunicació efectiva, una planificació detallada i una execució àgil de cada sprint.

A més, tenim la figura del Scrum Master que va rotant, permetent així que diferents membres de l'equip assumeixin aquest rol i aportin noves perspectives a la gestió del projecte. Aquesta pràctica

fomenta sobretot el compromís amb la metodologia Scrum i amb el projecte, a més de desenvolupar habilitats de lideratge dins del grup.

Per ser fidels a la metodologia que seguim i organitzar millor el projecte, fem diferents reunions durant el sprint que en els següents punts expliquem.

2.1.1. Sprint planning

El planning és una reunió que fem al principi del sprint, és una etapa clau per establir les bases del sprint i definir tot el que tractarem. Primerament, definim els objectius del sprint, posteriorment establim les prioritats a desenvolupar i escollim les èpiques que ens duren a complir els propòsits, cal tenir en compte que els punts que escollim tinguin sentit per abastar el nostre projecte, ja que, si ens passem pot portar males dinàmiques al grup o frustració, i si no arribem no acabarem el desenvolupament. En ser 3 sprints intentem comprendre un 35% per sprint.

Seguidament, dividim el grup en diferents parts, corresponents a les èpiques principals sobre les quals treballarem durant el sprint. Aquest enfocament ens permetrà evitar complicacions durant la integració del codi (merges). Dins d'aquests subgrups, ens organitzarem també segons les especialitats de front-end i back-end, buscant optimitzar la coordinació. Tot i això, mantenim un enfocament de treball transversal, ja que, tot l'equip col·labora i ofereix suport. De cara a les històries s'ha decidit dividir-les en subtasques amb els seus criteris d'acceptació i testing corresponents.

Al final es proposen també objectius no tan relacionats amb aspectes tècnics sinó individuals o d'equip que s'han d'intentar complir també i es parlaran a la retro.

2.1.2. Retrospective

Després de cada sprint, organitzem una sessió de retrospectiva amb l'equip. El nostre objectiu principal és reflexionar junts sobre el que hem aconseguit, identificant el que ha funcionat bé i els aspectes a millorar. Iniciem la sessió establint clarament els objectius i assegurant-nos que tothom se senti còmode per compartir les seves experiències. És important per a nosaltres recollir les opinions de tots, així que dediquem temps a expressar les nostres reflexions, tant positives com negatives.

Durant la sessió, analitzem els comentaris i debatem obertament per aprofundir en els aprenentatges i identificar les accions de millora. Ens fem càrrec de documentar aquestes accions i assignar responsabilitats per assegurar que s'implementin. Al final, sempre recapitem les decisions preses i agraïm a l'equip la seva participació activa.

És fonamental per a nosaltres recollir feedback sobre com ha anat la sessió per millorar-les progressivament. Aquest enfocament ens ajuda a ser més eficients i a mantenir un bon ambient de treball i poder aplicar les solucions que hem recopilat posteriorment.

2.1.3. Dailys

Nosaltres realitzem reunions diàries tres cops per setmana, no cada dia, una pràctica que ens ha ajudat a mantenir l'equilibri entre el treball col·laboratiu i el temps dedicat a l'execució individual. En aquestes reunions, cadascun de nosaltres comparteix en què ha treballat recentment, en què es preveu treballar a continuació, i si hi ha algun obstacle que ens estigui bloquejant. Aquesta estructura ens permet estar al dia sobre el progrés del projecte i identificar ràpidament qualsevol problema que pugui necessitar atenció addicional. Duran no més de 20 minuts i ens assegurem que tothom tingui l'oportunitat de parlar i de ser escoltat.

2.1.4. Sprint review

Durant la review, primer fem un repàs de les metes del sprint i discutim fins a quin punt hem aconseguit complir amb els objectius establerts. Seguidament, fem la demostració, mostrant les funcionalitats desenvolupades, com funcionen en un entorn real i com responen a les necessitats d'usuari que vam identificar a l'inici del sprint. Aquesta part és particularment valuosa, ja que el professor (product owner) pot aportar una perspectiva externa, donant-nos correccions i indicant aspectes que potser no havíem considerat.

2.1.5. Refinement

Realitzem sessions de refinament de manera opcional, convocant-les quan sorgeix la necessitat. Aquestes sessions les fem a través de Discord, ja que permet una interacció en temps real tant per veu com per text, facilitant així la col·laboració a distància.

En aquestes reunions, repassem les històries d'usuari, tasques o èpiques que requereixen una definició més clara o una distribució més equitativa del treball. Discutim i resolem qualsevol dubte que pugui haver-hi, i en casos on és necessari, ajustem els criteris d'acceptació per assegurar que tothom té una comprensió clara dels objectius. Aquest enfocament flexible i a demanda del refinament ens permet adaptar-nos millor als canvis i assegurar que l'equip estigui ben preparat per avançar amb confiança en el desenvolupament del projecte.

2.2. Repository management

Per a la gestió eficient del nostre repositori, hem optat per l'estratègia Gitflow en combinació amb el sistema de control de versions Git, creiem que aquesta part és fonamental per assegurar un alt estàndard de qualitat en el codi que arriba a producció. Git ens ajuda a mantenir un entorn de treball net i gestionar el cicle de vida del nostre codi de manera eficient.

2.2.1. Gestió de branques

La nostra branca principal, master, representa la versió estable i llista per a producció, aquí es troba el codi que és completament testejat i preparat per a ser desplegat a producció, aquesta branca reflecteix el que està corrent actualment en l'entorn de producció.

La branca develop, en canvi serveix com a entorn d'integració contínua, on es fusionen les característiques completes. Aquesta branca recull tot el treball imminent que està en procés, però encara no és prou estable per passar a màster.

2.2.2. Flux de treball

Per cada nova funcionalitat o millora del desenvolupament, creem una branca específica des de develop dins la carpeta features. Aquesta branca està dedicada exclusivament al desenvolupament d'aquesta nova funcionalitat i porta el nom d'aquesta o similar. Només una vegada que aquesta està completada, revisada i testejada, es procedeix a fer merge request de nou a develop.

En el context de les merge request del nostre projecte, tenim una pràctica establerta que ens ajuda a mantenir la qualitat i sobretot l'eficiència del nostre codi: la majoria dels reviews es desenvolupem en parella, un mètode conegut com a "peer programming". D'aquesta manera la persona que ha fet el codi li explica a l'altre les funcionalitats i millorem la qualitat d'aquestes i accelerem molt més el procés per fer-les.

Per això normalment el mergeem directament al nostre repositori principal, ja que durant la sessió de peer programming ja s'ha revisat i discutit en detall. No obstant això, en situacions on sorgeixen dubtes o quan es tracta de funcionalitats particularment crítiques o complexes, optem per organitzar una trobada virtual a través de Discord amb altres membres per discutir-ho.

Quan detectem errors crítics com es comenta a l'apartat de bug management, creem branques de hotfix directament des de develop, ja que, encara que l'error es trobés a màster s'hauria de tornar a passar el procés de testing. Aquestes correccions d'urgència es desenvolupen de manera aïllada per a no interferir amb el flux de treball regular de desenvolupament. Un cop el hotfix està preparat, es fusiona i testreja, assegurant que les correccions es propaguen per tot el sistema.

Abans de llançar una nova versió de l'aplicació, iniciem el procés creant una branca release directament des de la branca develop. Aquesta branca de release es converteix en l'espai on es realitzen els últims ajustos i es duen a terme tests unitaris exhaustius per assegurar que tot funcioni correctament. Aquest període de preparació és crucial per garantir la fiabilitat i estabilitat de l'aplicació abans del seu llançament.

Un cop hem confirmat que la branca release compleix amb tots els criteris de qualitat i funcionalitat, procedim a fusionar-la amb la branca master. Immediatament després de la fusió, realitzem un pas addicional molt important: la creació d'un tag. Aquest tag s'associa amb el número de versió corresponent a la nova release. Per exemple, si la versió de l'aplicació és la 2.5, el tag seria v2.5. Els tags són essencials perquè serveixen com a referències estables per a versions específiques de l'aplicació, facilitant així la gestió de les versions i permetent un accés ràpid a qualsevol versió anterior en cas de necessitat.

Gitflow és una peça fonamental del nostre èxit en el desenvolupament de software, ja que ens aporta una estructura que simplifica la gestió de múltiples línies de desenvolupament, alhora que facilita l'entrega contínua de codi d'alta qualitat als nostres usuaris.



Figura 1. Diagrama de GitFlow

Merge request:

Quan volem integrar noves funcionalitats o correccions a la branca principal, iniciem un merge request (MR) des de la interfície de GitHub o la integració que té el IDE. Primer, assegurem que la branca de la funcionalitat estigui actualitzada amb els últims canvis de la branca develop o master, depenent del cas. Aleshores, creem el MR i assignem almenys un revisor del nostre equip, qui serà responsable de revisar el codi, assegurant-se que compleix amb els nostres estàndards de qualitat i

que passa tots els tests automatitzats. Un cop el revisor aprova el MR, i no hi ha conflictes, procedim a fer el merge de la branca al destí seleccionat. Moltes vegades es fan els merge directament com s'ha explicat anteriorment.

2.3. Communication within the team

En el nostre enfocament integral de comunicació en l'equip de desenvolupament, hem establert canals específics per a abordar diverses necessitats. La nostra dinàmica de comunicació es basa en l'ús combinat de WhatsApp i Discord, adaptant-nos a les necessitats específiques de cada moment i facilitant un entorn col·laboratiu efectiu.

Utilitzem WhatsApp per a coses diàries i informals, que ens permet resoldre dubtes ràpids i coordinar detalls, com comentaris puntuals o avisos d'última hora. Aquesta eina és un canal ràpid i eficient per a comunicacions que requereixen una resposta immediata o per a compartir informació de forma àgil i despreocupada.

D'altra banda, Discord es converteix en el nostre espai central per a la col·laboració més formal i estructurada. Hem creat diversos canals de text i veu, organitzats per temes específics com back-end, front-end, planificació de tasques, issues, i altres coses, facilitant així la distribució dels espais de discussions i mantenint l'ordre en la comunicació. A més, la capacitat de Discord per a suportar reunions virtuals, compartir pantalla, i els bots automatitzats per a notificacions de GitHub i Taiga, fa que sigui totalment clau per poder fer peer programming i resoldre problemes.

Les nostres reunions d'equip es realitzen de 3 cops per setmana, dues a classe i una a través de Discord al cap de setmana, amb l'objectiu de tenir temps entremig per continuar avançant. Aquestes reunions es voten amb una aplicació que s'anomena StrawPoll, en cas que s'hagin de fer reunions focalitzades no s'utilitza. Les sessions són de tipus daily stand-up de 20 minuts aproximadament. En aquestes trobades, cada membre exposa els avenços duts a terme, els plans a curt termini, i qualsevol dificultat trobada, amb l'objectiu d'identificar obstacles i buscar solucions col·lectives. Aquestes sessions ens permeten mantenir una visió clara del progrés del projecte i ajustar el nostre pla d'acció de manera dinàmica.

Aquesta estructura de comunicació versàtil ens permet mantenir-nos connectats de manera efectiva, abordar ràpidament els desafiaments i garantir un flux constant d'informació per al progrés eficient del nostre projecte.

2.4. Quality management

En el nostre compromís amb l'excel·lència i el lliurament d'un producte de qualitat, hem incorporat pràctiques efectives de gestió de qualitat en el nostre procés de desenvolupament. Una de les eines fonamentals que utilitzem són els linters, els quals exerceixen un paper crucial en la identificació i correcció primerenca d'errors en el nostre codi font. Aquests linters, en aplicar regles predefinides i estàndards de codificació, ens permeten mantenir una base de codi consistent i lliure de possibles problemes abans d'avançar en el cicle de desenvolupament. Aquesta implementació proactiva no sols millora la qualitat del codi, sinó que també contribueix a l'eficiència de l'equip en reduir el temps dedicat a la correcció d'errors en etapes posteriors del desenvolupament.

Els assistents de codi i la intel·ligència artificial, que proporcionen suggeriments de codi en temps real, també formen part del nostre desenvolupament, augmentant la productivitat i millorant la qualitat del codi amb propostes eficients. Per altra banda, les revisions de codi estructurades i el desenvolupament basat en proves, que reforcen la nostra gestió de qualitat. Aquestes revisions asseguren que el codi compleixi els requisits que hem establert amb els companys, minimitzant errors i augmentant l'eficiència del desenvolupament, tots aquests punts formen un marc que ens permet lliurar un projecte de qualitat.

Freqüentment, practiquem el peer programming, una tècnica que ens ha ajudat significativament. La major part de les vegades, ens reunim per Discord, aprofitant la seva capacitat per compartir pantalla i comunicar-nos en temps real. Aquesta metodologia ens permet treballar junts en el codi, compartir diferents punts de vista i solucionar problemes de manera més eficaç. Hem observat que el peer programming augmenta la qualitat del nostre codi, ens ajuda a detectar errors més ràpidament i a aplicar millors pràctiques de programació des de l'inici. A més, compartir aquest temps ens permet aprendre l'un de l'altre, millorant les nostres habilitats i coneixements tècnics. La veritat és que aquesta experiència ha enriquit la nostra manera de treballar i continuarem amb aquesta pràctica.

Un altre aspecte clau del nostre procés és la integració amb eines com Swagger, que utilitzem per a la documentació i disseny d'APIs de manera clara i eficaç. Això facilita enormement la detecció i correcció d'errors, a més de permetre una millor comunicació entre els desenvolupadors. La integració amb Swagger ens permet definir, crear, documentar i utilitzar de manera efectiva les nostres APIs, garantint que s'ajustin perfectament a les necessitats del projecte i compleixin amb els requisits de qualitat que ens exigim.

2.5. Testing strategy

Per assegurar l'estabilitat i fiabilitat del nostre software, hem acordat amb l'equip adoptar una estratègia de proves que integra diferents nivells de testing dins del cicle del nostre desenvolupament. Vam començar fent servir una combinació de WebMvcTest per a provar les nostres APIs de Spring MVC, JpaTest per a validar la nostra capa de persistència, i en alguns casos

més complexos, SpringBootTest per a una simulació més completa de l'entorn de l'aplicació. Finalment ens hem decantat per fer-ho tot amb Spring Boot Test i i utilitzar H2 per emmagatzemar les dades per realitzar els test, en alguns casos pot ser excessiu però garanteix en un futur l'escalabilitat d'aquests.

Les proves amb dades de test es realitzen inicialment en local durant la implementació de funcionalitats. Això permet als desenvolupadors verificar el comportament del codi en un entorn controlat i amb dades específicament dissenyades per a aquest propòsit. Una vegada la funcionalitat està completa, es torna a fer aquest procés de verificació abans de fusionar els canvis a la branca de desenvolupament i havent sigut revisat a la request.

Un cop les funcionalitats són publicades a la branca de desenvolupament, es realitzen tests d'integració més representatius del comportament en un entorn similar al de producció. Així mateix, quan el codi és promogut entorn de producció, es duen a terme tests amb dades reals per a confirmar la integritat de les funcionalitats en el context final d'usuari. Bàsicament, fem proves a cada fase que es troba el codi per tal d'assegurar-nos.

En qualsevol moment que es detecti un error en aquestes fases, estem preparats per revertir els canvis ràpidament, prevenint l'impacte de qualsevol codi defectuós. Tot seguit, es prioritzarà la resolució del problema i se seguirà el nostre procés establert de revisió de codi per a garantir que la solució sigui efectiva i robusta. Aquests casos amb una correcta verificació de les merge-request haurien de ser molt puntuals.

És imprescindible que totes les proves d'històries d'usuari es compleixin satisfactòriament abans de considerar la possibilitat de fer un merge a la branca de desenvolupament. Per assegurar que quan s'executin els tests d'integració contínua a través del nostre sistema de CI, a GitHub Actions, cada nova incorporació al codi compleixi amb els requisits esperats tant per al front-end com al back-end. Les GitHub Actions s'activaran amb cada push, executant automàticament els tests unitaris i d'integració per a validar canvis recents i mantenir la nostra base de codi en un estat constant de llestesa per a la producció. Aquesta eina és la que ens servirà per fer el nostre deploy.

2.6. Management of configurations

En l'àmbit de la gestió de configuracions, de forma individual, abans de fer un merge amb develop cadascú serà responsable de fer ús de l'eina proporcionada per gradle "build", que és una pipeline generada pel propi gradle que entre moltes de les coses que fa, et comprova que el codi compila sense errors i fa l'execució completa dels tests.

També farem una pipeline de CI en Git, per comprovar el mateix que es comprova en fer un merge a develop, pero en aquest cas s'executarà quan es faci el merge a la branca master.

2.7. Interaction with colleagues

En la primera fase de la Inception 1 ens van informar sobre la necessitat de col·laborar amb dos dels altres grups, programant un servei per un d'ells i rebent un servei d'un altre grup. Després de fer una pluja d'idees, vam veure que el grup amb l'API que ens resultaria de major utilitat era CulturaConnect, encarregats de la gestió d'activitats culturals. El servei que els hi vam sol·licitar va ser el de rebre des de la seva aplicació d'horaris i disponibilitat de certs locals en una data determinada, per saber si un local en concret estaria disponible en tancar un litigi per poder realitzar un esdeveniment de clausura d'aquest. CulturaConnect ha acceptat la nostra proposta. En una classe presencial posterior, l'equip de Farm2Table va venir a parlar amb nosaltres per demanar-nos una extensió del nostre xatbot d'ajuda. La seva idea era crear un servei semblant al nostre, on els usuaris puguin plantejar dubtes i el sistema detecti el seu problema i el redirigeixi a un article d'ajuda. En poc temps vam acordar ampliar el nostre xatbot d'ajuda perquè pugui utilitzar-se també en la seva aplicació, i ho vam acceptar.

En la nostra iniciativa de millorar la interacció amb els altres grups de treball, hem implementat una estratègia eficaç a través de Discord. Hem creat canals específics i compartit enllaços de Discord amb cada grup, establint així un espai virtual dedicat per a coordinar i acordar els serveis a rebre i proporcionar. Aquest enfocament ens brinda una plataforma dinàmica i accessible per a la comunicació en temps real, facilitant la discussió de requisits, la resolució de problemes i la presa de decisions col·laborativa.

Tot i els canals mencionats anteriorment tenim una persona encarregada de la comunicació ambdós equips per tal d'estar informats de primera mà. Aquesta persona va actualitzant-se periòdicament per tal de tenir arribat el moment un document amb la informació necessària per desenvolupar i integrar les dues APIs.

2.8. Bug management

La nostra estratègia de comunicació i gestió de bugs s'articula a través de l'ús combinat de WhatsApp i un canal especialitzat d'issues. Quan sorgeix un bug, el protocol que hem creat inicia amb la creació d'una entrada en aquest canal del Discord d'issues, on s'adjunta una descripció detallada del problema juntament amb l'etiqueta de bug. Aquesta acció assegura que tot l'equip estigui al corrent de manera ràpida i eficaç.

Complementàriament, documentem i seguim tots els bugs utilitzant les issues de Taiga, que ens proporciona un control exhaustiu sobre la documentació i l'evolució de cada incidència. La resolució d'aquests bugs és majoritàriament prioritària; un cop detectats, els abordem segons un sistema de prioritats:

Alta: Les incidències d'alta prioritats són tractades de manera urgent, són incidències bloquejants per l'evolució del projecte.

Mitjana: Les de mitjana prioritats requereixen una resolució ràpida, però sense la urgència de les d'alta, normalment solen ser coses més puntuals de parts aïllades de l'aplicació, però que sí que afecten directament el funcionament d'aquesta.

Baixa: Les de baixa prioritats són incidències visuals o menys crítiques que s'aborden amb prou celeritat, però sense interrompre altres tasques evolutives, és important, tot i això, notificar-les sempre per tal de portar un registre adequat.

El Taiga ens permet, d'altra banda tenir-los assenyalats per colors i etiquetes, també hem afegit un plugin que automàticament envia un correu a l'hora de crear-los per tal de tenir a tothom al corrent en cas que no hagin mirat els altres canals de comunicació, aquest enfocament híbrid de comunicació garanteix que mantenim un flux de treball àgil i que tots els membres de l'equip estan ben informats i coordinats. Tots aquests bugs són tractats a part a les branques de fix del Github i posteriorment revisades, amb aquest enfocament integral, assegurem que el nostre producte mantingui un alt nivell de qualitat i funcionalitat.

Aquest bug una vegada s'ha detectat i informat amb totes les eines necessàries que hem vist anteriorment, es tracta a la branca de hotfix que es crea a partir de develop, on una vegada solucionat tornarà a passar els testos d'integració necessaris.

2.9. NFRs treatment

En la nostra anàlisi detallada, hem identificat i definit amb precisió els Requisits No Funcionals (NFR) específics que són crucials per a l'èxit del nostre projecte, en un punt anterior els trobem més detalladament. Ens hem centrat particularment en àrees crítiques que impacten directament en l'experiència de l'usuari i en la robustesa del sistema. Aquests inclouen els temps de resposta, on ens esforcem per assegurar una interactivitat àgil i eficient per als usuaris. Així mateix, hem avaluat i establert criteris específics per a l'escalabilitat del sistema, assegurant-nos que pugui manejar un augment sostingut en la càrrega de treball sense degradació del rendiment.

Cal remarcar que molts d'aquests NFRs es transformen en criteris d'acceptació i els gestionem al Taiga amb l'especificació de les històries d'usuari. A més, hem posat un èmfasi especial en la seguretat de dades, implementant mesures proactives per a salvaguardar la integritat i confidencialitat de la informació. Això inclou l'aplicació de pràctiques de codificació segura i l'encriptació de dades sensibles. En conjunt, la nostra identificació i definició clares dels NFR ens permeten abordar àrees crítiques amb enfocament i prioritat, assegurant que el nostre projecte compleixi amb estàndards elevats en termes de rendiment, escalabilitat i seguretat.

2.10. IDEs

Per dur a terme el nostre treball de desenvolupament, hem escollit utilitzar els entorns de desenvolupament integrats (IDEs) de JetBrains, una decisió clau per a la nostra arquitectura i realització de codi. Aquests IDEs ofereixen un conjunt d'avantatges significatius que encaixen perfectament amb les nostres necessitats tecnològiques i metodològiques que exposarem seguidament.

Per al desenvolupament de back-end en Java, hem seleccionat IntelliJ IDEA i pel que fa al desenvolupament de front amb Ionic hem escollit WebStorm. Aquests IDEs estan especialment dissenyats per a tecnologies de front-end i back-end respectivament, oferint suport especialitzat que facilita el desenvolupament d'aplicacions complexes i reactives.

Un altre punt fort dels IDEs de JetBrains és la seva integració amb eines de control de versions com git i GitHub. Aquests IDEs tenen capacitats excepcionals per a la gestió de repositoris, permetent-nos realitzar tota mena d'operacions de control de versions directament des de l'IDE. Això simplifica el flux de treball, ja que podem cometre canvis, gestionar branques, mergear codi i revisar el historial sense haver de sortir de l'entorn de desenvolupament.



Figura 2. JetBrains Logo

2.10.1. Plugins utilitzats i coding assistants

En el nostre projecte, estem aprofitant al màxim la funcionalitat dels IDEs de JetBrains i el seu marketplace mitjançant diferents plugins que milloren el nostre entorn de desenvolupament i optimitzen el nostre flux de treball. Aquest que explicarem a continuació són els principals, tot i que també tenim opcionalment alguns de personalització i database navigator entre altres.

SonarLint és un linter i analitzador de codi estàtic que s'integra directament amb l'IDE una vegada l'instal·lem. Ens ajuda a detectar bugs, vulnerabilitats de seguretat en el codi en temps real i altre tipus d'errors mentre codifiquem. L'utilitzem configurant els seus regles segons els estàndards de

qualitat de codi que hem adoptat en el nostre projecte. Quan SonarLint identifica una inconsistència o suggeriment de millora, ens mostra immediatament una alerta. Aquesta alerta inclou no només la notificació de l'error sinó també una explicació detallada de per què es considera un problema i recomanacions específiques per a la seva resolució. A més, l'eina remarca directament la línia o bloc de codi afectat dins de l'IDE, facilitant així que el desenvolupador pugui veure ràpidament on es necessita fer canvis.

El plugin de **Gradle** també és essencial per al nostre back-end Java, ja que utilitzem aquest sistema de construcció i automatització per gestionar les dependències, configurar els entorns i desplegar els nostres serveis.

En el nostre projecte, integrem **GitHub Copilot** a IntelliJ com a assistent de codificació principal. Aquesta eina d'IA proporciona suggeriments de codi en temps real, optimitzant el procés de desenvolupament i millorant l'eficiència. La seva capacitat per generar codi a partir del context actual accelera significativament la producció de codi de qualitat. Per tant, el plugin d'aquest també el tenim instal·lat en els bàsics del projecte.

En paral·lel, comptem amb l'assistència d'eines com Chat GPT o Geminis. Aquestes plataformes ens permeten realitzar consultes sobre qüestions de programació més generals o buscar exemples de codi i solucions a problemes comuns. Amb una àmplia base de dades de coneixement, aquestes eines ens donen respostes ràpides i ens ajuden a comprendre millor noves tècniques i pràctiques de programació, tot i que donen respostes molt generals a la majoria dels casos.

Finalment IntelliSense, per altra banda, ja ve integrat de sèrie amb els IDEs de JetBrains, que proporciona una completa funcionalitat de codificació assistida al projecte. Aquest sistema ens millora la velocitat amb què escrivim codi i també ens ajuda a reduir errors, ja que, ens suggereix noms de variables, mètodes, i corregeix errors sintàctics a mesura que escrivim.

2.11 Deploy de l'aplicació

Per organitzar eficientment els nostres desplegaments i assegurar que tot funciona com s'espera, hem planificat fer releases al final del segon i tercer sprint de cada cicle de desenvolupament. Aquestes releases estan dissenyades per coincidir amb les demos que presentem a classe, mostrant els avenços que tenim fins al moment.

Per a realitzar el desplegament de la nostra aplicació, el procés comença amb la creació d'una branca release a partir de la branca develop, on s'implementen totes les noves funcionalitats i millores. Durant aquesta etapa, es realitzen tests exhaustius per assegurar que el codi funcioni correctament. Aquest testing està automatitzat mitjançant GitHub Actions, el que ens permet executar tests cada cop que es fa un push a qualsevol branca, garantint així que els canvis introduïts no generen errors ni problemes.

Un cop la branca release passa tots els tests i està preparada, procedim a fer merge d'aquesta amb la branca master. Seguidament, marquem aquesta actualització amb un tag que reflecteix la versió del llançament, facilitant així la gestió de versions i el seguiment de qualsevol canvi específic. Això assegura que estem tots sincronitzats, tant el front com el back. A més, utilitzem etiquetes de commit específiques que clarament indiquen sobre què pertany el canvi i es descriu breument la funcionalitat o la correcció realitzada. Aquesta pràctica facilita la traçabilitat de cada canvi i ajuda a evitar confusions sobre quina versió de cada part de l'aplicació està sent desplegada.

És important destacar que mantenim bases de dades diferenciades per als entorns de producció i desenvolupament. Això és essencial per evitar que les dades de prova interfereixin amb les dades reals dels usuaris en producció. Aquesta separació també ens permet realitzar canvis i proves sense el risc d'afectar la integritat de les dades de producció. Inicialment hem tingut problemes amb el Virtech i hem decidit tenir la oficial per producció i per altra banda les locals per a realitzar canvis. Això ens ha sigut fàcil, ja que, la migració funciona correctament desde un principi i el propi script per crear les taules i actualitzar-ho amb un cron també.

Finalment, el desplegament de la versió final a l'entorn de producció es realitza mitjançant eines automatitzades que configurem segons les necessitats específiques del projecte (utilitzant també el Virtech). Aquesta fase de deploy inclou l'actualització dels servidors, la base de dades de producció i la verificació que tots els serveis funcionin correctament després de l'actualització (tenint en compte també els dels altres grups).

Per acabar per poder crear la APK hem utilitzat Android Studio, ja que ens proporciona moltes facilitats a l'hora de realitzar el deploy però per altra banda limita als sistemes operatius als quals es pot exportar finalment aquesta.

3. Technical description

3.1. Overall conception of the architecture

3.1.1. Physical architecture

L'arquitectura física d'un sistema és essencial per comprendre la seva estructura i com els components interactuen amb l'entorn físic. Aquest diagrama ens proporciona una representació visual dels elements hardware, connexions i distribució física que componen la infraestructura del nostre sistema.

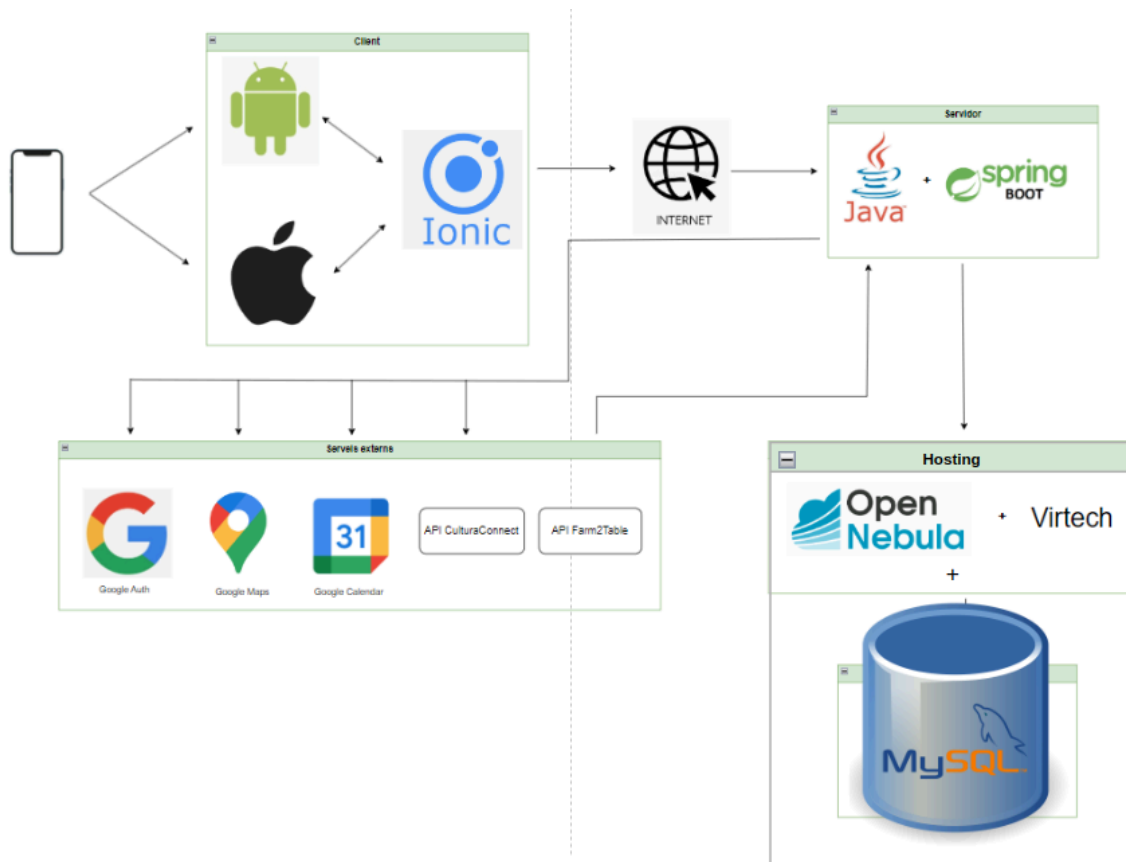


Figura 3. Diagrama complet de l'arquitectura física del sistema

Hosting

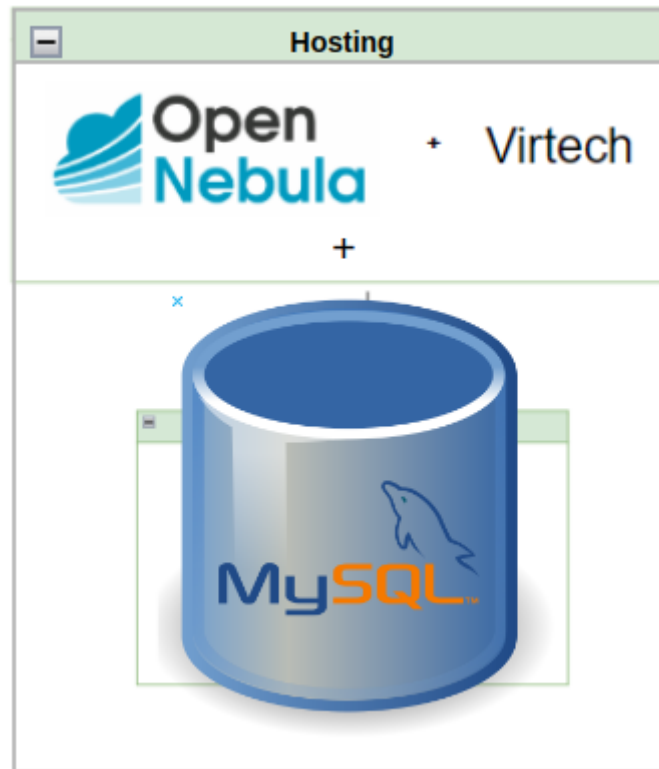


Figura 4. Diagrama del hosting

Hem decidit utilitzar Virtech com a hosting per la nostra aplicació a causa de la facilitat que ens proporciona la universitat donant-nos un entorn específic per crear, gestionar i utilitzar màquines virtuals de forma autònoma tot i que no siguin de grans capacitats.

També es va fer una anàlisi del que podíem trobar al mercat de manera gratuïta i vam veure que no hi hauria massa avantatja fent ús d'un altre host comparat amb la simplicitat que ens ofereix l'opció escollida.

En la decisió del tipus de base de dades, tots vam estar d'acord en fer-ne ús d'una base de dades relacional (en aquest cas MySQL) pel coneixement que ja tenim d'assignatures prèvies. També la integració que facilita Spring Boot amb JPA (Java Persistence API) fa que tingui sentit l'ús d'aquesta BBDD en la nostra aplicació.

Server



Figura 5. Diagrama del servidor

L'elecció d'utilitzar Java amb Spring Boot com a back-end per la nostra aplicació es pot basar en una sèrie de consideracions tècniques i beneficis que ens ofereix aquesta combinació.

- Spring Boot simplifica el desenvolupament de les aplicacions Java en proporcionar configuracions predeterminades convencionals. Això permet un desenvolupament més ràpid i centrar en la lògica de negoci sense la necessitat de configuracions extenses.
- Spring Boot facilita la integració amb tecnologies modernes, com serveis al cloud o sistemes de missatgeria, que permeten construir aplicacions modernes i escalables.
- Spring Boot simplifica la creació de APIs RESTful amb el seu enfocament basat en convencions.
- Tot el nostre equip ha treballat anteriorment desenvolupant en Java, per tant, no hauré de patir una corba d'aprenentatge que sí que hauré de patir en el front-end.

Serveis externs

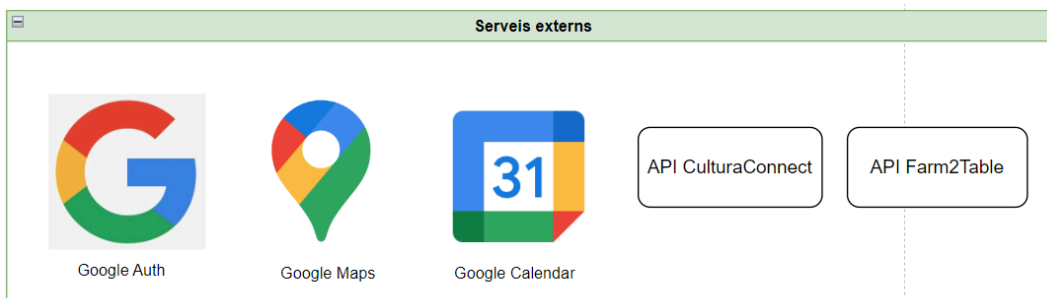


Figura 6. Diagrama dels serveis externs

Per a desenvolupar la nostra aplicació farem ús de serveis externs:

- De l'API de *Google* en farem molt d'ús, ja que donarem la possibilitat de fer l'autenticació amb el seu servei. També en necessitarem del *Google Maps* per mostrar en un mapa interactiu les licitacions existents. I finalment el *Google Calendar* per donar la possibilitat de sincronitzar totes les dates importants en el teu calendari principal del telèfon mòbil.
- CulturaConnect és una aplicació desenvolupada per un altre subgrup de la nostra classe. Ens ofereixen un servei per determinar si en un lloc, data i hora concrets es podria fer una reserva del local corresponent per realitzar un esdeveniment.

També nosaltres proporcionarem un servei a Farm2Table que consistirà a crear un sistema que donades unes paraules clau proporcionarem determinats articles relacionats amb la cerca demanada.

Client

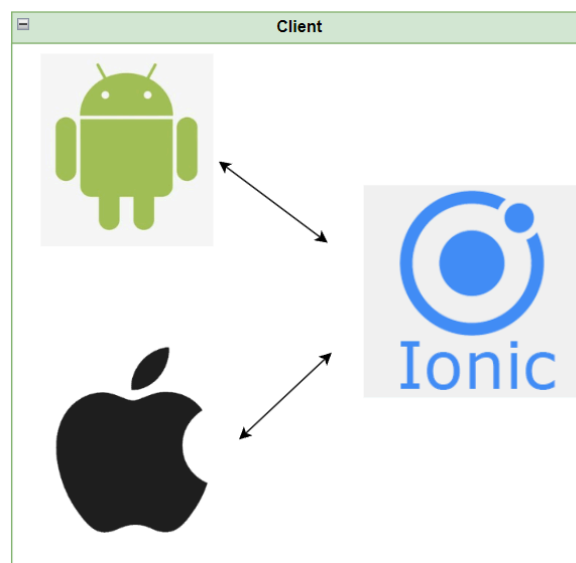


Figura 7. Diagrama del client

Pel desenvolupament del front-end hem escollit Ionic per diverses raons:

- La més important és que és un framework basat en Angular, el qual dues persones ja coneixen. Tenint en compte que cap altra persona en té coneixement d'un altre framework, hem decidit aprofitar l'experiència d'aquestes dues persones per fer no anar tot el grup perdut i reduir la corba d'aprenentatge.
- Ionic permet el desenvolupament multiplataforma. Això vol dir que amb sol codi font es pot desplegar l'aplicació en iOS, Android i, fins i tot, en la web.

- Ionic permet l'accés a les capacitats natives del dispositiu, com la geolocalització i les notificacions, de les quals en farem ús.

3.1.2. Architectural pattern(s) applied

Hem aplicat el patró Model-View-Controller (MVC):

- Model: Representa les dades del domini, en el nostre cas les Entity en el costat del servidor de cada taula de la BBDD.
- View: Es genera dinàmicament en el costat del client a través del framework frontend Ionic.
- Controller: En el costat de servidor els controllers són classes anotades amb `@Controller`, gestiona les sol·licituds del client, invoca la lògica de negoci apropiada i retorna una resposta adequada.

Aquest patró ens ajuda en la separació de les responsabilitats com ja hem explicat; en l'escalabilitat, ja que en separar les preocupacions és més fàcil fer canvis en una part de l'aplicació sense afectar les altres; i en el manteniment, ja que la claredat que proporciona MVC facilita als desenvolupadors a entendre ràpidament com estan organitzades les diferents parts de l'aplicació i realitzar canvis de manera més segura.

3.2. Domain layer

3.2.1. Domain model diagram (optional)

El diagrama UML es pot trobar a l'apartat 1.3

Exemple del flux del domini per una classe exemple Persona:

Domini del backend:

Aquesta imatge que tenim a continuació correspon al domini del back-end, les dues línies que podem observar corresponen a l'entrada d'informació desde el controlador i a la comunicació amb el repositori, d'esquerra a dreta respectivament.

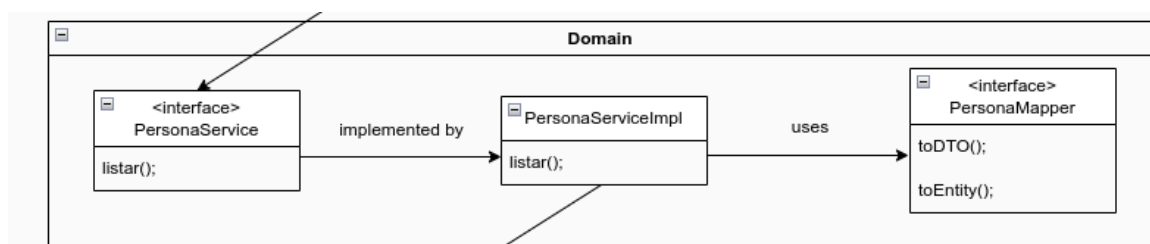


Figura 8. Diagrama domain model back

Domini del frontend:

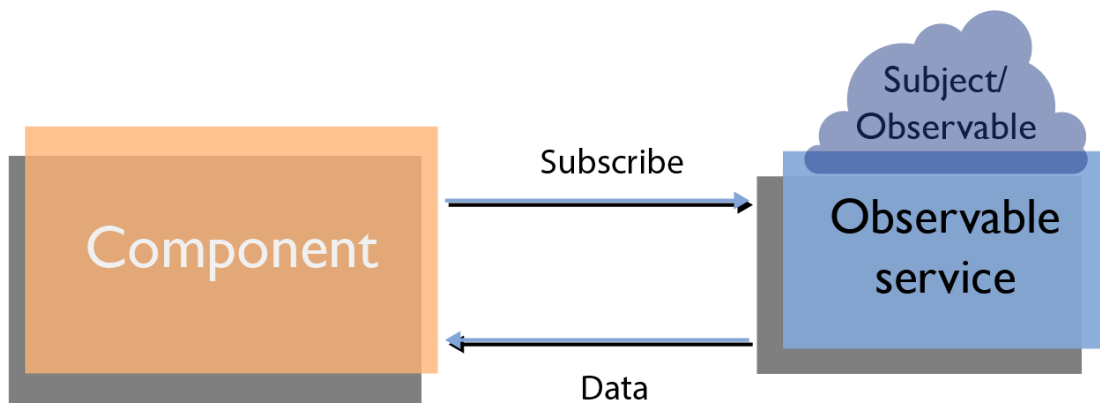


Figura 9. Diagrama domain model front

3.2.2. Design patterns applied

Pel que fa a patrons purament de disseny, ens vam plantejar quins dels patrons coneguts ens podia ser d'utilitat. Vam descartar d'aplicar cap patró d'estratègia i d'estat, ja que no s'aplicava per la lògica de la nostra aplicació. Parlant del patró plantilla ens vam plantejar si aplicar-ho pels usuaris, perquè tenim diferents tipus d'usuari segons el seu rol, però vam decidir no aplicar-ho, ja que les funcions que necessitem per cada un no varien en implementació. Per altra banda vam aplicar els següents patrons tant al back com al front.

BACKEND

Mapper

El patró Mapper, és una tècnica per mapejar dades entre objectes que no necessàriament tenen una correspondència directa entre els seus atributs. En Java Spring, aquest patró l'hem utilitzat per convertir dades entre la capa de domini/model i els DTOs (Objectes de Transferència de Dades). L'objectiu és separar la lògica de negoci de la lògica de la base de dades, facilitant així la mantenibilitat i l'escalabilitat del codi.

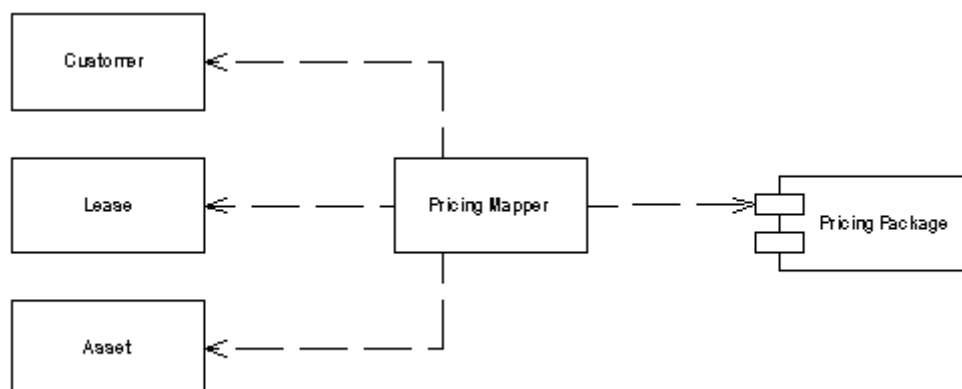


Figura 10. Mapper theory example

```

4 usages  msardamasri +1
@Mapper(componentModel = "spring", unmappedTargetPolicy = ReportingPolicy.IGNORE,
        uses = {UsuariMapper.class})
public interface MissatgeMapper {

    pol.perez.castillo
    MissatgeDTO toDTO(MissatgeEntity entity);

    pol.perez.castillo
    MissatgeEntity toEntity(MissatgeDTO dto);

1 usage  msardamasri
    @Mapping(target = "idemissor", source = "emissor.idUsuari")
    @Mapping(target = "idreceptor", source = "receptor.idUsuari")
    MissatgeCustomDTO toCustomDTO(MissatgeDTO missatgeDTO);

1 usage  msardamasri
    default Page<MissatgeCustomDTO> toPageCustomDTO(Page<MissatgeDTO> missatgeDTOPage) {
        return missatgeDTOPage.map(this::toCustomDTO);
    }
}

```

Figura 11. Mapper usage example

Aquest és un exemple del nostre codi on s'utilitza el patró del *Mapper*, degut a la separació de capes que tenim, existeix un mapper per a cada entitat diferent.

Repository

El patró Repository actua com una capa d'abstracció entre la lògica del negoci i la capa d'accés a dades. En el context de Java Spring, aquest patró l'hem implementat en Spring Data JPA. Els repositoris en Spring ens han permès tractar les operacions de la base de dades d'una manera orientada a objectes, on les taules es poden manejar com col·leccions d'entitats. Per exemple, mètodes com `findById()` ens permeten recuperar entitats específiques sense necessitat d'escriure consultes SQL detallades.

```

4 usages msardamasri +2
public interface ContracteRepository extends JpaRepository<ContracteEntity, Long> {

1 usage msardamasri
    @Query("SELECT c.codiExpedient FROM ContracteEntity c")
    List<String> findAllCodiExpedients();

1 usage msardamasri
    @Query("SELECT c FROM ContracteEntity c WHERE " +
        "(:codiExpedient IS NULL OR c.codiExpedient LIKE %:codiExpedient%) AND " +
        "(:tipusContracte IS NULL OR c.tipusContracte LIKE %:tipusContracte%) AND " +
        "(:ambit IS NULL OR c.ambit.nom LIKE %:ambit%) AND " +
        "(c.llocExecucio LIKE %:llocExecucio% OR :llocExecucio IS NULL) AND " +
        "(:valorContracte IS NULL OR c.valorEstimatContracte < :valorContracte) AND " +
        "(:fechaActual IS NULL OR c.terminiPresentacioOfertes > :fechaActual) AND " +
        "(c.objecteContracte LIKE %:objecteContracte% OR :objecteContracte IS NULL) AND " +
        "(c.procediment LIKE %:procediment% OR :procediment IS NULL)")
    Page<ContracteEntity> findByFilterPaginated(@Param("codiExpedient") String codiExpedient,
        @Param("tipusContracte") String tipusContracte,
        @Param("ambit") String ambit,
        @Param("procediment") String procediment,
        @Param("llocExecucio") String llocExecucio,
        @Param("valorContracte") Double valorContracte,
        @Param("objecteContracte") String objecteContracte,
        @Param("fechaActual") Date fechaActual,
        Pageable pageable);

```

Figura 12. Repository usage example

Aquest és un exemple del nostre codi on s'utilitza el patró del *Repository*, existeix un repository per a cada entitat diferent i aquestes extenen JPA com es mostra a la capçalera de la classe, seguidament hi ha una query d'exemple que utilitzem per a buscar amb filtres.

Singleton

El patró Singleton garanteix que una classe té una única instància en l'aplicació i proporciona un punt d'accés global a ella. En Spring, aquest patró s'utilitza de manera predeterminada per a la gestió de beans; és a dir, quan defines un bean en el context de Spring, aquest s'instancia com un singleton a menys que s'especifiqui el contrari. Això ens ha sigut útil per mantenir la coherència i reduir l'ús de memòria.

```

msardamasri +1
@Service
@RequiredArgsConstructor
public class ContractServiceImpl implements ContractService {

    private final ContractRepository contractRepository;
    private final ContractMapper contractMapper;
    private final AmbitService ambitService;
    private final LotService lotService;
    private final FavoritsService favoritsService;
  
```

Figura 13. Singleton usage example

En aquesta imatge de la implementació d'un servei tenim el patró singleton, que amb l'ajuda de la llibreria Lombok i el no requeriment d'arguments per el constructor tenim una sola instància d'aquests objectes.

Decorator i Injecció de Dependències

El patró Decorator permet afegir funcionalitats a objectes de manera dinàmica, i s'implementa en Java a través de la composició. Spring ens facilita aquest patró mitjançant la injecció de dependències, permetent que els objectes siguin configurats i ampliat amb noves responsabilitats de forma neta i desacoblada. La injecció de dependències en Spring redueix la necessitat de codi boilerplate i augmenta la flexibilitat del codi.

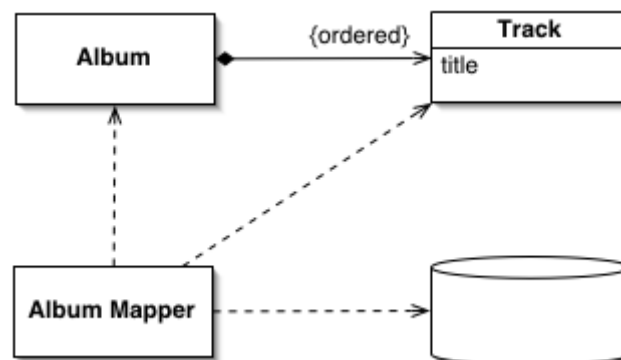


Figura 14. Injection dependency theory usage example

Utilització de DTOs

Els DTOs (Objectes de Transferència de Dades) s'utilitzen per transferir dades entre processos, en aquest cas, entre la capa de negoci i els clients. En el nostre projecte, els DTOs ens ajuden a separar la representació de les dades de la lògica de negoci interna de l'aplicació, proporcionant una capa d'abstracció que facilita, per exemple, la serialització i la transmissió de dades.

```
16 usages  👤 msardamasri
@Data
@AllArgsConstructor
@NoArgsConstructor
@Builder
public class FiltreDTO {
    private String codiExpedient;
    private String tipusContracte;
    private String ambit;
    private String procediment;
    private String llocExecucio;
    private Double valorContracte;
    private String objecteContracte;
}
```

Figura 15. DTOs usage example

Capa de Servei

La capa de servei en una aplicació Spring és on es maneja la lògica de negoci. Actua com un mediador entre la capa de presentació i la capa d'accés a dades, gestionant les transaccions, la lògica de negoci, i les crides als mètodes dels repositoris. És aquí on nosaltres hem integrat les operacions d'alt nivell que l'aplicació ha de realitzar.

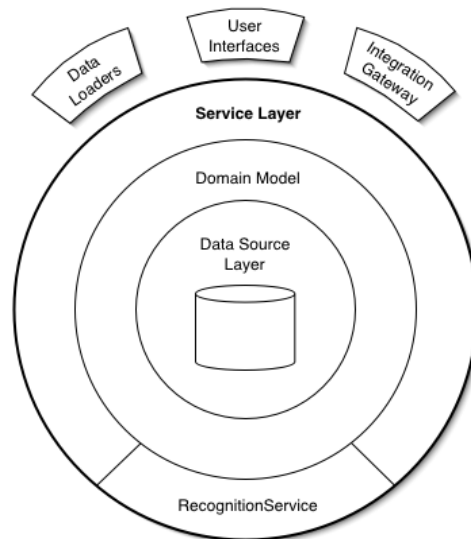


Figura 16. Service theory example

```

7 usages 1 implementation msardamasri +2
public interface AmbitService {

    1 usage 1 implementation msardamasri
    void saveAmbits(List<AmbitDTO> AmbitDTOs) throws AdjudicatBaseException;

    1 usage 1 implementation msardamasri
    Set<String> getAllCodiAmbit();

    1 usage 1 implementation msardamasri
    AmbitDTO getAmbit(Long codi);

    1 implementation XinXZP
    List<AmbitDTO> listar();
}

```

Figura 17. Service layer usage example

```
msardamasri +2
@Service
@RequiredArgsConstructor
public class AmbitServiceImpl implements AmbitService {

    private final AmbitRepository ambitRepository;
    private final AmbitMapper ambitMapper;

    1 usage msardamasri
    @Override
    @Transactional
    public void saveAmbits(List<AmbitDTO> AmbitDTOs) throws AdjudicatBaseException {
        for (AmbitDTO dto : AmbitDTOs) {
            try {
                ambitRepository.save(ambitMapper.toEntity(dto));
            } catch (DataAccessException e) {
                throw new RepositoryException(e, RepositoryConstants.ERROR_BBDD);
            }
        }
    }
}
```

Figura 18. Service implementation example

En aquests últims dos exemples de codi tenim exemples de com implementem els serveis en el nostre projecte, primer les funcions definides a les interfícies i seguint implementades a la classe on es controla el negoci d'aquestes. Aquest també fa que es compleixi el principi de Liskov.

Inversió de Control a Spring

La Inversió de Control (IoC) és un principi de disseny que s'implementa a Spring a través del contenidor IoC. En lloc que els objectes creïn o gestionin les seves dependències, aquestes són injectades pel contenidor Spring. Un exemple que tenim al projecte és la configuració de la seguretat web, on Spring maneja la creació d'objectes i la gestió del seu cicle de vida, permetent que els desenvolupadors es concentren en implementar la lògica específica de l'aplicació.

FRONTEND

Observer dels Endpoints

El patró Observer s'utilitza en el desenvolupament frontend per gestionar la comunicació entre els components i els serveis que interactuen amb endpoints (APIs). En el nostre context d'Angular, aquest patró es pot implementar mitjançant l'ús de RxJS (Reactive Extensions for JavaScript), que permet a components subscriure's a observables. Aquests observables poden representar respostes d'endpoints. Quan un endpoint és cridat i retorna dades, l'observable notifica tots els seus observadors (subscriptors), que aleshores poden reaccionar a aquestes dades, actualitzant la UI o processant les dades com sigui necessari.

```

1+ usages  paucampillo
ultimesLicitacions(idContracte: number) : Observable<any> {
  return this.get(endpoint: this.HISTORIC + `/${idContracte}`);
}

1+ usages  paucampillo
loadOferta() : void {
  this.licitarService.ultimesLicitacions(this.contracte.idContracte).subscribe(next: ultimesLicitacions => {
    this.ultimesLicitacions = ultimesLicitacions;

    if (this.ultimesLicitacions.length > 0) {
      this.preuMesBaix = this.ultimesLicitacions[0].importAdjudicacioAmbIva;
      this.calculateFontSize();
    }
  });
}

```

Figura 19. Observer implementation example

Implementació de Components a Angular

Angular es basa en una arquitectura de components, on cada component gestiona una part de la UI i la lògica associada a aquesta. Un component a Angular es defineix utilitzant una classe decorada amb `@Component`, que especifica la plantilla HTML, els estils i altres metadades. Els components són modulars i reutilitzables, facilitant així el manteniment i l'escalabilitat de les aplicacions. Angular també proporciona un sistema de binding de dades (two-way data binding), que permet una interacció fluida entre el model de dades i la vista.

```

1+ usages  julia +3
@Component({
  selector: 'app-home',
  templateUrl: 'home.page.html',
  styleUrls: ['home.page.scss'],
  standalone: true,
  providers: [TranslateService],
  imports: [MenuComponent, IonicModule, TarjetaContracteComponent, NgForOf, RouterLink,
    FiltrosComponent, NgIf, HeaderGenericoComponent, FormsModule, TranslateModule],
})

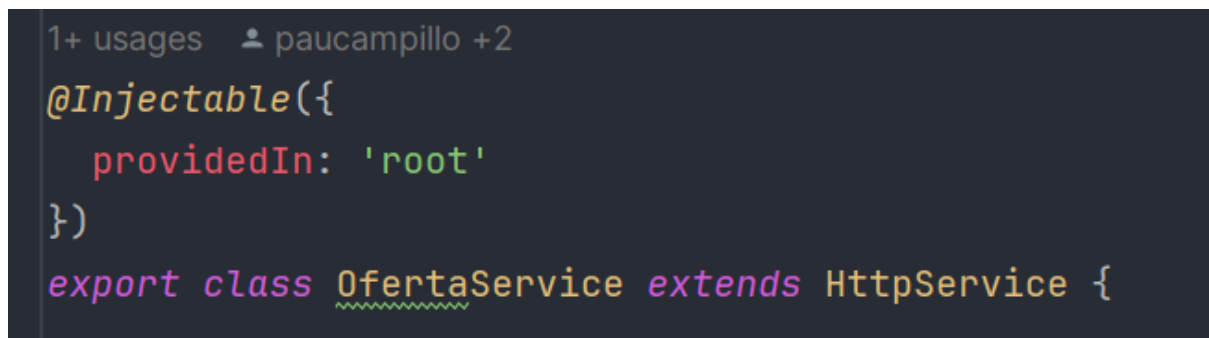
export class HomePage implements OnInit {

```

Figura 20. Component implementation example

Connector

En Angular, el patró Connector (conegut més formalment com a Injecció de Dependències) és una tècnica clau per a la gestió de dependències entre classes. Angular utilitza decoradors com `@Injectable`, `@Component`, entre d'altres, per definir com les dependències són proporcionades i injectades. La injecció de dependències permet desacoplar les classes de les seves dependències, facilitant així el testing i millorant la mantenibilitat del codi. Per exemple, un servei decorat amb `@Injectable` pot ser injectat en un component, proporcionant-li funcionalitats com la comunicació amb un backend sense que el component necessiti saber com es realitzen aquestes crides.



```
1+ usages  👤 paucampillo +2
@Injectable({
  providedIn: 'root'
})
export class OfertaService extends HttpService {
```

Figura 21. Connector implementation example

3.2.3. S.O.L.I.D.

PRINCIPI DE RESPONSABILITAT ÚNICA (S) : En el nostre disseny, cada interface i cada classe tenen responsabilitats úniques.

PRINCIPI D'OBERT/TANCAT (O) : El nostre disseny permet l'extensió de les funcionalitats sense modificar el codi existent.

PRINCIPI DE SUBSTITUCIÓ DE LISKOV (L) : Pel nostre projecte si aplica, degut a que és una aplicació amb diferents interfícies preparat per tenir varies implementacions per a cada funcionalitat del servei.

PRINCIPI DE SEGREGACIÓ D'INTERFÍCIES (I): En el nostre disseny, tenim dividit mínim per cada entity les seves interfaces específiques.

PRINCIPI D'INVERSIÓ DE DEPENDÈNCIES (D): En el nostre cas tenim les interfaces definides en alt nivell i les implementacions concretes en un nivell més baix. Això ens facilita el desacoblament entre components i facilita la reutilització de codi.

3.3. Database diagram (UML)

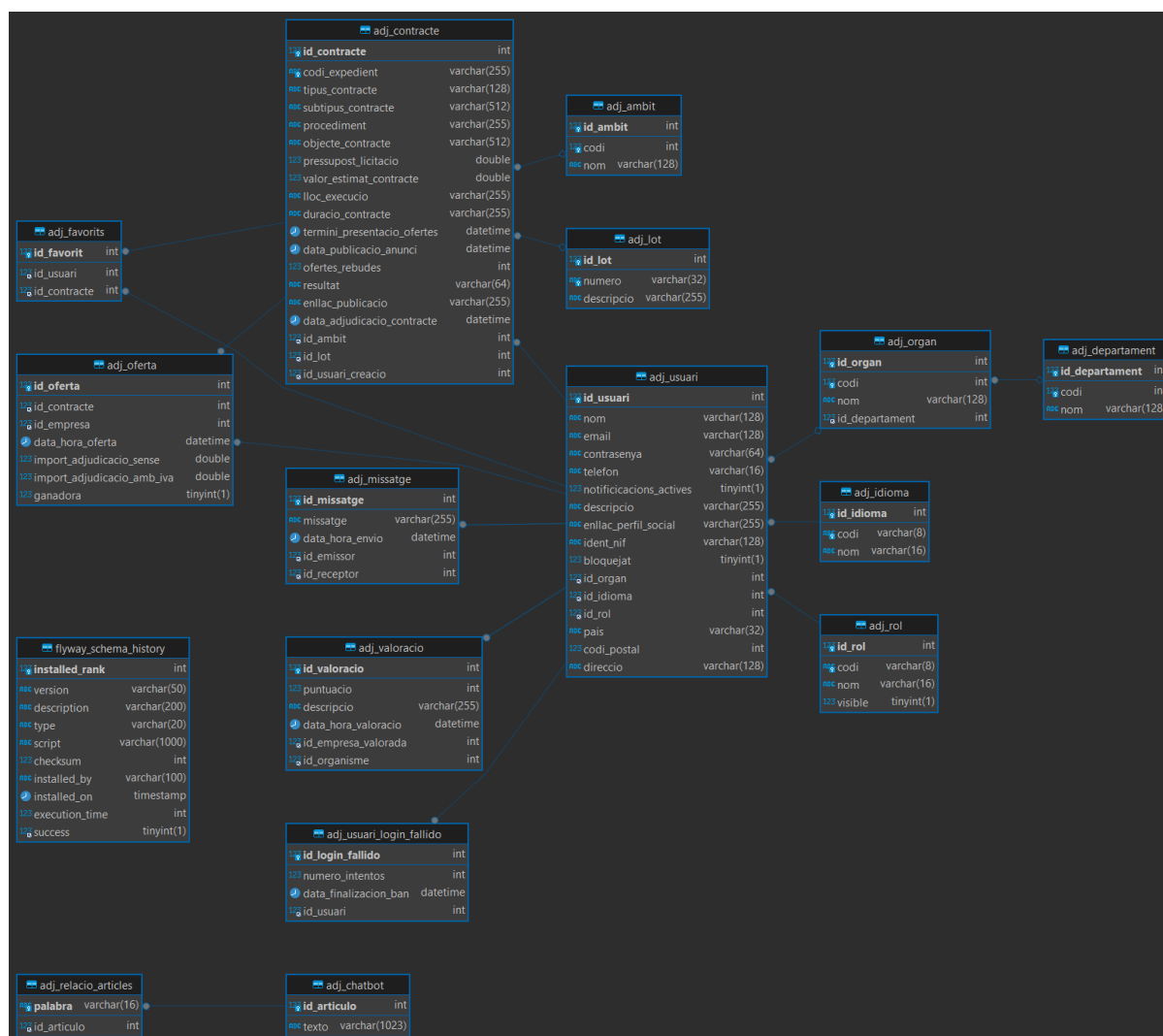


Figura 22. Diagrama BD

(Les relacions de claus foranes del UML estan representades al costat del tipat de l'atribut, indicat amb un requadre blanc petit amb una flecha dins)

3.4. Instrumentation and list of technologies

En el nostre sistema utilitzem servidors Java amb Spring Boot per al back-end, per la seva capacitat de simplificar el desenvolupament d'aplicacions Java oferint configuracions predeterminades i facilitant la integració amb tecnologies modernes i la creació d'APIs restful. Aquesta elecció es deu a l'experiència prèvia del nostre equip amb Java, així com als avantatges que Spring Boot aporta en termes de desenvolupament ràpid i enfocament basat en convencions.

Pel que fa a les bases de dades, hem optat per utilitzar MySQL, una base de dades relacional i *open source*, a causa del coneixement previ del nostre equip i la integració suportada per Spring Boot a

través de JPA (una llibreria utilitzada en aquest context). Això ens permet gestionar les dades de manera eficient i mantenir una estructura de dades coherent que suporta les operacions del nostre sistema. La gestió de versions de la base de dades es realitza mitjançant eines de migració que permeten actualitzar l'esquema de la base de dades de manera controlada i segura a mesura que el projecte evoluciona.

Finalment, en el desenvolupament del nostre sistema, fem ús del llenguatge de programació Java per al back-end, aprofitant les facilitats que ofereix Spring Boot. Per al front-end, utilitzem Ionic, un framework que ens permet crear aplicacions multiplataforma amb Angular, facilitant l'ús de capacitats natives del dispositiu com la geolocalització i les notifikacions. Angular fa ús de diferents llenguatges de programació, entre ells tenim: typescript per la lògica i css i html per l'estil. Aquestes eleccions de tecnologies reflecteix el nostre enfocament cap a la creació d'una aplicació escalable i accessible des de diferents plataformes tant mòbils com web si fos el cas, aprofitant les habilitats prèvies i l'experiència del nostre equip.

3.5. APIs

El nostre projecte incorpora una API restful que permet la integració amb altres aplicacions i serveis per proporcionar accés a dades relacionades amb licitacions i concursos públics. Aquesta API permet als usuaris accedir a diverses funcionalitats de la nostra aplicació, com la cerca de contractes de licitació, la gestió d'usuaris i la interacció amb els contractes disponibles.

Estem utilitzant diverses APIs externes. D'una banda, emprarem una API de Google Maps per la integració de mapes interactius que mostrin la ubicació de les empreses adjudicatàries de licitacions. D'altra banda, integrarem l'API de Google Calendar en el sistema per permetre als usuaris programar esdeveniments relacionats amb les licitacions, com dates límit de presentació d'ofertes o altres esdeveniments relacionats amb el procés de licitació. Finalment, usem Google OAuth perquè els usuaris usin les seves credencials de Google per iniciar sessió a la plataforma de manera segura.

El consum de serveis es realitza principalment a través de la nostra API interna. Els diferents components de la nostra aplicació, tant del front-end com del back-end, interactuen amb aquesta API per obtenir informació necessària.

Pel que fa al subministrament de serveis als companys d'equip, l'API interna és clara i fàcil d'utilitzar. Hem documentat cada punt de l'API, els paràmetres necessaris i les respostes esperades i estem en constant comunicació per informar de qualsevol canvi o actualització que es fa.

Per consumir dades obertes de la Generalitat de Catalunya, estem fent servir l'API proporcionada per Virtech. Aquesta ens permet accedir a informació rellevant sobre licitacions i concursos públics en temps reals.

3.6. Development tools and working environment

Per dur a terme el projecte hem fet servir diferents frameworks. Pel desenvolupament del back-end utilitzem Spring, un marc de treball que aporta escalabilitat i robustesa, permetent que l'aplicació sigui sòlida i eficient. El front-end l'hem desenvolupat amb Ionic, una plataforma que ens permet crear aplicacions mòbils i web amb un únic codi base. Aquesta elecció ens donava flexibilitat per desenvolupar interfícies atractives, funcionals i coherents. Hem comptat amb el suport de WebStorm per optimitzar el flux de treball.

Amb relació a la gestió del nostre repositori de codi, hem optat per una estratègia GitFlow, que ens proporciona una manera clara i organitzada per treball conjuntament sobre el mateix codi. Hem pogut garantir una interacció fluida de les diferents contribucions de codi amb les branques del repositori.

Per assegurar la qualitat i estabilitat del nostre software, hem implementat una estratègia de proves que engloba diferents nivells i tipus de prova. Hi ha proves unitàries per validar el comportament de components individuals, proves d'integració per verificar la interacció entre diferents mòduls de l'aplicació i proves funcionals per avaluar el comportament amb més perspectiva.

Finalment, el procés de desplegament del nostra software ha sigut amb GitHub Continuous Deployment. Aquesta eina permet automatitzar l'entrega de noves versions de la nostra aplicació de manera consistent i eficient, reduint el risc d'errors que nosaltres podríem fer.

