



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escola Tècnica Superior d'Enginyeria Informàtica

ConnecTFG: Sistema de Missatgeria i Notificacions

Treball Fi de Grau

Grau en Enginyeria Informàtica

AUTOR/A: Martín Torres, Josep

Tutor/a: Penadés Gramage, María Carmen

CURS ACADÈMIC: 2022/2023



## Agraïments

---

Després de diversos mesos de treball, vull expressar el meu sincer agraïment a totes les persones que han estat al meu costat durant aquest trajecte. En primer lloc, agrair a Matías Leandro Fraga per brindar-me la oportunitat de col·laborar amb ell en aquest projecte. La seua dedicació, destresa, així com la seua capacitat per a treballar en equip i la bona energia que transmet, han sigut essencials per al desenvolupament d'un treball de qualitat i una gran experiència.

Seguint amb les persones que han participat en aquest projecte, m'agradaria agrair a María Carmen Penades Gramage, la persona encarregada de tutoritzar aquest treball, per la seua guia, que ens ha permès millorar la qualitat i l'abast de ConneCTFG. La seua orientació, recomanacions i suport han estat crucials en el desenvolupament d'aquest projecte.

També vull agrair als meus amics, que sempre han estat al meu costat, pel seu recolzament quan més falta feia. En especial a Eric, Gonzalo, Pau, Pablo i Jordi, qui va suggerir el nom final del projecte. El vostre suport incondicional, ànims i rialles han sigut clau en aquest treball. Sou un recordatori constant que l'equilibri entre la vida acadèmica i personal és essencial per a l'èxit.

A més, m'agradaria agrair a Ramón per la seua opinió crítica en la part de disseny. Gràcies al seu punt de vista em pogut oferir una millor experiència d'usuari, al corregir les errates de navegabilitat amb les que comptava l'aplicació inicialment.

No vull oblidar-me d'Òscar Mateo Martínez, qui em va brindar l'oportunitat de fer les pràctiques en COM3 Studio i em va ensenyar les bases del que s'ha emprat en aquest projecte. Sense la seua guia en Angular, Node, Express i Socket.io no hauria sigut capaç de desenvolupar aquest projecte.

Per finalitzar, vull agrair als meus pares i germana per el seu amor incondicional i la confiança que sempre han tingut amb mi quan ni jo la tenia. Sense ells no hauria sigut capaç de finalitzar aquest trajecte.



# Resum

El present Treball de Fi de Grau (TFG) es centra en l'exploració, disseny i desenvolupament de ConnectTFG, una aplicació web que aspira a millorar la relació entre l'estudiantat de diferents graus a través d'una xarxa social per a realitzar projectes multidisciplinaris en els seus TFG.

ConnectTFG és un projecte col·laboratiu, on s'ha treballat de manera conjunta amb el company del Grau en Enginyeria Informàtica, Matías Leandro Fraga, qui ha estat un component clau. Matías ha treballat principalment en el sistema de xarxa social i gestió d'usuaris, mentre que la meua part englobava el sistema de missatgeria i notificacions. Tot i això, ambdós hem participat de forma activa en la realització de tota l'aplicació, sent la comunicació entre els dos imprescindible.

L'objectiu principal de ConnectTFG és facilitar la interacció i la comunicació entre estudiants d'àmbits diversos, amb la finalitat de crear col·laboracions fructíferes en els seus TFG. A través d'aquesta plataforma, els usuaris poden trobar possibles companys/es de TFG amb interessos i objectius comuns, amb qui establir converses privades per aclarir dubtes i decidir si desitgen treballar conjuntament.

Un component clau de ConnectTFG que s'analitza en aquest treball és el sistema de missatgeria i notificacions. Aquest sistema permet als usuaris interactuar en temps real amb altres membres de la xarxa social, afavorint la discussió d'idees i la coordinació dels seus esforços en els respectius TFGs. A més, el sistema de notificacions de ConnectTFG és una funcionalitat valiosa que manté als estudiants informats de les noves actualitzacions dins de l'aplicació. Aquesta característica permet als usuaris rebre alertes sobre noves sol·licituds de connexió o confirmació d'acceptació per a altres projectes als quals s'han postulat.

Respecte a les tecnologies emprades, s'ha utilitzat el *framework* Angular per al *frontend* i Node.js, Express, MongoDB i Socket.io per al *backend*. A més, s'ha seguit una metodologia àgil Kanban.

**Paraules clau:** Xarxa Social, TFG, Multidisciplinari, Aplicació Web, Kanban

---



# Resumen

El presente Trabajo de Fin de Grado se centra en la exploración, diseño y desarrollo de ConneCTFG, una aplicación web que aspira a mejorar la relación entre el estudiantado de diferentes grados a través de una red social para realizar proyectos multidisciplinarios en sus TFG.

ConneCTFG es un proyecto colaborativo, donde se ha trabajado de forma conjunta con el compañero del Grado en Ingeniería Informática, Matías Leandro Fraga, quien ha sido un componente clave. Matías ha trabajado principalmente en el sistema de red social y gestión de usuarios, mientras que mi parte englobaba el sistema de mensajería y notificaciones. Sin embargo, ambos hemos participado de forma activa en la realización de toda la aplicación, siendo la comunicación entre los dos imprescindible.

El objetivo principal de ConneCTFG es facilitar la interacción y la comunicación entre estudiantes de diferentes ámbitos, con el fin de crear colaboraciones fructíferas en sus Trabajos de Fin de Grado. A través de esta plataforma, los usuarios pueden encontrar posibles compañeros/as de TFG con intereses y objetivos comunes, con los que establecer conversaciones privadas para despejar dudas y decidir si desean trabajar conjuntamente.

Un componente clave de ConneCTFG que se analiza en este trabajo es el sistema de mensajería y notificaciones. Este sistema permite a los usuarios interactuar en tiempo real con otros miembros de la red social, favoreciendo la discusión de ideas y la coordinación de sus esfuerzos en sus respectivos TFGs. Además, el sistema de notificaciones de ConneCTFG es una valiosa funcionalidad que mantiene a los estudiantes informados de las nuevas actualizaciones dentro de la aplicación. Esta característica permite a los usuarios recibir alertas sobre nuevas solicitudes de conexión o confirmación de aceptación para otros proyectos a los que se han postulado.

Respecto a las tecnologías utilizadas, se ha utilizado el *framework* Angular para el *frontend* y Node.js, Express, MongoDB y Socket.io para el *backend*. Además, se ha seguido una metodología ágil Kanban.

**Palabras clave:** Red Social, TFG, Multidisciplinario, Aplicación Web, Kanban

---





# Abstract

This Final Degree Thesis (TFG) focuses on the exploration, design and development of *ConneCTFG*, a web application that aspires to improve the relationship between students of different degrees through a social network to carry out multidisciplinary projects in their TFG.

*ConneCTFG* is a collaborative project, where we have worked together with the Computer Engineering degree classmate, Matías Leandro Fraga, who has been a key component. Matías has worked mainly on the social network system and user management, while my part included the messaging and notification system. Despite this, we have both actively participated in the realization of the entire application, communication between the two being essential.

The main objective of *ConneCTFG* is to facilitate interaction and communication between students from various fields, with the aim of creating fruitful collaborations in their Final Degree Thesis. Through this platform, users can find potential TFG colleagues with common interests and goals, with whom to establish private conversations to clarify doubts and decide if they wish to work together.

An essential component of *ConneCTFG* that is analyzed in this work is the messaging and notification system. This system allows users to interact in real time with other members of the social network, promoting the discussion of ideas and the coordination of their efforts in the respective TFGs. Additionally, *ConneCTFG*'s notification system is a valuable feature that keeps students informed of new updates within the app. This feature allows users to receive alerts about new connection requests or confirmation of acceptance for other projects to which they have applied.

Regarding the technologies used, the Angular framework has been used for the frontend and Node.js, Express, MongoDB and Socket.io for the backend. In addition, an agile Kanban methodology was followed.

**Key words:** Social Network, TFG, Multidisciplinary, Web Application, Kanban

---



# Índex

---

<b>Índex</b>	<b>xi</b>
<hr/>	
<b>Índex de figures</b>	<b>xv</b>
<b>Índex de taules</b>	<b>xvii</b>
<b>1 Introducció</b>	<b>1</b>
1.1 Motivació . . . . .	1
1.2 Objectius . . . . .	2
1.3 Estructura de la memòria . . . . .	3
<b>2 Estudi del mercat</b>	<b>5</b>
2.1 Idea de negoci . . . . .	5
2.2 Competidors . . . . .	6
2.3 Anàlisi DAFO . . . . .	8
<b>3 ConneCTFG</b>	<b>11</b>
3.1 Funcionalitats . . . . .	11
3.2 <i>Backlog</i> . . . . .	11
3.3 Arquitectura . . . . .	12
<b>4 Tecnologia</b>	<b>15</b>
4.1 <i>Frontend</i> . . . . .	15
4.1.1 Angular . . . . .	15
4.1.2 HTML i SCSS . . . . .	16
4.2 Backend . . . . .	17
4.2.1 Node.js . . . . .	17
4.2.2 Express . . . . .	17
4.2.3 Socket.IO . . . . .	17
4.2.4 MongoDB . . . . .	17
4.3 Ferramentes externes . . . . .	18
4.3.1 Git . . . . .	18
4.3.2 GitHub . . . . .	18
4.3.3 Worki . . . . .	18
<b>5 Desenvolupament</b>	<b>19</b>
5.1 Organització en <i>sprints</i> . . . . .	19

5.2	<i>Sprint 0</i>	20
5.2.1	Model de domini	20
5.2.2	Avaluació de l' <i>Sprint</i>	21
5.3	<i>Sprint 1</i>	21
5.3.1	Xat	22
5.3.2	Avaluació de l' <i>Sprint</i>	30
5.4	<i>Sprint 2</i>	32
5.4.1	Gestió de candidats	33
5.4.2	Sistema de notifikacions	35
5.4.3	Avaluació de l' <i>sprint</i>	39
5.5	<i>Sprint 3</i>	40
5.5.1	Enviament d'arxius al xat	41
5.5.2	Internacionalització	44
5.5.3	Avaluació de l' <i>sprint</i>	46
<b>6</b>	<b>Desplegament</b>	<b>49</b>
6.1	Introducció	49
6.2	<i>Frontend</i> (Firebase)	49
6.3	<i>Backend</i> (Render)	52
6.4	Escenari d'ús	53
<b>7</b>	<b>Conclusions i treballs futurs</b>	<b>57</b>
7.1	Conclusions	57
7.2	Relació amb els estudis	58
7.3	Treball futur	58
	<b>Bibliografia</b>	<b>61</b>

---

## Apèndixs

<b>A</b>	<b>Descripció de les UT</b>	<b>65</b>
A.1	<i>Sprint 1</i>	65
A.1.1	<i>Toolbar</i>	65
A.1.2	<i>Cambiar nombre del grupo</i>	65
A.1.3	<i>Añadir descripción al grupo</i>	66
A.1.4	<i>Indicador de estado del usuario</i>	66
A.2	<i>Sprint 2</i>	66
A.2.1	<i>Eliminar participantes del grupo</i>	66
A.2.2	<i>Eliminar grupo</i>	66
A.2.3	<i>Editar descripción del grupo</i>	67
A.3	<i>Sprint 3</i>	67
A.3.1	<i>Rediseño de la interfaz (más intuitiva)</i>	67

---

A.3.2	<i>Indicar autor del mensaje . . . . .</i>	67
A.3.3	<i>Diferenciar los mensajes del usuario de los otros . . . . .</i>	67
A.3.4	<i>Visualización de las conexiones actuales . . . . .</i>	68
A.3.5	<i>Desplegar la aplicación web . . . . .</i>	68
A.3.6	<i>Encuesta de usabilidad y análisis de resultados . . . . .</i>	68
<b>B</b>	<b>Sobre ODS</b>	<b>69</b>



# Índex de figures

---

2.1	Esquema DAFO de ConnectFG . . . . .	9
3.1	Arquitectura de ConnectFG . . . . .	13
3.2	Arquitectura del backend de ConnectFG. Extreta de [3] . . . . .	14
4.1	Arquitectura d'Angular. Extreta de [6] . . . . .	16
4.2	Exemple d'utilitat d'HTML i SCSS. Extreta de [11] . . . . .	16
5.1	Model de domini de ConnectFG . . . . .	20
5.2	Gràfica burndown en hores de l' <i>sprint</i> 0 . . . . .	21
5.3	<i>Mockup</i> de la UT 224 . . . . .	22
5.4	<i>Mockup</i> de la UT 255 . . . . .	23
5.5	<i>Mockup</i> de la UT 147 . . . . .	23
5.6	Esquema del patró observador . . . . .	26
5.7	Exemple dels esdeveniments al enviar un missatge a un grup . . . . .	27
5.8	Diagrama de flux acumulat de l' <i>sprint</i> 1 . . . . .	30
5.9	Treball finalitzat vs no finalitzat (en hores) de l' <i>sprint</i> 1 . . . . .	31
5.10	Gràfica de burndown en hores de l' <i>sprint</i> 1 . . . . .	32
5.11	<i>Mockup</i> de les UT 256 i 257 . . . . .	33
5.12	<i>Mockup</i> actualitzat de la gestió de candidats . . . . .	34
5.13	<i>Mockup</i> de les UT 144 i 145 . . . . .	36
5.14	Diagrama de flux acumulat de l' <i>sprint</i> 2 . . . . .	39
5.15	Gràfica burndown en hores de l' <i>sprint</i> 2 . . . . .	40
5.16	<i>Mockup</i> de la UT 239 . . . . .	41
5.17	<i>Mockup</i> de la UT 162 . . . . .	45
5.18	Arxius que contenen les traduccions . . . . .	45
5.19	Arxiu amb les traduccions al castellà . . . . .	46
5.20	Diagrama de flux acumulat de l' <i>sprint</i> 3 . . . . .	47
5.21	Gràfica burndown en hores de l' <i>sprint</i> 3 . . . . .	48
6.1	Consola de Firebase . . . . .	50
6.2	Variables d'entorn per al desenvolupament . . . . .	50
6.3	Variables d'entorn per a producció . . . . .	50
6.4	Fitxers de la carpeta dist al fer el build . . . . .	51

6.5	Dashboard de Render . . . . .	52
6.6	Variables d'entorn del backend . . . . .	53
6.7	Configuració de les variables d'entorn en Render . . . . .	53
6.8	Exemple d'una sol·licitud de connexió . . . . .	54
6.9	Finestra de connexions . . . . .	54
6.10	Exemple de creació d'un xat . . . . .	55
6.11	Exemple d'iniciar la conversa . . . . .	55



# Índex de taules

---

2.1	Anàlisi dels competidors de ConnectFG . . . . .	8
3.1	Llistat d'UTs en el backlog . . . . .	12
5.1	<i>Sprints</i> del desenvolupament . . . . .	19
5.2	Llistat d'UT de l' <i>sprint</i> 1 . . . . .	22
5.3	Proves d'acceptació del xat . . . . .	28
5.4	Llistat d'UTs de l' <i>sprint</i> 2 . . . . .	32
5.5	Proves d'acceptació realitzades en l' <i>sprint</i> 2 . . . . .	35
5.6	Proves d'acceptació del sistema de notifikacions . . . . .	39
5.7	Llistat d'UTs de l' <i>sprint</i> 3 . . . . .	41
5.8	Proves d'acceptació d'enviament d'arxius al xat . . . . .	44



---

---

# CAPÍTOL 1

## Introducció

---

En aquest capítol introduïrem el projecte ConneCTFG, centrant-nos en els nostres objectius i la motivació per a crear-lo, així com l'estructura que presentarà aquesta memòria.

### 1.1 Motivació

---

El desenvolupament de la plataforma ConneCTFG sorgeix com a resposta a la necessitat d'oferir als estudiants universitaris una ferramenta que millori la seua experiència durant l'etapa de realització del Treball de Fi de Grau (TFG). Els TFGs sovint representen una experiència solitària i desafiant per a molts estudiants. No obstant això, la participació en projectes multidisciplinaris pot oferir una oportunitat excepcional per experimentar un enfocament de treball real en un context acadèmic. Aquesta motivació es fonamenta en la nostra experiència en el projecte col·laboratiu de l'assignatura PIN (*Proyecto de Ingeniería de software*) del grau. En aquest projecte, vam poder reconèixer el gran potencial dels projectes multidisciplinaris i la escassa presència que tenen en els TFGs actuals, en part deguda a les dificultats de comunicació entre estudiants de diferents graus.

Durant el procés de desenvolupament d'aquest projecte col·laboratiu, hem comptat amb la col·laboració de Matías Leandro Fraga, també estudiant del Grau en Enginyeria Informàtica, on cadascú s'ha encarregat d'una part de l'aplicació. Matías ha exercit un paper fonamental en la creació i l'evolució de ConneCTFG, ocupant-se de la gestió de publicacions, la implementació de perfils editables per als usuaris i la creació d'un *feed* que mostra les propostes, entre altres tasques essencials.

A més de la seua participació en aquestes àrees, Matías també s'ha encarregat de documentar l'altra meitat de l'aplicació, particularment els aspectes relacio-

nats amb el sistema de xarxa social i la gestió dels usuaris. Es recomana consultar aquesta documentació per a obtenir una comprensió completa del procés de desenvolupament de ConnecTFG.

## 1.2 Objectius

---

L'objectiu principal d'aquesta aplicació és proporcionar una plataforma digital que permeti als alumnes comunicar-se i trobar els col·laboradors idonis per a fer el TFG, independentment del grau que cursen. A més, volem fomentar la formació d'equips multidisciplinaris i, per tant, promoure la diversitat de coneixements i habilitats que es comparteixen al treballar en aquest tipus d'equips.

Aquest objectiu principal es desglossa en diversos subobjectius:

1. Desenvolupar una interfície d'usuari atractiva e intuïtiva per a proporcionar als usuaris una experiència positiva i facilitar la seua utilització de l'aplicació.
2. Crear un sistema de missatgeria en temps real integrat en la plataforma, permetent als usuaris intercanviar informació sobre els seus interessos o coneixements sense haver de recórrer a una plataforma externa.
3. Proporcionar als usuaris un perfil detallat amb informació rellevant sobre ells mateixos, com ara el seu grau acadèmic, àrees d'interès o habilitats en les que destaquen. Aquesta informació facilitarà que els estudiants puguin filtrar eficientment els usuaris amb els quals els agradaria col·laborar.
4. Millorar constantment la tecnologia utilitzada en l'aplicació per a garantir un rendiment òptim i una experiència d'usuari sense problemes.
5. Implementar una metodologia àgil Kanban per a la gestió de projectes i el desenvolupament de l'aplicació. Aquesta metodologia permetrà una planificació flexible, una millora contínua i una major adaptació als canvis necessaris durant el desenvolupament.

Aquests subobjectius han estat formulats per aconseguir l'objectiu principal, que consisteix en millorar l'experiència dels estudiants universitaris i fomentar la col·laboració mitjançant la plataforma ConnecTFG.

---

## 1.3 Estructura de la memòria

---

Aquest document està estructurat en 7 capítols, explorant en cadascun d'ells un aspecte crucial en el procés de desenvolupament d'aquest projecte.

En el segon capítol, es descriu ConnecTFG en l'àmbit de negoci i s'analitza la seua viabilitat en el mercat actual.

En el tercer capítol es descriuen les funcionalitats clau de ConnecTFG i l'arquitectura que s'ha seguit en el desenvolupament.

En el quart capítol s'analitzen en profunditat les tecnologies utilitzades en el desenvolupament, tant del *frontend* com del *backend*, així com altres ferramentes emprades.

En el cinqué capítol es detalla el procés de desenvolupament de ConnecTFG dividit en 4 *sprints*. S'analitza cada *sprint* individualment, descrivint les tasques realitzades i els reptes superats.

En el sisé capítol es detalla el procés de desplegament de la plataforma ConnecTFG, destacant els passos per fer la plataforma accessible als usuaris.

En el setè i últim capítol es conclou el treball, establint una relació entre el projecte i els objectius aconseguits, així com descriuint les funcionalitats que es desenvoluparan en un futur.

A més, apareixen les cites bibliogràfiques emprades durant l'elaboració del TFG, així com un parell d'annexes.

En l'annex A es descriuen en detall totes les UT desenvolupades al llarg dels 4 *sprints*.

Finalment, en l'annex B es relaciona el TFG amb els Objectius de Desenvolupament Sostenible.



---

---

## CAPÍTOL 2

# Estudi del mercat

---

En aquest capítol ens centrarem en descriure l'aplicació ConnectTFG en l'àmbit de negoci, explicant primerament la seua idea de negoci, analitzant les seues debilitats, amenaces, fortaleces i oportunitats en el mercat i, per finalitzar, la compararem amb els seus competidors principals

### 2.1 Idea de negoci

---

La idea de negoci de la nostra aplicació és la de proporcionar una plataforma que facilite la connexió entre estudiants universitaris interessats en treballar en projectes de final de grau multidisciplinaris.

En aquesta aplicació web els usuaris podran registrar-se i crear un perfil, amb la possibilitat d'afegir informació rellevant, com ara el seus interessos o habilitats, si així ho desitgen. A més, una vegada els estudiants ja han establert connexions entre ells, la plataforma proporciona un sistema de missatgeria i notificacions en temps real. Açò permet als usuaris comunicar-se d'una forma fluida i efectiva, compartint idees o resolvent dubtes del projecte, entre altres. A banda, el sistema de notificacions manté als estudiants informats de les actualitzacions en les seues propostes de TFG o en les seues connexions en temps real, permetent una interacció més ràpida.

Respecte a la forma de generar ingressos de l'aplicació, s'ha optat per un enfocament basat en la publicitat i en limitar el nombre de connexions que pots realitzar si no eres usuari *premium*. A més, s'obriria la possibilitat a l'usuari de poder pagar per promocionar les seues publicacions i que apareguen en les posicions inicials del *feed*, per tal d'obtindre una major interacció.

---

## 2.2 Competidors

---

En el mercat actual, existeixen diverses plataformes i recursos que ofereixen serveis similars, sobretot pel que fa a la part de connectar amb estudiants. A continuació, es presenta un anàlisi dels principals competidors de ConneCTFG:

- LinkedIn: és una xarxa social que connecta a professionals i estudiants universitaris de diferents àmbits. Encara que el seu enfocament és principalment l'àmbit laboral, també permet que els usuaris notifiquen la seua cerca de companys de treball per mitjà d'una publicació en la plataforma, i que al seu torn, altres estudiants busquen connectar-se amb ell per treballar de forma conjunta. A més, es poden observar els perfils dels usuaris, pel que, en aquest aspecte, és una aplicació bastant similar a ConneCTFG.
- Facebook: és una xarxa social molt coneguda, que també compta amb un sistema integrat de publicacions. Així doncs, aquest sistema pot emprarse per compartir o anunciar que estàs buscant un company per realitzar el TFG i, d'aquesta manera, tractar de contactar amb ell. A més, aquesta plataforma té integrada les trucades i videotrucades, pel que resulta més senzill parlar i debatre sobre els seus objectius i avaluar la seua compatibilitat ràpidament. D'altra banda, no és una plataforma centrada específicament en l'àmbit educatiu, pel que, encara que compta amb un gran nombre de perfils, aquests tenen un enfocament més social o personal que acadèmic, fent que Facebook no siga una aplicació idíllica per a la cerca d'un company de TFG, tot i que, en principi, les seues funcionalitats permeten fer-ho.
- PEIX (Pràctiques en Empresa per a Informàtics a la Xarxa): és una plataforma de la Universitat Politècnica de València (UPV) per a ajudar als alumnes informàtics en la cerca d'empreses on fer les pràctiques. Aquesta permet crear un perfil d'estudiant, on pots fer el teu currículum virtual i donar-te a conèixer. Cal tenir en compte que aquesta plataforma no permet als estudiants fer publicacions ni interactuar entre ells. No obstant això, si aquesta restricció fora eliminada, al tractar-se d'una aplicació que conté principalment perfils acadèmics, sí que podria arribar a ser una competència directa de ConneCTFG.

Després d'haver analitzar profundament els seus principals competidors, s'han identificat funcionalitats amb les que ens podem diferenciar, com podem observar en la taula 2.1.



<b>Funcionalitat</b>	<b>Facebook</b>	<b>LinkedIn</b>	<b>PEIX</b>	<b>ConnecTFG</b>
<i>Registrarse</i>	✓	✓	✓	✓
<i>Iniciar sesión</i>	✓	✓	✓	✓
<i>Visualizar feed</i>	✓	✓	✓	✓
<i>Visualizar perfil</i>	✓	✓	✓	✓
<i>Añadir post</i>	✓	✓		✓
<i>Editar post</i>	✓	✓		✓
<i>Eliminar post</i>	✓	✓		✓
<i>Añadir descripción del usuario</i>	✓	✓	✓	✓
<i>Editar descripción del usuario</i>	✓	✓	✓	✓
<i>Buscador de TFGs</i>		✓		✓
<i>Postularse como candidato</i>		✓	✓	✓
<i>Eliminarse como candidato</i>		✓	✓	✓
<i>Aceptar candidato</i>				✓
<i>Rechazar candidato</i>				✓
<i>Enviar mensaje</i>	✓	✓		✓
<i>Recibir mensaje</i>	✓	✓		✓
<i>Recibir notificaciones</i>	✓	✓		✓
<i>Eliminar notificación</i>	✓	✓		✓
<i>Dar me gusta a una publicación</i>	✓	✓		✓
<i>Internacionalización</i>	✓	✓		✓
<i>Añadir comentario al post</i>	✓	✓		✓
<i>Feed de publicaciones cerradas</i>				✓
<i>Cerrar post</i>				✓
<i>Añadir TFG al perfil</i>				✓
<i>Enviar archivos al chat</i>	✓	✓		✓
<i>Descargar archivos del chat</i>	✓	✓		✓
<i>Cerrar sesión</i>	✓	✓	✓	✓
<i>Crear grupo</i>	✓	✓		✓
<i>Cambiar nombre del grupo</i>	✓	✓		✓
<i>Eliminar grupo</i>	✓	✓		✓
<i>Añadir participantes al grupo</i>	✓	✓		✓
<i>Eliminar participantes del grupo</i>	✓	✓		✓
<i>Añadir descripción al grupo</i>	✓	✓		✓
<i>Editar descripción del grupo</i>	✓	✓		✓
<i>Editar publicación</i>	✓	✓		✓
<i>Buscador de usuarios</i>	✓	✓		✓

Indicador de estado del usuario	✓	✓		✓
---------------------------------	---	---	--	---

**Taula 2.1:** Anàlisi dels competidors de ConnecTFG

Després d'estudiar la taula 2.1, s'ha conclòs que hi ha un conjunt de funcionalitats que es podrien considerar essencials per tal de diferenciar-nos de la competència. Aquestes funcionalitats son totes les relacionades amb l'especialització de ConnecTFG en referència als TFGs, com poden ser un buscador de TFGs o acceptar/rebutjar candidats segons t'interesse, el que ens representaria com a la única aplicació enfocada en l'àmbit acadèmic per a la connexió d'estudiants de diferents graus amb la finalitat de realitzar el TFG junts.

Com a conclusió, es pot observar que tot i que la nostra aplicació comparteix alguns aspectes com el sistema de publicacions o la missatgeria amb els seus competidors, trobem una clara diferenciació en el seu enfocament purament acadèmic.

## 2.3 Anàlisi DAFO

L'anàlisi DAFO (Debilitats, Amenaces, Fortaleses i Oportunitats) [1] és una ferramenta d'anàlisi estratègic molt útil per tal d'avaluar la situació actual d'un projecte en termes de factors interns i externs. A continuació, es presenta l'anàlisi DAFO de ConnecTFG, el qual també es pot veure esquematitzat en la figura 2.1.

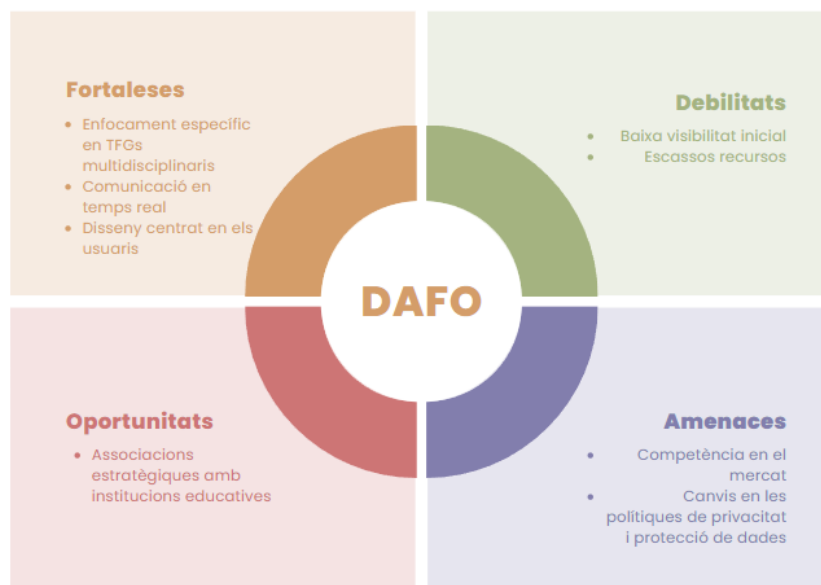
- Debilitats

- Baixa visibilitat inicial: Al tractar-se d'una plataforma nova, es pot enfrontar a dificultats en la seua etapa inicial per a donar-se a conèixer entre els estudiants universitaris i competir amb altres xarxes socials existents.
- Escassos recursos: Al ser una empresa emergent, no es compta amb una gran inversió de capital. Açò implica que no es tinguen els recursos necessaris per a fer una gran campanya de màrqueting inicial per a donar-se a conèixer.

- Amenaces

- Competència en el mercat: Existeixen altres plataformes disponibles que poden servir per trobar companys de TFG, com ara: LinkedIn, Facebook, ... pel que alguns estudiants poden decidir-se per aquestes al tindre una major trajectòria.

- Canvis en les polítiques de privacitat i protecció de dades: Un canvi en les normatives podria afectar la forma en la que ConnectTFG gestiona i utilitza la informació de les dades dels usuaris, el que pot ocasionar un augment de desconfiança per part dels estudiants.
- Fortaleses
    - Enfocament específic en TFGs multidisciplinaris: La nostra aplicació està enfocada a connectar estudiants de diferents graus amb una idea comú per un projecte. Aquest fet ens diferencia d'altres xarxes socials que, en general, no solen estar enfocades en l'àmbit acadèmic.
    - Comunicació en temps real: L'enviament de missatges i notificacions instantànies permet resoldre dubtes, compartir informació i prendre decisions de manera immediata, millorant l'experiència d'usuari i fomentant la col·laboració entre l'estudiantat.
    - Disseny centrat en els usuaris: L'aplicació està dissenyada per a ser fàcil d'utilitzar i ajustant-se a les necessitats reals dels estudiants en el procés de selecció de companys.
  - Oportunitats
    - Associacions estratègiques amb institucions educatives: ConnectTFG és una aplicació capacitada per col·laborar amb universitats i facultats, oferint serveis exclusius als seus estudiants. Un exemple podria ser un feed només per als estudiants de la mateixa facultat, facilitant la visibilitat en eixa organització i, traduint-se al cap i a la fi en un augment del nombre d'usuaris.



**Figura 2.1:** Esquema DAFO de ConnectTFG



---

## CAPÍTOL 3

# ConnecTFG

---

Aquest capítol està centrat a presentar d'una forma general l'aplicació ConnecTFG, descrivint primerament les funcionalitats desenvolupades i les que s'han quedat en el *backlog*, així com analitzant l'arquitectura de *software* que s'ha utilitzat,

### 3.1 Funcionalitats

---

Les funcionalitats de l'aplicació ConnecTFG es divideixen en cinc grups ben diferenciats, que alhora compten amb funcionalitats més específiques. Aquests grups són: gestió d'usuaris, gestió de publicacions, gestió de notifikacions, gestió de candidats i gestió de reunions.

En aquests grups, cal destacar els tres últims, que són els que conformen el sistema de missatgeria i notifikacions. Entre les funcionalitats que s'inclouen en aquests grups trobem el xat, necessari per a que els estudiants es comuniquen sense necessitar cap aplicació externa, acceptar/rebutjar les sol·licituds de candidats, per tal de triar les persones amb les que t'interessa col·laborar, i rebre notifikacions, les quals mantenen als usuaris informats en temps real.

### 3.2 *Backlog*

---

El *backlog* és el conjunt de funcionalitats que s'han quedat per fer, ordenades per importància, i que s'haurien d'implementar en els futurs *sprints*. En el cas de ConnecTFG són les que es poden observar en la taula 3.1.

Ordre	Identificador	Nom
15	242	<i>Filtro por grado</i>
23	228	<i>Filtro por universidad</i>

45	208	<i>Añadir rol a los usuarios</i>
50	218	<i>Compartir perfil</i>
55	204	<i>Añadir universidades (Admin)</i>
60	205	<i>Editar universidades (Admin)</i>
65	207	<i>Eliminar universidades (Admin)</i>
70	233	<i>Indicar cuando el usuario está escribiendo</i>
75	206	<i>Iniciar llamada</i>
80	161	<i>Dark / Light Mode</i>
82	211	<i>Iniciar videollamada</i>
84	210	<i>Unirse a la llamada</i>
85	212	<i>Terminar la videollamada</i>
90	229	<i>Terminar la llamada</i>
95	230	<i>Unirse a la videollamada</i>
100	213	<i>Activar audio en una llamada / videollamada</i>
105	234	<i>Activar webcam en videollamada</i>
110	214	<i>Desactivar webcam en videollamada</i>
115	232	<i>Silenciar audio en una llamada / videollamada</i>
120	215	<i>Grabación de vídeo en las videollamadas</i>
125	235	<i>Grabación de audio en las llamadas</i>
130	316	<i>Compartir escritorio en las llamadas</i>
131	236	<i>Grabación de audio y vídeo en las videollamadas</i>
135	231	<i>Enviar emojis</i>
143	220	<i>Descargar archivos del chat</i>
145	240	<i>Enviar audios en el chat</i>
150	221	<i>Reproducir audios en el chat</i>

**Taula 3.1:** Llistat d'UTs en el backlog

### 3.3 Arquitectura

L'arquitectura client-servidor [2] és un dels estils arquitectònics distribuïts més coneguts, on existeix un servidor i múltiples clients que es connecten al servidor per recuperar tots els recursos necessaris per funcionar. En aquest cas, el client, que és l'usuari final, interacciona amb el *frontend*, i el *backend* es troba en el servidor.

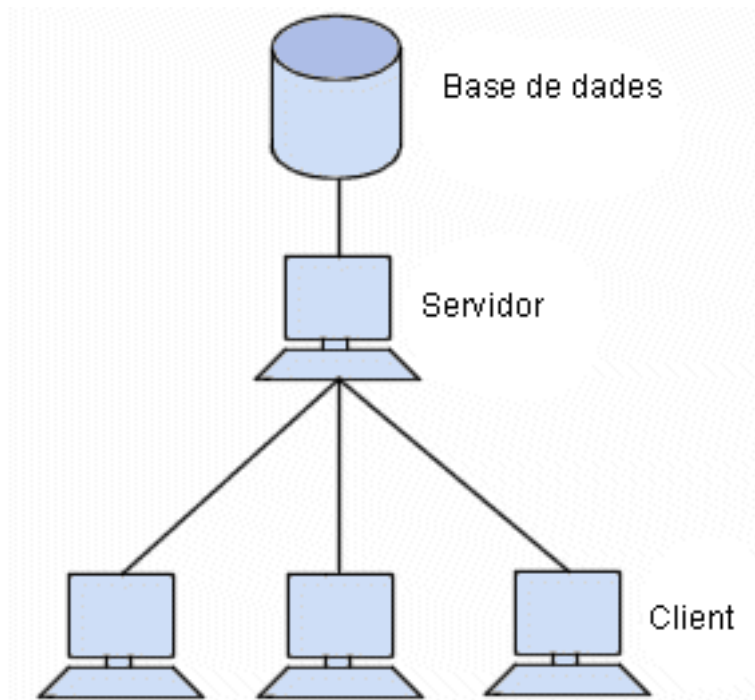


Figura 3.1: Arquitectura de ConnecTFG

A continuació, es descriurà breument l'arquitectura de l'aplicació, tal com mostra la figura 3.1:

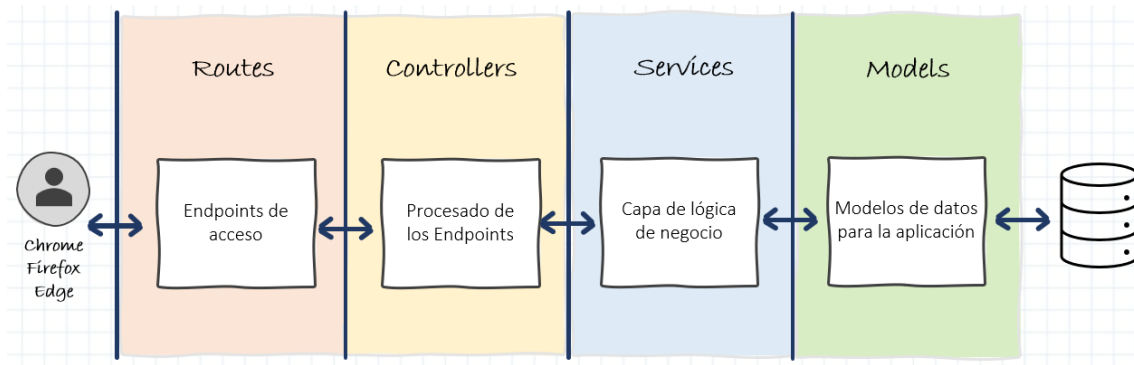
- Client (*Frontend*):

El *frontend* és la capa visual de l'aplicació, amb la qual interacciona l'usuari, i està dissenyada per a navegadors en ordinador.

- Servidor (*Backend*):

Esta part és l'encarregada de processar les sol·licituds del client i realitzar operacions en la base de dades. A més, s'han seguit les pautes de l'arquitectura en capes per desenvolupar-lo, diferenciant aquestes 4: rutes, controladors, serveis i models [3], com es pot observar en la figura 3.2.

- Capa de rutes: secció de codi express que associarà una petició http/https amb el controlador pertinent per gestionar la petició.
- Capa de controladors: reben les sol·licituds de les rutes, es comuniquen amb la capa de serveis i converteixen les sol·licituds en respostes https.
- Capa de serveis: inclou tota la lògica del *backend*. Per dur a terme les operacions realitza cridades tant a classes dins d'aquesta capa com a classes de la capa inferior (model).
- Capa de models: representa els models de dades de la nostra aplicació.



**Figura 3.2:** Arquitectura del backend de ConnetTFG. Extreta de [3]

En resum, l'arquitectura de l'aplicació es basa en una arquitectura client-servidor, on el client s'encarrega de la presentació e interacció amb l'usuari i el servidor, de la lògica de negoci, la gestió de dades i dels esdeveniments en temps real. A més, s'ha utilitzat Angular com a *framework* del *frontend* i Node.js, Express.js, MongoDB i Socker.io per al *backend*, les quals s'explicaràn amb més detall en el següent capítol.



---

---

# CAPÍTOL 4

## Tecnologia

---

En aquesta secció ens centrarem en els llenguatges, *frameworks*, llibreries i aplicacions emprades per al desenvolupament de l'aplicació, els quals, com ja s'ha comentat en l'apartat anterior, són: Angular, Node.js, Express.js, MongoDB i Socket.io, amb l'ajuda d'aplicacions com: Worki o Moqups, entre altres.

### 4.1 *Frontend*

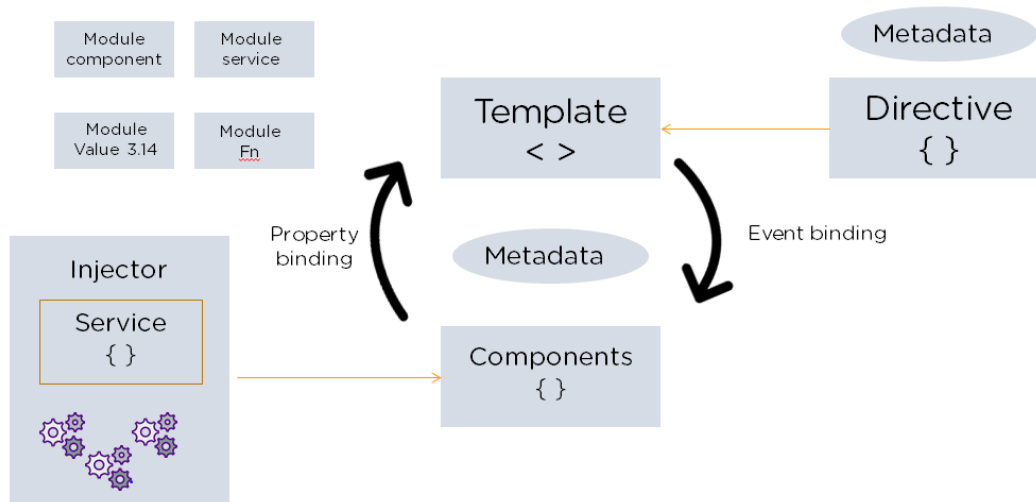
---

En aquesta secció, es detallaran les tecnologies clau emprades al *frontend* de l'aplicació, és a dir, la part de l'aplicació amb la qual els usuaris interactuen directament. Aquestes tecnologies són Angular, HTML i SCSS.

#### 4.1.1. Angular

Angular [4] és un framework de JavaScript de codi obert, escrit en Typescript, mantés per Google que té com a objectiu principal el desenvolupament d'aplicacions web SPA (Single Page Application), és a dir, aplicacions d'una sola pàgina, que destaquen per la seua rapidesa i velocitat al només necessitar carregar un únic arxiu, millorant l'experiència d'usuari. A més, aquestes SPA servixen tant per a versions mòbils com d'escriptori, sent dinàmiques i escalables, gràcies en part a les directives d'angular, que permeten ampliar el comportament del codi HTML amb accions com mostrar un element només si es compleix una condició (\*ngIf) o mostrar-lo un nombre variable de vegades en funció dels elements d'un *array* (\*ngFor), entre altres. Tanmateix, l'escalabilitat es podria atribuir a la capacitat de dividir l'aplicació en mòduls, components o serveis; al tindre el codi fragmentat, es facilita la tasca de crear noves funcionalitats així com de mantindre el codi existent. Entre els principals avantatges que té, cal destacar-ne dos més, l'enllaç de les dades entre l'HTML i Typescript (*Data Binding*), tant unidireccional

com bidireccional, i la capacitat d'automatitzar proves emprant Jasmine i Karma, dos ferramentes importants en el context del *testing* que estan integrades en el *framework* [5], [6], [7].



**Figura 4.1:** Arquitectura d'Angular. Extreta de [6]

#### 4.1.2. HTML i SCSS

D'una banda, HTML, o *HyperText Markup Language*, [8], [9] és un llenguatge de marcatge on es declara l'estructura dels elements de la interfície de l'aplicació, com ara: imatges, llistes, paràgrafs, etc.

D'altra banda, SCSS [10] és el llenguatge que s'ha utilitzat per a definir el disseny i presentació visual de l'aplicació, aquest consta amb la sintaxi de CSS però amb tots els avantatges de SASS, com poden ser la utilització de variables, nidificació, herència dels selectors, etc.

En resum, HTML declara l'estructura de la pàgina web i SCSS li agrega l'estil, com es pot observar en la figura 4.2.



**Figura 4.2:** Exemple d'utilitat d'HTML i SCSS. Extreta de [11]

---

## 4.2 Backend

---

En aquesta secció, es detallaran les tecnologies clau emprades al *backend* de l'aplicació, és a dir, la part de l'aplicació que gestiona la lògica, el processament de dades i la comunicació amb la base de dades. Aquestes tecnologies són Node.js, Express, Socket.io i MongoDB.

### 4.2.1. Node.js

Node.js [12],[13] és un entorn en temps d'execució de JavaScript que s'executa en el costat del servidor i que destaca per l'enfocament en la programació asíncrona i sense bloqueig, a més de permetre una gran quantitat de connexions simultànies amb un alt nivell de rendiment, per el que té una gran rellevància en aplicacions en temps real i escalables.

### 4.2.2. Express

Express [14] és un *framework* d'aplicacions web per a Node.js que proporciona eines robustes per desenvolupar aplicacions de *backend* escalables, com ara: peticions i respostes HTTP, encaminament i *middleware*. Així doncs, és l'encarregat de comunicar-li les peticions HTTP als controladors del *backend* per tal de processar la petició i, una vegada processada, respondre-li al client en format JSON (en el cas de ConnectFG).

### 4.2.3. Socket.IO

Socket.io [15],[16] és una llibreria de JavaScript de codi obert per a Node.js que permet una comunicació TCP bidireccional en temps real entre clients i servidor, el que fa possible la creació de web interactives en temps real. En el nostre cas, ha sigut utilitzada per al correcte funcionament del xat i de les notificacions.

### 4.2.4. MongoDB

MongoDB [17] és una base de dades NoSQL (no relacional) orientada a documents, que s'utilitza per a emmagatzemar dades de forma flexible i escalable. A diferència d'una base de dades relacional, MongoDB no es basa en taules i columnes, sinó que les dades s'emmagatzemen com a col·leccions i documents, els quals són parells clau/valor i serien l'equivalent a un registre en una base de da-

des tradicional. A més, és una base de dades molt ràpida, fet que s'agraeix en les xarxes socials, on hi ha un gran moviment de dades.

## 4.3 Ferramentes externes

---

En aquesta secció es detallen algunes ferramentes que han resultat essencials en el desenvolupament de l'aplicació, com son Git, GitHub i Worki.

### 4.3.1. Git

Git és un sistema de control de versions distribuït, que emmagatzema els canvis realitzats en el codi en un repositori local, el qual permet treballar sense connexió o de forma remota amb facilitat. Els desenvolupadors confirmen el seu treball localment i, a continuació, sincronitzen la seua còpia del repositori amb la còpia del servidor. En aquesta còpia del servidor és la que ens permet treballar amb equip i estar sempre al dia de les actualitzacions dels teus companys.

### 4.3.2. GitHub

Github [18] és un portal per a gestionar les aplicacions que utilitzen el sistema Git, on es pot allotjar remotament el teu codi i té integrades les principals funcions de Git per al control de versions, com ara: comparar les diferències entre versions, restaurar versions antigues si alguna cosa ix mal, treballar amb distintes rames, etc. A més, permet que altres persones puguen mirar i descarregar-se les diferents versions d'una aplicació i, fins i tot, col·laborar en conjunt.

### 4.3.3. Worki

Worki [19] és una plataforma destinada a la gestió de projectes àgils, on es pot gestionar en equip, i que permet als usuaris organitzar tasques, assignar responsables, registrar temps per a mantenir un seguiment correcte del projecte i, el més important, proporciona uns gràfics que permeten determinar en quin punt es troba el projecte en qualsevol moment.

---

# CAPÍTOL 5

## Desenvolupament

---

En aquest capítol, es proporciona una visió integral de la metodologia emprada durant el procés de desenvolupament de ConnecTFG, així com una descripció en profunditat dels quatre *sprints* en els quals s'ha estructurat aquest treball.

### 5.1 Organització en *sprints*

---

Pel desenvolupament d'aquest projecte hem optat per seguir la metodologia àgil Kanban, amb l'ajuda de Worki, que ens permet realitzar el seguiment del temps dedicat en cada UT (Unitat de Treball), a més de proporcionar-nos les gràfiques i estadístiques necessàries per a mantindre una evolució correcta de l'aplicació. Per tant, durant aquest temps de desenvolupament hem estat seguint una estratègia MVP (*Most Valuable Product*), pel que hem anat desenvolupant ConnecTFG per *sprints*, d'una duració d'un mes cadascun, anant afegint la funcionalitat més important en el seu moment però sempre tenint una aplicació funcional.

Aquests *sprints* es poden observar en la taula 5.1, amb la data de quan es va iniciar i quan es va finalitzar.

<i>Sprint</i>	Data Inicial	Data Final
<i>Sprint 0</i>	08/02/2023	08/03/2023
<i>Sprint 1</i>	09/03/2023	05/04/2023
<i>Sprint 2</i>	06/04/2023	03/05/2023
<i>Sprint 3</i>	12/05/2023	09/06/2023

**Taula 5.1:** *Sprints* del desenvolupament

A continuació, estudiarem en detall els avanços que es van dur a terme en cada *sprint*, així com una avaluació individual dels resultats proporcionats.

## 5.2 Sprint 0

Aquest *sprint*, que actua com a fase preparatòria del projecte, es va dedicar a analitzar la viabilitat de ConnectFG, estudiant punts importants en l'inici d'un projecte, com són: l'anàlisi DAFO i l'anàlisi de competidors. D'altra banda, una vegada estudiat el mercat i enfocant-se en les oportunitats que té el projecte, es va procedir a definir funcionalitats, prioritzar característiques, crear el model de domini, dissenyar la base de dades i definir un diagrama de flux de l'aplicació. Per finalitzar l'*sprint*, es van definir les tecnologies a utilitzar, es van crear els repositoris en GitHub i es va preparar l'entorn de desenvolupament (Visual Studio Code) per poder començar a treballar.

### 5.2.1. Model de domini

El model de domini és una representació conceptual d'un sistema o àrea de coneixement específica que descriu els elements principals, les seues interaccions i les restriccions que existeixen en eixe domini en particular. L'objectiu d'un model de domini [20] és capturar i organitzar el coneixement sobre el domini en un format comprensible i estructurat, permetent a l'equip de desenvolupament tindre una visió clara i comú del problema a resoldre, així com estandarditzar els noms per a referir-se als diferents objectes, com per exemple que tots anomenen "Usuari" a l'usuari, i no l'anomenen "Client", "Estudiant" o d'altres.

Per tal d'aconseguir que siga comprensible per tothom, es va inventar el llenguatge UML (Llenguatge Unificat de Modelatge), creat per tal de definir un llenguatge de modelatge visual comú i capaç d'explicar tant la part sintàctica com semàntica de l'arquitectura i el comportament del sistema.

En la figura 5.1 es pot observar el model de domini de ConnectFG però en aquest document només ens centrarem en els objectes 'User', 'Notification' i 'Group'.

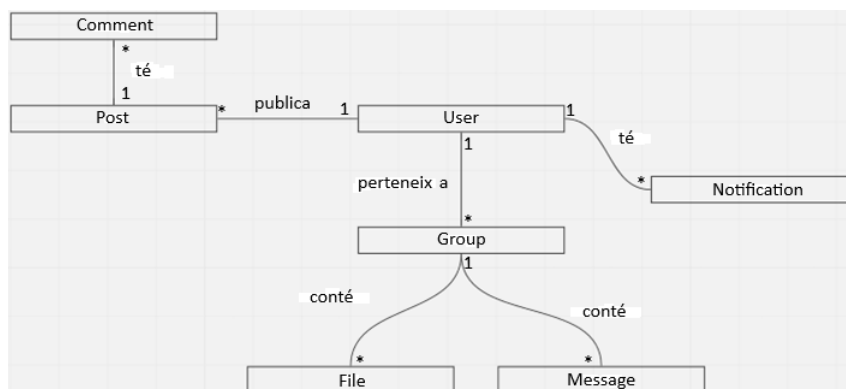


Figura 5.1: Model de domini de ConnectFG

### 5.2.2. Avaluació de l'*Sprint*

En l'àmbit del desenvolupament del software, l'avaluació i anàlisi dels resultats obtinguts en cada *sprint* són fonamentals per a mesurar el progrés de l'equip i millorar contínuament. Així doncs, anem a avaluar la gràfica de *burndown* en hores obtinguda en Worki que s'observa en la figura 5.2.

Aquesta gràfica ens permet observar l'evolució real del projecte en hores, tenint en compte la duració total estimada (línia verda), les hores registrades (línia blava) i les hores restants (línia roja), permetent mantindre un correcte seguiment del progrés del projecte e identificar possibles desviacions o retards en el desenvolupament, gràcies a la línia de referència (línia morada).

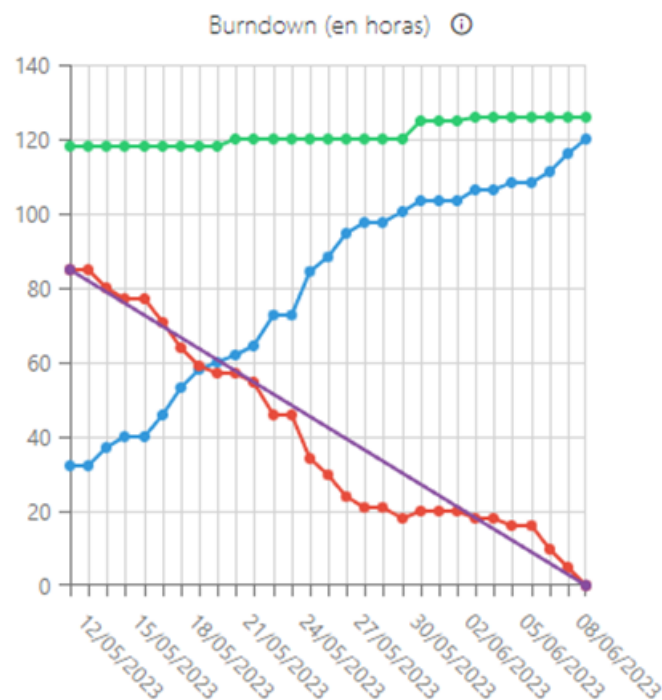


Figura 5.2: Gràfica burndown en hores de l'*Sprint* 0

Així doncs, podem concloure que l'equip ha mantingut un ritme constant de treball durant aquest *sprint*, estant la gran part del temps al nivell de la línia de referència, el que significa que l'estimació de les UT va ser precisa.

## 5.3 *Sprint* 1

En aquest *sprint* és on va començar el desenvolupament de l'aplicació en sí, centrant-nos en una de les funcionalitats principals de ConnecTFG, el sistema de missatgeria, acompanyada d'un *toolbar* (barra de navegació) per poder navegar per l'a-

plicació, així com un indicador de l'estat de l'usuari. El conjunt ordenat d'UTs realitzades durant aquest *sprint* es poden observar en la taula 5.2.

Ordre	Identificador d'UT	Nom	Estimació (en h)
5	316	<i>Toolbar</i>	3
15	255	<i>Crear grupo</i>	7
25	224	<i>Añadir participantes al grupo</i>	7
35	147	<i>Enviar mensaje</i>	12
45	146	<i>Recibir mensaje</i>	6
55	223	<i>Cambiar nombre del grupo</i>	3
65	225	<i>Añadir descripción al grupo</i>	3
75	241	<i>Indicador del estado del usuario</i>	10

Taula 5.2: Llistat d'UT de l'*sprint* 1

### 5.3.1. Xat

Amb l'objectiu d'implementar aquesta funcionalitat, es van generar diverses UT més petites, cadascuna encarregada d'implementar una part específica, amb la intenció d'avançar progressivament en l'assoliment d'aquesta funcionalitat.

Primerament, necessitem un grup amb almenys 2 integrants per tal de poder xatejar. És per això que les UT 255 i 224 es van realitzar en una etapa primerenca de l'*sprint*.



Figura 5.3: Mockup de la UT 224



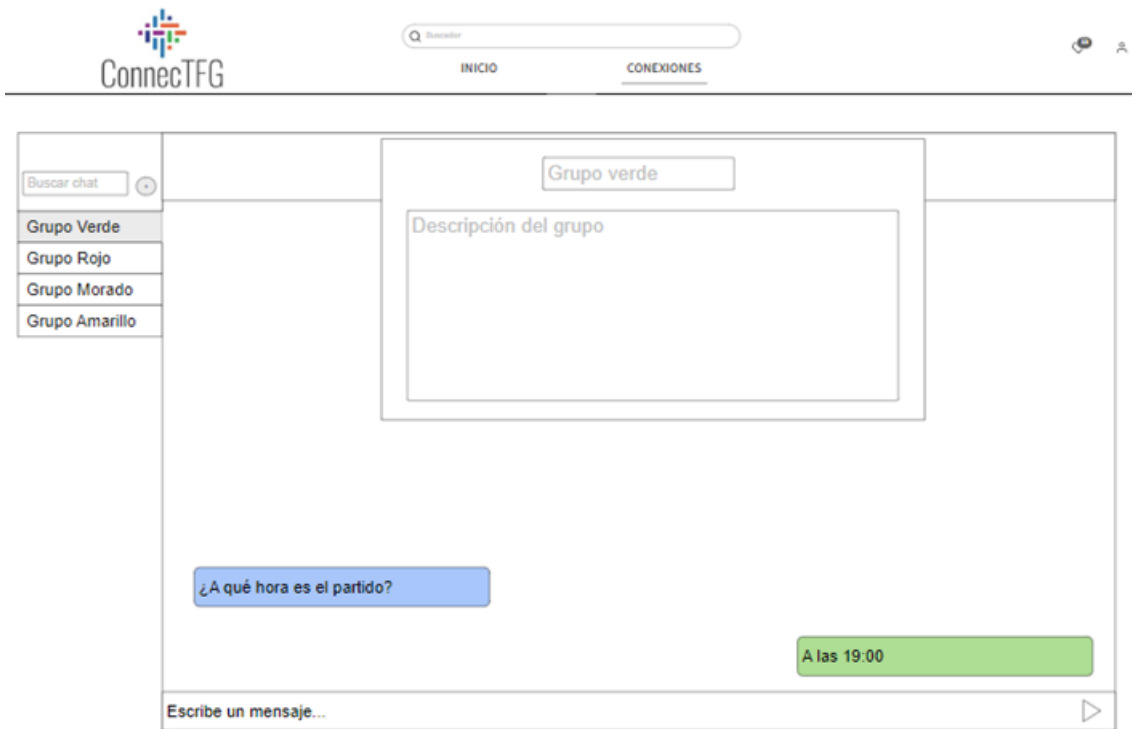


Figura 5.4: Mockup de la UT 255

En les figures 5.3 i 5.4 es pot observar el disseny inicial amb el que comptava la finestra del xat, però que va ser redissenyada per tal d'integrar ambdós funcionalitats en un mateix diàleg d'edició, per tal de no saturar a l'usuari amb moltes finestres per realitzar tasques.

Una vegada creat un grup i havent-li afegit un nombre d'usuaris suficient, es va procedir a desenvolupar la funcionalitat d'enviar i rebre missatges, amb l'ajuda del *mockup* que es pot veure en la figura 5.5.



Figura 5.5: Mockup de la UT 147

Ací és on van començar a sorgir més problemes; ja que, encara que havíem treballat amb Socket.io prèviament (i per això s'havia elegit per aquesta tasca), no l'havia enfocat a les converses en temps real, desencadenant una sèrie de problemes, entre els quals cal destacar la sincronització de missatges, és a dir, que l'usuari poguera veure els missatges que estava enviant sense haver de recarregar la pàgina.

Així doncs, per abordar aquest problema, es va plantejar la utilització d'un patró observador, per tal que l'usuari a l'entrar a un grup es subscriba als missatges d'aquest, i tant quan envie com quan reba missatges, s'introduïsquen dins de l'*array* de missatges que es visualitza en el xat.

El patró observador [21] és un patró de disseny de comportament que et permet definir un mecanisme de subscripció per notificar a diversos objectes sobre qualsevol esdeveniment que li succeeix a l'objecte que estan observant. Per tal d'aconseguir-ho en Typescript s'ha utilitzat 'Subject' de la llibreria 'rxjs', i els passos per a reproduir-ho serien els següents:

1. Configurar el servidor de Socket.io en un servei del *frontend* (SocketService en aquest cas).

```
1 setupSocketConnection() {
2   this.socket = io(this.SOCKET_ENDPOINT, {
3     query: {
4       userId: this.loginService.getUserId()
5     }
6   })
7
8   ...
9   this.socket.on('message', (message: any) => {
10    console.log('Nuevo mensaje recibido del servidor: ',
11              message);
12    this.messagesSubject.next(message);
13  });
14
15  ...
16 }
```

2. Configurar el servidor de Socket.io en el backend.

```
1 io.on('connection', (socket) => {
2   ...
3   socket.on('join', (group_id) => {
4     socket.join(group_id);
5   });
6 }
```

```

7   socket.on('newMessage', (data) => {
8       console.log('Nuevo mensaje recibido: ', data);
9       io.to(data.groupId).emit('message', data.message);
10  });
11  ...
12  })

```

3. En el servei SocketService, declarem una variable "Subjecte".

```

1   messagesSubject = new Subject<Message>();

```

4. Definim un mètode que retorne el subjecte com a observable.

```

1   getMessages(): Observable<Message> {
2       return this.messagesSubject.asObservable();
3   }

```

5. En el component del xat, injectem el servei que s'ha comentat en els punts 1 i 2.

```

1   constructor(private socketService: SocketService, ...) {}

```

6. En el ngOnInit d'aquest mateix component ens subscriuim a aquest observable.

```

1   this.socketService.getMessages().subscribe(message => {
2       this.messages.push(message);
3       this.loading = false;
4   },
5   err => {
6       this.loading = false;
7   }
8   )

```

En aquest moment, el patró ja està configurat per tal que, davant qualsevol canvi en l'estat de la variable observada 'messageSubject', és a dir, quan hi haja un missatge nou, aquest serà inserit en l'array 'messages', que és el que s'estarà visualitzant en el xat.

D'altra banda, falta notificar-li els missatges nous que s'envien al subjecte. Per tal d'aconseguir-ho s'hauran de seguir els passos següents:

7. Definir un mètode en el servei per a enviar els missatges al *websocket*.

```

1   sendMessage(groupId: number, message: Message) {
2       this.socket.emit('newMessage', { groupId, message });
3   }

```

El mètode 'emit' és l'encarregat d'enviar la tupla groupId, message al backend, qui, com podem observar en la configuració del pas 2, enviarà aquest missatge a tots els usuaris que estiguen connectats i units al grup.

A més, com podem observar en el pas 1, quan un usuari rep un missatge des del backend, s'executa el mètode 'next' sobre 'messagesSubject, el que fa que tots els usuaris subscrits a aquest subjecte siguin notificats de que hi ha un missatge nou i actualitzen el seu xat.

8. Per finalitzar, ja només caldria definir un mètode en el component per tal de cridar al mètode sendMessage del servei al prémer el botó d'enviar un missatge.

```

1  sendMessage () {
2  ...
3  if ( this.selectedGroup.id != null ) {
4  ...
5      this.socketService.sendMessage( this.selectedGroup.id ,
6          message );
7  }
8  ...
9  }

```

Amb açò, tindriem implementat un patró observador dins de ConnectFG, com es pot observar en les figures 5.6 i 5.7, on els 'Subscriber' són els usuaris que pertanyen a algun grup (els usuaris estan subscrits a tots els grups dels que forma part) i el 'Publisher' seria el servidor de Socket.io, el qual notifica als subscriptors sempre que rep un missatge nou.

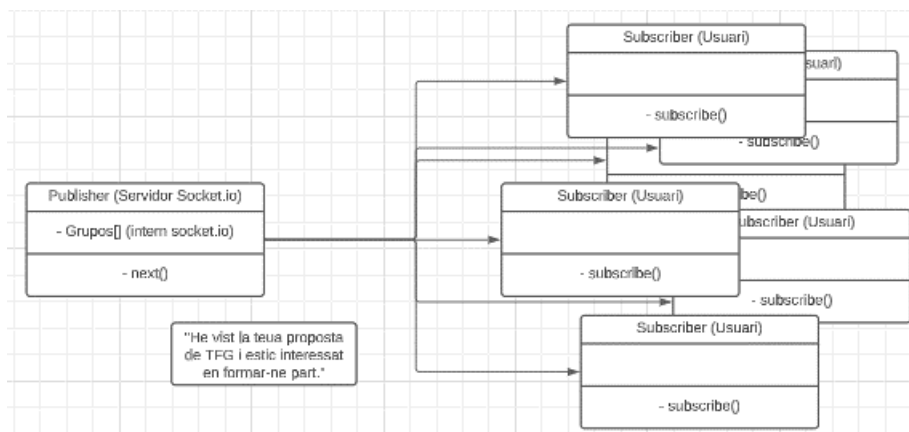
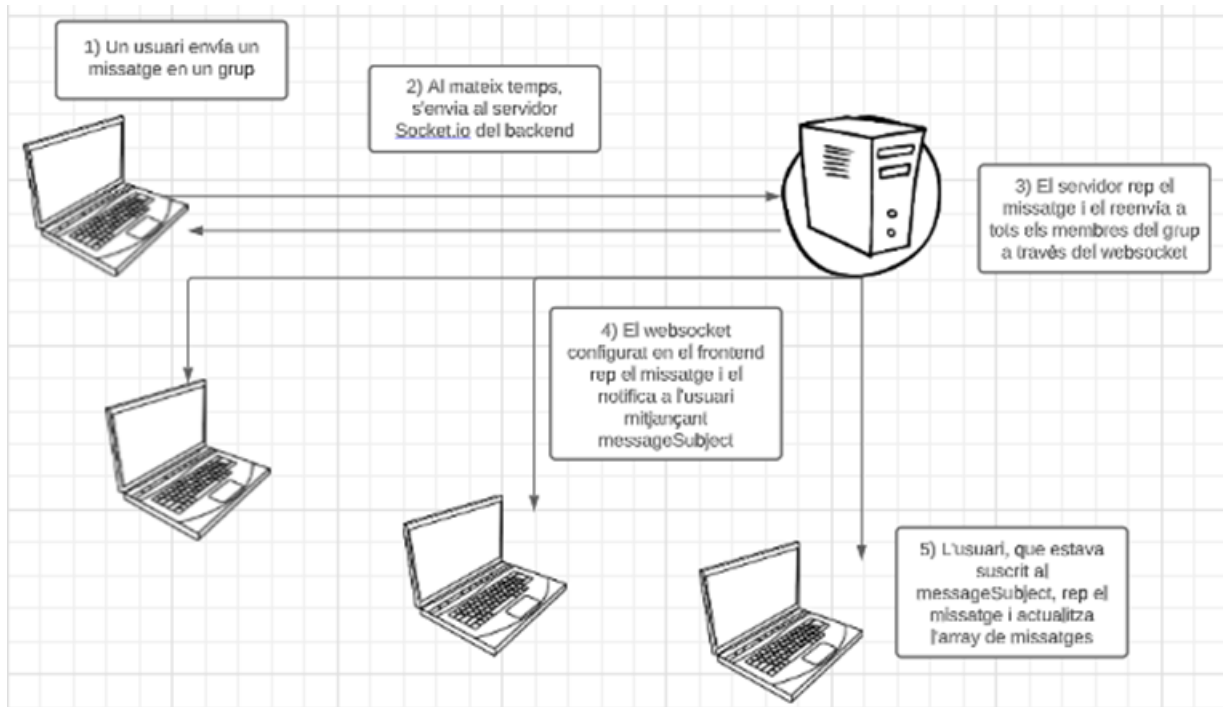


Figura 5.6: Esquema del patró observador



**Figura 5.7:** Exemple dels esdeveniments al enviar un missatge a un grup

Per finalitzar aquesta funcionalitat i assegurar-nos de que funciona correctament, s'han realitzat una sèrie de proves d'acceptació que es poden observar en la taula 5.3.

Id PA	Nom PA	Descripció	Resultat
84	Enviament de missatge exitós a un usuari	<p>CONDICIONS</p> <ul style="list-style-type: none"> <li>· L'usuari té com a mínim una connexió.</li> </ul> <p>PASSOS</p> <ul style="list-style-type: none"> <li>· L'usuari es dirigeix al xat d'alguna connexió.</li> <li>· L'usuari escriu un missatge.</li> <li>· Prem el botó d'enviar missatge.</li> </ul> <p>RESULTAT</p> <ul style="list-style-type: none"> <li>· El missatge s'envia correctament i es crea el missatge en el xat.</li> </ul>	OK

85	Enviament de missatge exitós a un grup	<p>CONDICIONS</p> <ul style="list-style-type: none"> <li>· L'usuari està en un grup</li> </ul> <p>PASSOS</p> <ul style="list-style-type: none"> <li>· L'usuari es dirigeix al xat del grup.</li> <li>· Escriu un missatge.</li> <li>· Prem el botó d'enviar missatge,</li> </ul> <p>RESULTAT</p> <ul style="list-style-type: none"> <li>· El missatge s'envia correctament i es crea el missatge en el grup.</li> </ul>	OK
86	Rebut de missatge exitós d'un usuari	<p>CONDICIONS</p> <ul style="list-style-type: none"> <li>· L'usuari té com a mínim una connexió.</li> </ul> <p>PASSOS</p> <ul style="list-style-type: none"> <li>· L'usuari es dirigeix al xat d'una connexió.</li> <li>· La seua connexió li envia un missatge.</li> </ul> <p>RESULTAT</p> <ul style="list-style-type: none"> <li>· L'usuari rep el missatge correctament i es crea en el xat.</li> </ul>	OK
87	Rebut de missatge exitós d'un grup	<p>CONDICIONS</p> <ul style="list-style-type: none"> <li>· L'usuari està en un grup.</li> </ul> <p>PASSOS</p> <ul style="list-style-type: none"> <li>· L'usuari es dirigeix al xat del grup.</li> <li>· Algun membre del grup envia un missatge.</li> </ul> <p>RESULTAT</p> <ul style="list-style-type: none"> <li>· L'usuari rep el missatge correctament i es crea en el grup.</li> </ul>	OK

**Taula 5.3:** Proves d'acceptació del xat

A més, es va decidir l'automatització de proves, amb l'ajuda de Jasmine i Karma [22], que ja estan integrats en Angular. Jasmine s'utilitza per a definir les regles que han de seguir els tests per tal d'aprovar-los i Karma és el que automatitza l'execució de tots els tests.

```
1   ...
2   it('Deuria enviar un missatge al premer el boto', fakeAsync(() => {
3     const selectedGroup: Group = {
4       id: 642863280530736,
5       name: 'Grup verd',
6       members: [
7         6428484113206182219,
8         6428480132061822133
9       ],
10      messages: [],
11      files: []
12    };
13    const activeUser: User = {
14      id: 6428484113206182219,
15      name: 'Josep',
16      surname: 'Martin Torres',
17      email: 'josepcdg@gmail.com',
18      password: '1234',
19      img_url: '',
20      tfg_url: '',
21      description: 'Estudiant d Enginyeria Informatica',
22      university_id: 0,
23      degree: '',
24      rol: '',
25      status: ''
26    };
27    const messageToSend = 'Missatge de prova';
28    const selectedFile = {
29      href: '',
30      filename: ''
31    };
32
33    const mockMessage: Message = {
34      emitter: activeUser,
35      text: messageToSend,
36      file: selectedFile
37    };
38
39    component.sendMessage();
40
41    expect(grupoServiceSpy.createMessage).toHaveBeenCalledWith(
42      selectedGroup.id, activeUser.id, messageToSend, undefined);
43    expect(socketServiceSpy.sendMessage).toHaveBeenCalledWith(
44      selectedGroup.id, mockMessage);
45
46    tick();
```

```

45
46     expect (component.messages).toContain (mockMessage) ;
47
48     });
49     ...

```

En aquesta peça de codi es pot observar com s'ha automatitzat la comprovació de que, a l'enviar un missatge en el xat, es criden correctament els mètodes `createMessage` i `sendMessage`, els quals creen un missatge en la base de dades i l'envien al servidor de `Socket.io`, respectivament. A més, una vegada executats ambdós mètodes, comprovem que el missatge s'haja creat e inclòs en l'*array* de missatges 'messages', que és el que conté els missatges que veu l'usuari.

### 5.3.2. Avaluació de l'Sprint

El diagrama de flux acumulat mostra el progrés de l'*sprint*, representant amb diversos colors les UTs depenent de l'activitat en la que es troben. En el nostre cas, el groc representa que la UT està en 'programar', el rosa que està en 'provar' i el blau en 'finalitzada'.

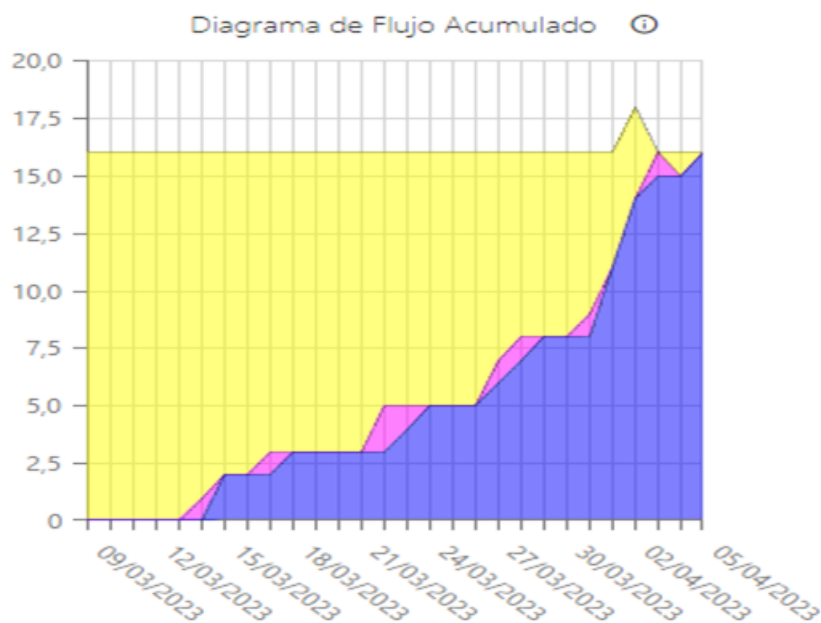
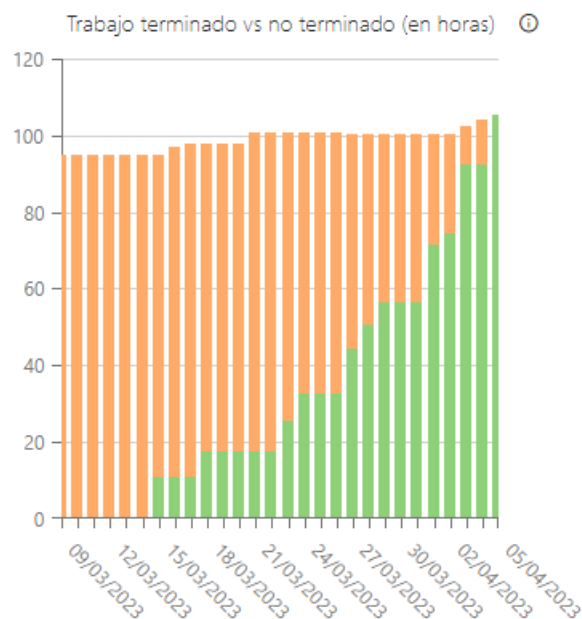


Figura 5.8: Diagrama de flux acumulat de l'*sprint* 1

A l'examinar la gràfica de la figura 5.8, es pot destacar el fet que en la primera setmana no es va acabar cap UT. Això és degut a que, al ser el primer *sprint* on es programava, vam tenir que crear una distribució de carpetes per tal de tenir els arxius organitzats i no tindre problemes a l'hora d'anar generant més codi i escalant l'aplicació. De mateixa forma, es pot contemplar com hi ha una



gran inclinació en la última setmana de l'*sprint*, indicant que eixa setmana es van acabar moltes més UT de les que s'haurien acabat si s'hagués seguit una finalització constant d'UT. L'explicació a aquest fenomen és que les UT que es van desenvolupar a la meitat de l' *sprint*, enviar i rebre missatges, tenien una duració molt llarga, de 12 i 6 hores respectivament, pel que, encara que no es va deixar de treballar durant tot l'*sprint*, pareix que es van deixar les tasques per al final. Aquestes afirmacions es veuen reforçades amb la gràfica de treball finalitzat vs no finalitzat que s'observa en la figura 5.9.



**Figura 5.9:** Treball finalitzat vs no finalitzat (en hores) de l'*sprint 1*

A més, gràcies a la gràfica que s'observa en la figura 5.10, es pot corroborar el que s'anomenava en el paràgraf anterior, ja que, llevat d'una setmana que per motius personals no vaig poder treballar, es pot veure com s'ha anat registrant temps constantment i, per tant, treballant, encara que no ho puga semblar en el diagrama de flux acumulat.

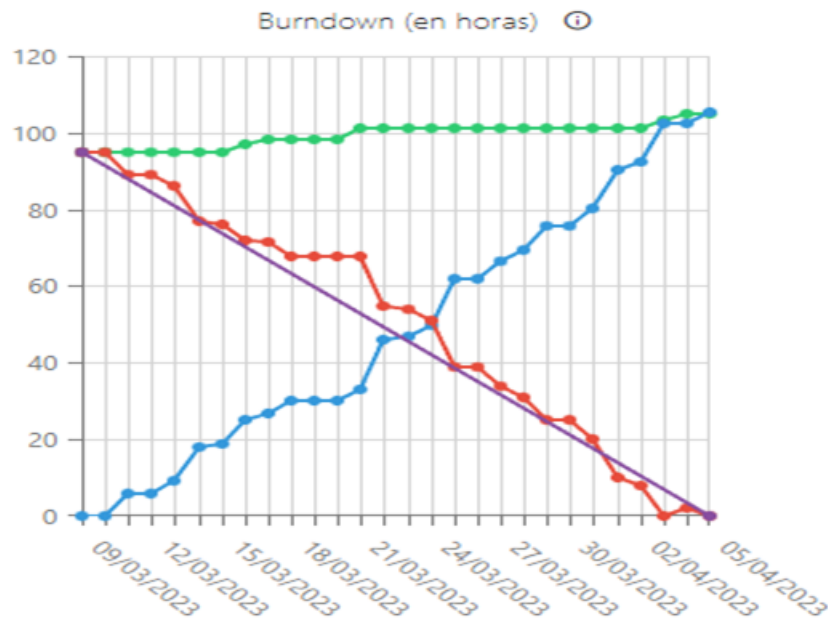


Figura 5.10: Gràfica de burndown en hores de l'sprint 1

## 5.4 Sprint 2

El 6 d'Abril va començar el segon *sprint*, enfocant-lo principalment a una altra de les principals funcionalitats de ConnectFG, el sistema de notifiacions. A més, també les funcions d'acceptar/rebutjar candidats, i capacitar als usuaris per poder editar la descripció del grup per posar més informació sobre el projecte o eliminar-lo, en cas de no arribar a cap acord. En la taula 5.4 es poden veure ordenades les UT realitzades durant aquest segon *sprint*.

Ordre	Identificador d'UT	Nom	Estimació (en h)
5	256	<i>Aceptar candidato</i>	10
15	257	<i>Rechazar candidato</i>	4
25	145	<i>Recibir notificaciones</i>	20
35	144	<i>Eliminar notificación</i>	4
45	246	<i>Eliminar participantes del grupo</i>	7
55	247	<i>Eliminar grupo</i>	4
65	245	<i>Editar descripción del grupo</i>	3

Taula 5.4: Llistat d'UTs de l'sprint 2

### 5.4.1. Gestió de candidats

Com ja s'ha comentat en capítols anteriors, els usuaris compten amb un *feed* on es troben les idees de TFG que han proposat altres estudiants i, si estan interessats en alguna, es postulen, convertint-se en candidats d'aquestes.

En aquest punt és on prenen part les UT 256 i 257; ja que, una vegada es postula algú, l'autor de la publicació rep una sol·licitud de connexió, que pot acceptar o rebutjar.

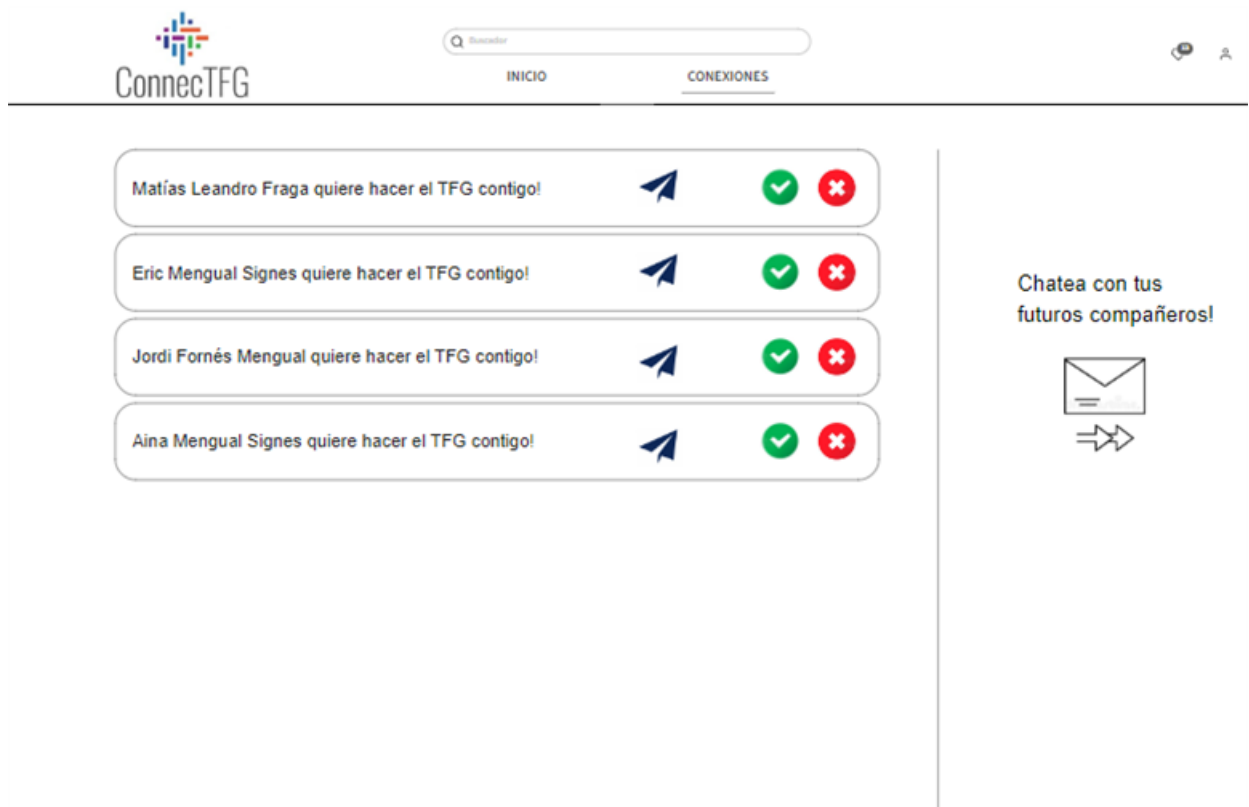
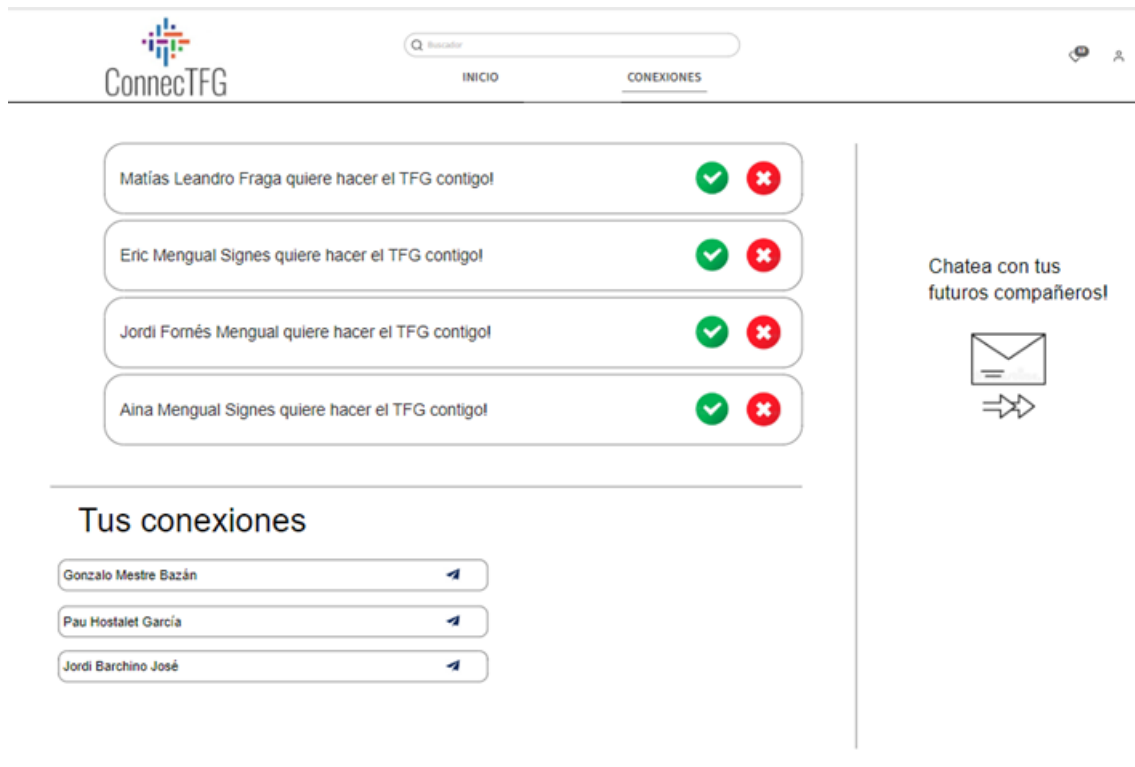


Figura 5.11: Mockup de les UT 256 i 257

En la figura 5.11 es pot veure el disseny inicial amb el que comptava la finestra de connexions, permetent a l'usuari escollir amb qui vol treballar i amb qui no. Aquesta ha sigut redissenyada en el tercer *sprint* per tal d'incloure en la mateixa finestra a les connexions que ja ha acceptat i així poder iniciar un xat amb ells, com es pot veure en la figura 5.12.



**Figura 5.12:** Mockup actualitzat de la gestió de candidats

A més, vam crear diverses proves d'acceptació per tal d'assegurar-nos que funcionava correctament, com es pot observar en la taula 5.5.

<b>Id PA</b>	<b>Nom PA</b>	<b>Descripció</b>	<b>Resultat</b>
94	Comprovar que els usuaris poden canviar l'estat dels candidats a acceptat	<p>CONDICIONS</p> <ul style="list-style-type: none"> <li>· L'usuari encara no ha sigut acceptat per l'administrador de la publicació.</li> </ul> <p>PASSOS</p> <ul style="list-style-type: none"> <li>· L'administrador de la publicació prem el <i>tick</i> verd de la notificació de que un usuari vol fer el tfg amb ell.</li> <li>· Prem acceptar en el diàleg de confirmació.</li> </ul> <p>RESULTAT</p> <ul style="list-style-type: none"> <li>· L'usuari s'afegeix correctament en la llista de candidats al post en la bd i com a connexió a l'administrador de la publicació.</li> </ul> <p>OBSERVACIONS</p> <ul style="list-style-type: none"> <li>· No poden unir-se dos vegades o més a les connexions de l'administrador.</li> <li>· No pot ser acceptat dos vegades o més per a un mateixa publicació.</li> </ul>	OK
95	Comprovar que els usuaris poden canviar l'estat dels candidats a rebutjat	<p>CONDICIONS</p> <ul style="list-style-type: none"> <li>· L'usuari encara no ha sigut acceptat per l'administrador de la publicació.</li> </ul> <p>PASSOS</p> <ul style="list-style-type: none"> <li>· L'administrador prem la creu roja en la notificació de que un usuari ha postulat en la seua publicació.</li> <li>· Prem acceptar en el diàleg de confirmació.</li> </ul> <p>RESULTAT</p> <ul style="list-style-type: none"> <li>· La notificació es borra i l'usuari no s'afegeix com a candidat.</li> <li>· Es mostra un <i>snackbar</i> notificant que s'ha eliminat correctament.</li> </ul>	OK

**Taula 5.5:** Proves d'acceptació realitzades en l'sprint 2

### 5.4.2. Sistema de notificacions

Aquesta aplicació compta amb un sistema de notificacions per tal de mantindre informats als usuaris sobre les novetats en les seues publicacions o connexions, ja siga perquè algun estudiant s'ha postulat a la seua publicació o ha acceptat la

seua sol·licitud de connexió. El *mockup* d'aquesta pàgina es pot veure en la figura 5.13.

Es pot accedir a la finestra de notificacions dirigint-se a la icona de campana que es troba en la capçalera. A més, aquesta icona conta amb un nombre en la part superior dreta que indica la quantitat de notificacions que ha rebut.



**Figura 5.13:** *Mockup* de les UT 144 i 145

Per tal que les notificacions funcionen en temps real, s'ha seguit el mateix patró que en el sistema de missatgeria en la part del *frontend* però lleugerament diferent en la part del backend. Això és degut a que, en el sistema de missatgeria, al pertànyer a un mateix grup, es sap el *websocket* al que has d'enviar el missatge; però això no funciona en aquest cas, ja que al no tenir cap relació entre els usuaris prèviament, no hi ha cap forma de conèixer el *websocket* al qual cal enviar-li la notificació.

Degut a això, el que es fa és crear una variable de tipus Map que emmagatzeme totes les tuples `userId` i `socketId` (*id* del *websocket*) dels usuaris que es connecten al servidor, com es pot observar en el següent fragment de codi:

```

1  const usersConnected = new Map();
2
3  io.on('connection', (socket) => {
4      ...

```

```
5     const socketId = socket.id;
6     const userId = socket.handshake.query.userId;
7     usersConnected.set(userId, socketId);
8     ...
9  })
```

D'altra banda, per tal de mantenir només els usuaris que estan connectats, necessitem que s'eliminen d'aquesta variable quan es desconnecten del servidor. Açò ho farem recorrent el map fins trobar una entrada en la que coincidisca el valor de l'*id* del socket i llevant-lo si el troba.

```
1     const usersConnected = new Map();
2
3     io.on('connection', (socket) => {
4         ...
5         socket.on('disconnect', () => {
6             for(const [userId, socketId] of usersConnected.entries()) {
7                 if (socketId === socket.id) {
8                     usersConnected.delete(userId);
9                     break;
10                }
11            }
12        });
13    })
```

D'aquesta forma, ja tenim una variable amb tots els usuaris que estan actius en l'aplicació en temps real, sabent el seu *id* d'usuari i l'*id* del seu websocket. Sabent açò, i de forma similar al sistema de missatgeria, les notificacions s'envien amb un mètode emit dirigit al *websocket* de l'usuari al que va dirigida la notificació.

```
1     const usersConnected = new Map();
2
3     io.on('connection', (socket) => {
4         ...
5         socket.on('newNotification', (data) => {
6             const { userId, message } = data;
7
8             const userSocketId = usersConnected.get(userId);
9
10            if (userSocketId) {
11                io.to(userSocketId).emit('notification', message);
12            }
13        });
14        ...
15    })
```

Una vegada el servidor ja està capacitat per a rebre notificacions i tenint el *frontend* acabat també, seguint els mateixos passos que en el sistema de missatgeria, mancaria sols crear els mètodes necessaris per a cridar-los quan volem enviar una notificació a algun usuari. Per això, serà necessari crear un servei 'NotificationService' on ubicarem tots els mètodes i que injectarem en els components on calga utilitzar-los, els quals són 'PublicationListComponent', per a les notificacions de presentar-se com a candidat, i 'ConnexionListComponent', per a les notificacions corresponents a l'acceptació de la connexió.

Finalment, s'haurien de passar totes les proves d'acceptació per tal de comprovar que funciona correctament i donar per acabades les UT. Aquestes es poden observar en la taula 5.6.

<b>Id PA</b>	<b>Nom PA</b>	<b>Descripció</b>	<b>Resultat</b>
96	Rebre avís sol·licitud de connexió	<p>CONDICIONS</p> <ul style="list-style-type: none"> <li>· Un usuari s'ha postulat en la teua publicació.</li> </ul> <p>PASSOS</p> <ul style="list-style-type: none"> <li>· Prem la campana de notificacions.</li> </ul> <p>RESULTAT</p> <ul style="list-style-type: none"> <li>· Veus la notificació.</li> </ul>	OK
98	Rebre avís t'han acceptat la sol·licitud de connexió	<p>CONDICIONS</p> <ul style="list-style-type: none"> <li>· Un usuari ha acceptat la teua sol·licitud de connexió.</li> </ul> <p>PASSOS</p> <ul style="list-style-type: none"> <li>· Prem la campana de notificacions.</li> </ul> <p>RESULTAT</p> <ul style="list-style-type: none"> <li>· Veus la notificació.</li> </ul>	OK

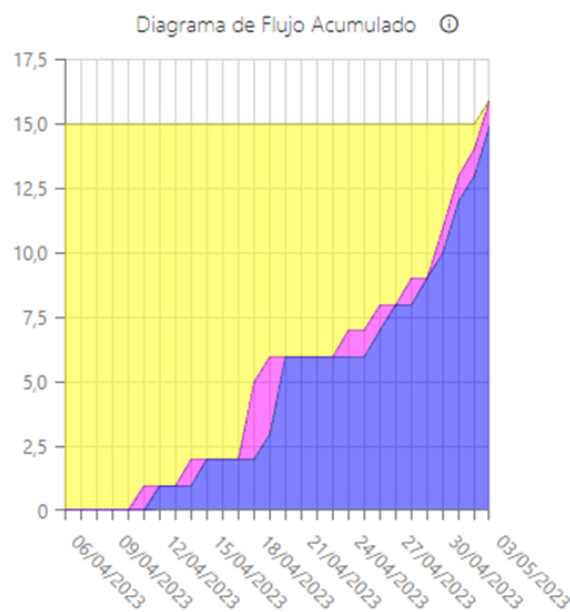


97	Eliminar notificació	<p><b>CONDICIONS</b></p> <ul style="list-style-type: none"> <li>· Tens 1 o més notificacions.</li> </ul> <p><b>PASSOS</b></p> <ul style="list-style-type: none"> <li>· Prem la campana de notificacions.</li> <li>· Prem la icona de eliminar.</li> </ul> <p><b>RESULTAT</b></p> <ul style="list-style-type: none"> <li>· S'elimina la notificació de la interfície.</li> </ul>	OK
----	----------------------	---	----

**Taula 5.6:** Proves d'acceptació del sistema de notificacions

### 5.4.3. Avaluació de l'*sprint*

En aquest *sprint*, analitzarem el diagrama de flux acumulat i el burndown en hores, per tal d'obtenir una comprensió més clara sobre l'evolució de l'*sprint* i del progrés general de l'equip.



**Figura 5.14:** Diagrama de flux acumulat de l'*sprint 2*

En el diagrama de la figura 5.14 podem observar com en la primera setmana no es va acabar cap UT, per les mateixes raons que en l'*sprint* anterior. A més, s'observa una evolució prou constant al llarg de tot l'*sprint*, a excepció dels dies 19 i 20 d'Abril, on es van acabar quatre UT i trobem una inclinació més accentuada. Açò es degut a que les UT que vam començar en la primera setmana tenien una

duració major i, per tant, es van finalitzar quasi simultàniament en la segona setmana.

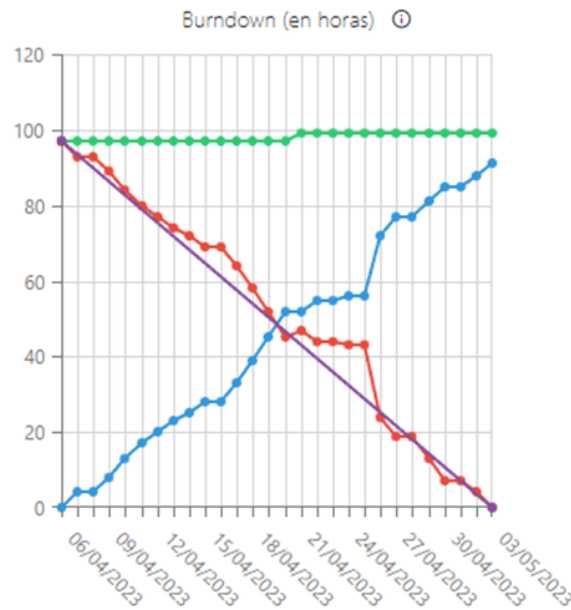


Figura 5.15: Gràfica burndown en hores de l'sprint 2

L'anàlisi del diagrama de flux acumulat es veu reforçat amb la gràfica de la figura 5.15, ja que, com en l'anterior *sprint*, es pot observar una constant dedicació durant tot el mes, encara que la primera setmana no s'havia acabat cap UT. Aquesta dedicació s'atura durant la setmana del 20 al 25 d'Abril, ja que havíem decidit programar de manera conjunta el diumenge 26, on vam recuperar totes les hores de la setmana pendents, com es pot observar en la gràfica. A més, vam aprofitar per provar presencialment que el sistema de notificacions funcionava correctament, seguint les proves d'acceptació, i donant-les per finalitzades els dies 26 i 27, com es pot observar en la gràfica de flux acumulat.

## 5.5 Sprint 3

En el tercer i últim *sprint* ens vam centrar en refinar l'aplicació per tal de presentarla en el TFG, refactoritzant algunes interfícies per tal d'oferir una millor navegabilitat i experiència d'usuari. Per tal d'assolir aquest objectiu, es van dur a terme un conjunt d'UT, les quals es poden observar ordenades en la taula 5.7.

Ordre	Identificador d'UT	Nom	Estimació (en h)
5	361	Rediseño de la interfaz (más intuitiva)	5
15	365	Indicar autor del mensaje	7
25	366	Diferenciar los mensajes del usuario de los otros	3
35	362	Visualización de conexiones actuales	10
45	239	Enviar archivos al chat	14
55	162	Internacionalización	15
65	364	Desplegar la aplicación web	3
75	363	Encuesta de usabilidad y análisis de resultados	5

Taula 5.7: Llistat d'UTs de l'sprint 3

### 5.5.1. Enviament d'arxius al xat

Una característica interessant a l'hora de buscar companys de TFG és la de passar documents pel xat, principalment enfocat a currículums, amb la intenció de presentar-te més formalment amb qui vols col·laborar.



Figura 5.16: Mockup de la UT 239

Per tal que l'usuari tinga la possibilitat d'enviar fitxers, s'ha afegit una icona en forma de clip a l'esquerra del botó d'enviar missatge, com es pot observar en la figura 5.16. Al prémer aquesta icona, s'obrirà una finestra mostrant els fitxers de tipus pdf de l'usuari, encara que també es poden enviar fitxers d'altres extensions, com per exemple png.

Centrant-nos en l'aspecte tecnològic d'aquesta funcionalitat, el que es fa és, una vegada seleccionat l'arxiu a enviar i premut el botó d'enviar, s'envia tant al backend com al socket, per tal de que s'envie als altres membres del grup també.

Açò s'aconsegueix creant un objecte de tipus 'Message', que conté els mateixos atributs que el model del backend, com es pot veure a continuació:

```
1 let message: Message = {
2   emitter: this.loginService.getActiveUser(),
3   text: this.messageToSend,
4   file: {
5     href: URL.createObjectURL(this.selectedFile!),
6     filename: this.selectedFile?.name!,
7   }
8 }
```

Aquest objecte és el que s'envia al socket per tal de que pugui tractar el missatge de la mateixa forma que tracta als del backend. En canvi, el backend s'envia l'arxiu seleccionat directament.

```
1 this.grupoService.createMessage(
2   this.selectedGroup.id, this.loginService.getUserId()!,
3   this.messageToSend, this.selectedFile!
4 ).subscribe(response => { });
5
6 this.socketService.sendMessage(this.selectedGroup.id, message)
```

El missatge que s'envia a través del webscoket es tracta de forma anàloga a un missatge sense text, com ja s'ha explicat en apartats anteriors. En aquesta ocasió anem a centrar-nos a com controlar un arxiu en el backend per tal de guardar-lo directament a la base de dades.

Primerament, en el *frontend*, cal guardar en un objecte de tipus `FormData` un nom, 'file' en el nostre cas, i tots els paràmetres que es volen transmetre al backend, en el nostre cas: l'*id* del grup on s'ha enviat el missatge, l'autor, un arxiu (si es que s'ha adjuntat) i el missatge en sí.

Una vegada realitzada la petició al backend, en la ruta de grups, (la primera capa per la que passen les peticions), s'ha d'importar la llibreria *multer*, que és

l'encarregada de gestionar les peticions de tipus 'multipart/form-data', és a dir, les peticions amb arxius.

```
1 const upload = multer()
2 ...
3 groupRouter.post('/messages', upload.single('file'), createMessage)
4 ...
```

Així doncs, quan es rep la petició, es crida al mètode del controlador, 'createMessage', passant-li l'arxiu que s'ha enviat des del *frontend* gràcies al mètode 'upload.single('file')', el qual indica que s'espera rebre un únic arxiu en la sol·licitud HTTP, anomenat 'file' en aquest exemple.

Una vegada en el mètode del controlador, es guardaran els paràmetres en variables i es cridarà al mètode 'createMessage' del servei de grups, on es crearà el missatge en el format que emmagatzema el backend, com es pot veure en aquest fragment de codi:

```
1 const message = new MessageModel({
2   text: text || '',
3   file: {
4     data: file.buffer,
5     contentType: file.mimetype,
6     filename: file.originalname
7   },
8   emitter: emitter
9 })
```

En aquest fragment de codi es pot observar com el fitxer, a més de guardar el seu contingut en bytes (el que seria file.buffer), també guarden el seu tipus de document i el nom original, això és fa per tal que es mantinga el nom original, ja que al emmagatzemar l'arxiu en bytes es perdria.

Una vegada completada la implementació del codi, tant en el *frontend* com en el backend, s'han de superar una sèrie de proves unitàries que tenia associades aquesta UT, com podem observar en la figura 5.8, on el resultat de les tres és 'OK'.

<b>Id PA</b>	<b>Nom PA</b>	<b>Descripció</b>	<b>Resultat</b>
118	S'envien arxius pdf	<p>CONDICIONS</p> <ul style="list-style-type: none"> <li>· L'usuari està en un grup/xat.</li> </ul> <p>PASSOS</p> <ul style="list-style-type: none"> <li>· Prem la icona de clip.</li> <li>· Selecciona un arxiu local en format pdf.</li> <li>· Envia el missatge.</li> </ul> <p>RESULTAT</p> <ul style="list-style-type: none"> <li>· S'envia correctament.</li> </ul>	OK
119	S'envien imatges	<p>CONDICIONS</p> <ul style="list-style-type: none"> <li>· L'usuari està en un grup/xat.</li> </ul> <p>PASSOS</p> <ul style="list-style-type: none"> <li>· Prem la icona de clip.</li> <li>· Selecciona un arxiu local en format d'imatge (jpg, png, ...)</li> <li>· Envia el missatge.</li> </ul> <p>RESULTAT</p> <ul style="list-style-type: none"> <li>· S'envia correctament.</li> </ul>	OK
120	No s'envien arxius no acceptats	<p>CONDICIONS</p> <ul style="list-style-type: none"> <li>· L'usuari està en un grup/xat.</li> </ul> <p>PASSOS</p> <ul style="list-style-type: none"> <li>· Prem la icona de clip.</li> <li>· Selecciona un arxiu local amb un format no acceptat (ni pdf ni imatge).</li> <li>· Prem el botó d'enviar el missatge.</li> </ul> <p>RESULTAT</p> <ul style="list-style-type: none"> <li>· No s'envia el missatge i es mostra un missatge d'error.</li> </ul>	OK

**Taula 5.8:** Proves d'acceptació d'enviament d'arxius al xat

### 5.5.2. Internacionalització

La internacionalització és el procés de dissenyar una aplicació de software de forma que puga adaptar-se i funcionar correctament en diversos idiomes, cultures i regions del món [23], en el nostre cas està adaptada al valencià i castellà.



Figura 5.17: Mockup de la UT 162

Com es pot observar en la figura 5.17, s'ha optat per un desplegable on l'usuari pot escollir entre els idiomes valencià i castellà. A més, buscant millorar l'experiència de l'usuari, l'aplicació detecta el idioma per defecte del navegador que està utilitzant i l'empra també en ConnectFG. Per tal d'implementar aquesta funcionalitat, s'ha fet ús de la llibreria ngx-translate, que permet la internacionalització en Angular.

Una vegada instal·lada la llibreria, s'ha de crear un arxiu json, per cada idioma que vols en l'aplicació, que continga totes les traduccions. Aquests arxius han d'estar ubicats en la ruta 'src/assets/i18n/', com es poden veure en la figura 5.18.

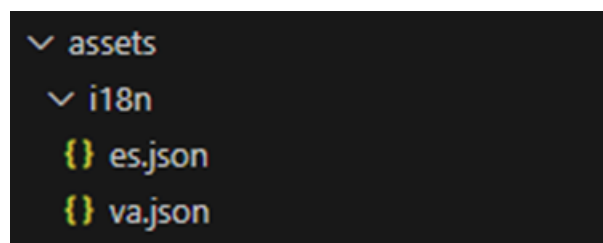


Figura 5.18: Arxius que contenen les traduccions

```

1  {
2    "botones": {
3      "cerrar": "Cerrar",
4      "guardar": "Guardar",
5      "editar": "Editar",
6      "borrar": "Borrar"
7    },
8
9    "header": {
10     "buscador": {
11       "placeholder": "Busca a tus futuros compañeros..."
12     },
13     "tabs": {
14       "inicio": "Inicio",
15       "chat": "Chat"
16     },
17     "perfil": {
18       "titulo": "Perfil",
19       "ajustes": "Ajustes",
20       "sobreNosotros": "Sobre nosotros",
21       "cerrarSesion": "Cerrar sesión"
22     }
23   }

```

Figura 5.19: Arxiu amb les traduccions al castellà

Com es pot observar en la figura 5.19, els arxius json es podrien considerar una llista de claus-valor, on les claus (text en blau) han de ser iguals en tots els fitxers de traduccions, mentre que el valor (text en taronja) és el que varia entre arxius.

Açò és degut a que, des del *frontend*, per tal d'obtindre la traducció, s'ha d'indicar la ruta de claus i una etiqueta 'translate' per a expressar que s'ha de buscar en l'arxiu json de l'idioma que l'usuari tinga seleccionat en eixe instant. Per tal d'entendre millor aquest concepte, aquest podria ser un exemple de text que es traduirà:

```

1  {{ 'header.tabs.inicio' | translate }}

```

Així doncs, com podem observar en l'anterior imatge, la ruta 'header.tabs.inicio', si l'idioma seleccionat és el castellà, es substituïra per 'Inicio' i, si es selecciona l'idioma valencià, es substituïra dinàmicament a 'Inici', com s'ha especificat en l'arxiu va.json.

### 5.5.3. Avaluació de l'*sprint*

En aquest apartat, a l'igual que en el dels *sprints* anteriors, analitzarem l'evolució de l'*sprint* per mitjà del diagrama de flux acumulat i el burndown en hores, en aquest ordre.



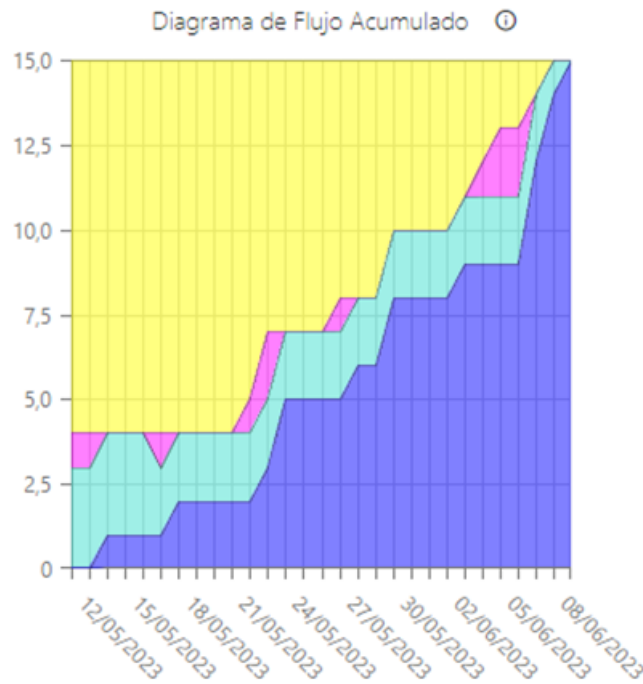
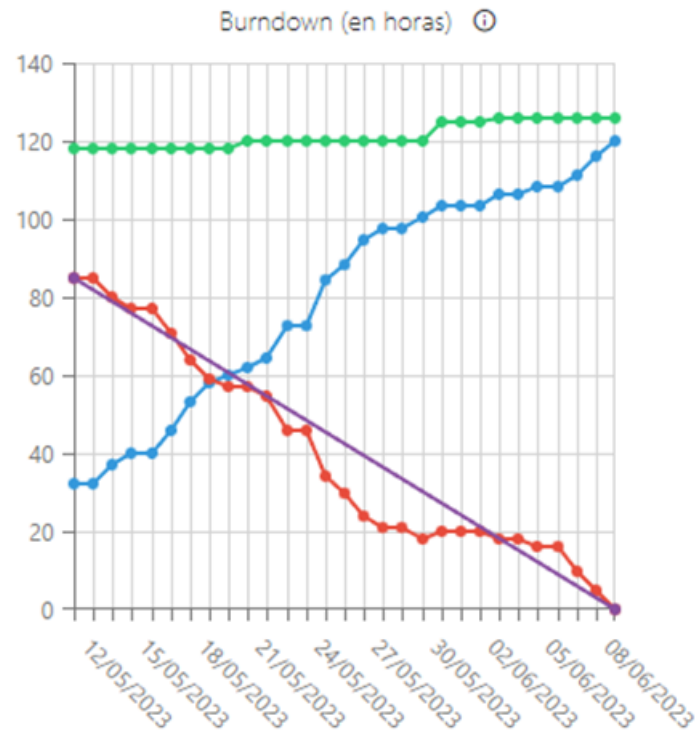


Figura 5.20: Diagrama de flux acumulat de l'sprint 3

En el diagrama de la figura 5.20 se puede observar una evolución notoria desde el inicio, debido a que primero se va a dedicar tiempo a las actividades de *refactoring* de la interfaz, las cuales, al ser de menor duración a las UT iniciales de *sprints* pasados, se van a acabar pronto. No obstante, se puede observar como en la semana del 30 de Mayo al 6 de Junio no se acaban casi UT. Esto se debe simplemente a la ausencia de mi compañero por motivos personales. Además, las UT que quedaban por hacer, la 162, 364 y 363, estaban relacionadas con las de Matías, por lo que resultó imposible avanzar el trabajo durante esta semana. No obstante, una vez más, se retomó el ritmo de trabajo, se finalizaron rápidamente un gran número de UT, al tratarse de UT de poca duración, y estando las cosas preparadas para incluir directamente su parte, solo se acabó. Este hecho se ve reflejado claramente en el pico del 7 de Junio, cuando se retomó la programación.



**Figura 5.21:** Gràfica burndown en hores de l'sprint 3

En aquesta gràfica, a l'igual que en l'sprint 0, podem observar una absoluta correlació amb el diagrama de flux acumulat, on, comparant les gràfiques podem veure com inicialment van acabant-se les UT poc a poc; posteriorment, a l'haver un esforç major, podem veure com també s'acaben moltes UT, seguit per la setmana en la que no es va comptabilitzar cap esforç (i per tant, no es va acabar cap UT), i acabant les restants en la última setmana.

---

---

# CAPÍTOL 6

## Desplegament

---

En aquest capítol, abordarem el desplegament de l'aplicació ConnecTFG, incloent tant el *frontend* com el *backend*. Explorarem els passos i les consideracions essencials per a la posada en marxa de la plataforma, assegurant l'accessibilitat per als usuaris finals.

### 6.1 Introducció

---

El desplegament de software, o *deploy* en anglès, [24], [25] és el procés de llançar una aplicació en un entorn de producció, on els usuaris finals poden utilitzar-lo independentment d'on es troben. Aquest procés compta amb una sèrie de tasques que pots realitzar i que detallarem en les següents seccions, ja que són diferents els passos seguits en el *frontend* dels del *backend*.

### 6.2 *Frontend* (Firebase)

---

Primerament, vam realitzar una cerca per tal de trobar aplicacions que compliren els nostres requisits: que fos gratuïta i fàcil de desplegar. D'aquest anàlisi van sorgir diversos noms, com ara Vercel, Heroku o Netlify, però ens vam acabar decidint per Firebase Hosting, un servei d'allotjament web proporcionat per Google que complia aquests requisits i que, a més, havíem utilitzat anteriorment en el projecte de l'assignatura PIN. A continuació es detallen els passos seguits per al desplegament del *frontend*:

1. Creació del projecte en Firebase: es va crear el projecte, definint el nom i unint col·laboradors.

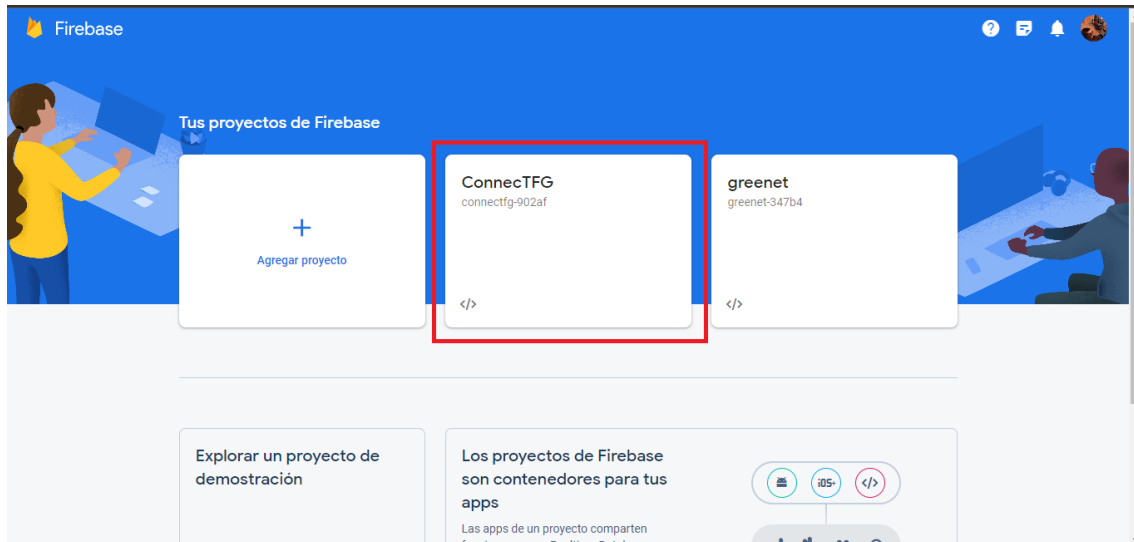


Figura 6.1: Consola de Firebase

2. Preparació: En el nostre cas ja estàvem preparats per a desplegar sense necessitat de variar el codi, ja que havíem estat variables d'entorn. Aquestes variables el que fan és que canvien el seu valor depenent de si et trobes en desenvolupament o en producció.

```

TS environments.ts X
connectFG-frontend > src > environments > TS environments.ts > ...
1 // This file can be replaced during build by using the `fileReplacements` array.
2 // `ng build` replaces `environment.ts` with `environment.prod.ts`.
3 // The list of file replacements can be found in `angular.json`.
4
5 export const environment = {
6   production: false,
7   secretKey: "TFGJOSEPMATIAS",
8   urlService: "http://localhost:8080",
9 };
10

```

Figura 6.2: Variables d'entorn per al desenvolupament

```

TS environment.prod.ts X
connectFG-frontend > src > environments > TS environment.prod.ts > ...
1 export const environment = {
2   production: true,
3   secretKey: "TFGJOSEPMATIAS",
4   urlService: "https://connectfg.onrender.com"
5 };
6

```

Figura 6.3: Variables d'entorn per a producció

Com es pot observar en les figures 6.2 i 6.3, en cas d'estar en desenvolupament, la url a la que es crida en els serveis és en *localhost* i, en canvi, en l'entorn de producció es crida a la url de Render [26], on està desplegat el backend. A més, un factor a tindre en compte al fer ús de variables d'entorn és la major seguretat que tens en cas de tindre el codi en un repositori públic; ja que, sempre que afegisques les variables d'entorn a l'arxiu `.gitignore`, la gent que llegeixi el codi vorà 'environment.urlService' però no sabrà el seu valor.

3. Generar aplicació Angular: el projecte compta amb un conjunt d'arxius i carpetes que, en alguns casos, poden arribar a ser moltes i derivar en una aplicació o pàgina web molt lenta. És per això que en angular comptem amb el comandament 'ng build', el qual compacta tota l'aplicació en un fitxer html, un css i uns quants fitxers javascript, emmagatzemats dins d'una carpeta anomenada 'dist', si no s'ha canviat en la configuració. En la figura 6.4 trobem un exemple d'aquesta carpeta amb els arxius creats al fer el *build*.

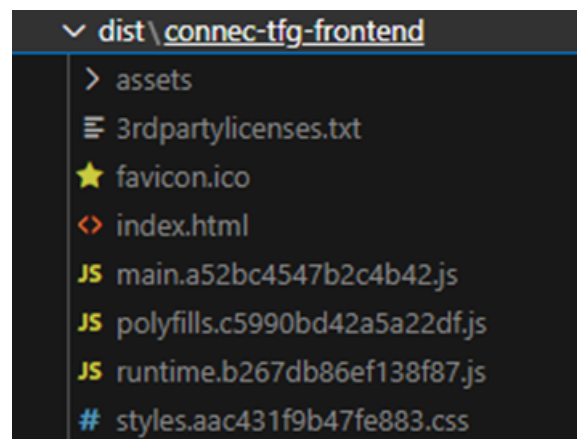


Figura 6.4: Fitxers de la carpeta dist al fer el build

4. Desplegament en Firebase Hosting: utilitzant Firebase CLI, des del mateix terminal de Visual Studio Code, s'executen els comandaments necessaris per a poder fer el deploy, els quals són iniciar sessió amb el compte de firebase, seleccionar un projecte, escollir quina funcionalitat desitges (Firebase Hosting en el nostre cas) i, una volta tot configurat, executes el comandament 'firebase deploy', el qual et crea una url on tindràs allotjada la teua aplicació i una carpeta, que per defecte s'anomena 'public', on es troben els fitxers que es veuran en la url donada. Per tal d'observar la nostra aplicació ConectTFG en aquesta url (<https://connectfg-902af.web.app/>), haurem de copiar els fitxers generats en el build i que podem trobar en la carpeta 'dist' en la carpeta 'public', generada per firebase. Actualment, si entrem en la url

esmentada, ja serem capaços d'observar l'aplicació completament desplegada, encara que no funcionarà fins que no despleguem també el backend.

## 6.3 Backend (Render)

Per al desplegament del *backend* també vam realitzar una cerca amb els requisits mencionats anteriorment, però aquesta vegada perquè funcionaren per aplicacions fetes amb Node.js, Express i es pogueren utilitzar els websockets perquè funcionara també la llibreria Socket.io. Amb aquests filtres, vam trobar noms com Heroku, Glitch o Vercel; però, finalment ens vam decantar per Render, un servei d'allotjament gratuït que permet executar el backend amb un certificat TLS gratuït, un nom CDN, protecció DDos i el més important, desplegaments automàtics de Git, el comporta que cada vegada que es feia un *push* a la rama 'main' del repositori de GitHub, es desplegaven els nous canvis sense la necessitat de fer res manualment. A continuació es detallen els passos per al desplegament del *backend*:

1. Configurar el projecte en Render: hi haurà que iniciar sessió amb el compte de GitHub i crear un nou servei web.

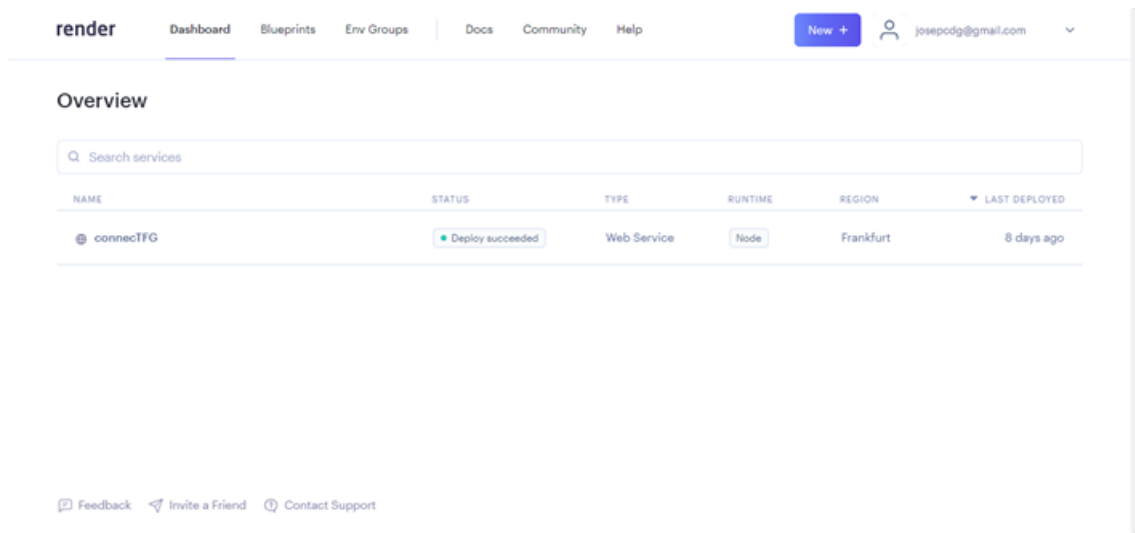


Figura 6.5: Dashboard de Render

2. Preparació: com ja s'ha esmentat en la part del *frontend*, en aquest cas també contàvem amb un arxiu 'env' on teníem les variables d'entorn, com es pot veure en la figura 6.6; però, com en aquest cas no es fa un build, sinó que s'executa el codi directament des del repositori, s'ha de configurar també en Render, com es pot observar en la figura 6.7.

```
.env
connectFG-backend > .env
1 MONGODB_URL='mongodb+srv://[redacted]:[redacted]@connectfg-cluster.jmqnczt.mongodb.net/?retryWrites=true&w=majority'
2 HTTP_PORT='8080'
3 HTTPS_PORT='443'
4
5 JWT_SECRET='ConnectFG'
```

Figura 6.6: Variables d'entorn del backend

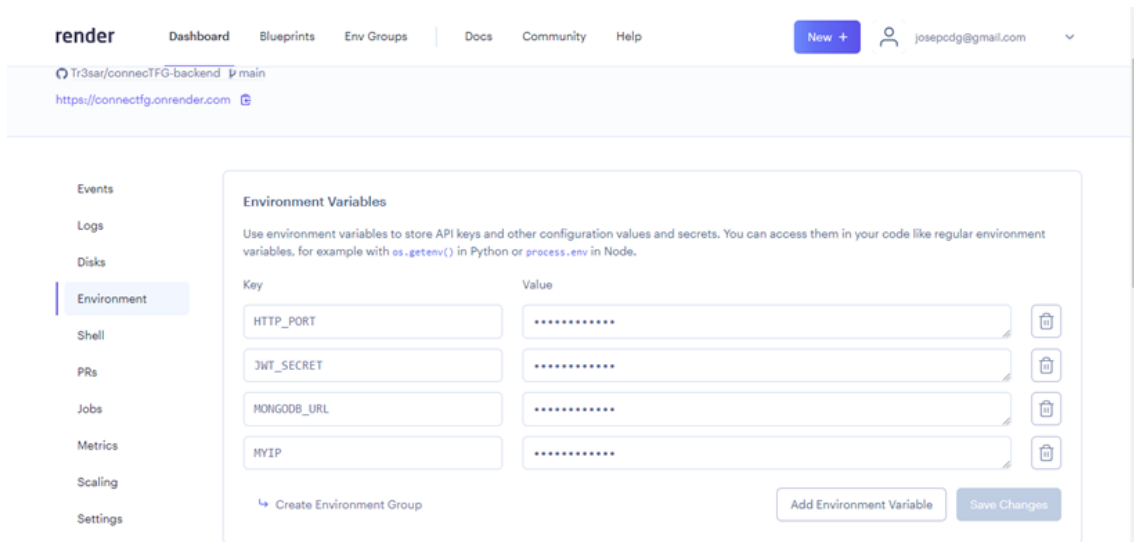


Figura 6.7: Configuració de les variables d'entorn en Render

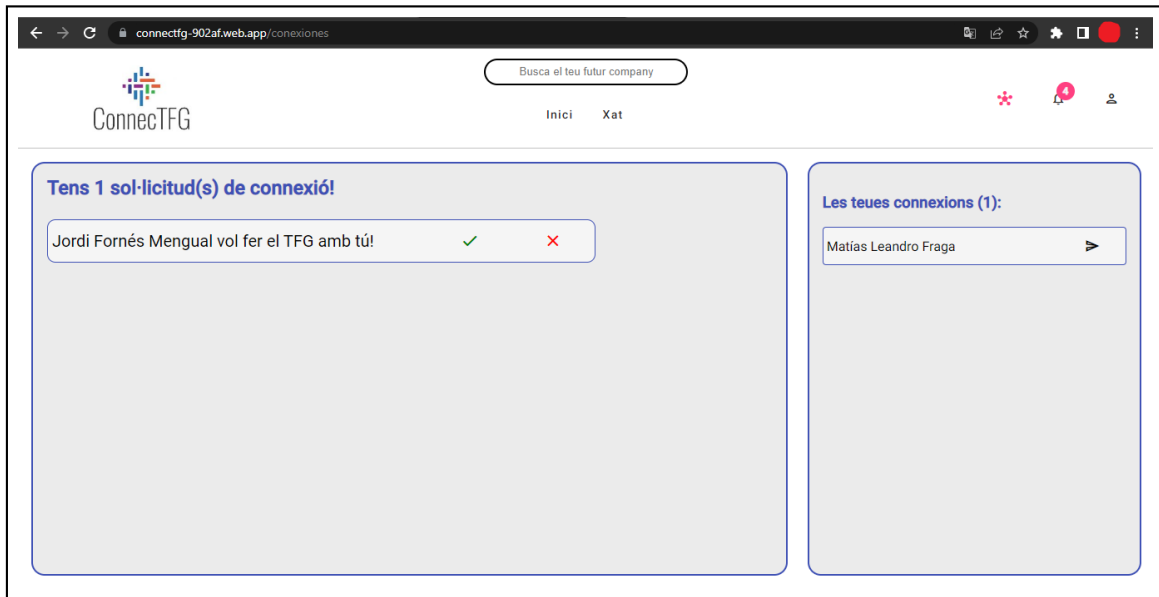
3. Desplegament: una vegada estiga tot configurat correctament, iniciarem el desplegament en Render, el qual s'encarregarà de clonar el repositori, instal·lar les dependències i executar l'aplicació Node.js, el que ens permetrà realitzar les consultes necessàries des del *frontend*, realitzant les consultes a la url que ens indica al visualitzar el projecte (<https://connectfg.onrender.com>).

## 6.4 Escenari d'ús

En aquesta secció es mostrarà un escenari d'ús de ConnectFG, il·lustrant el procés que seguirien els estudiants universitaris per tal de seleccionar els usuaris amb els que volen col·laborar e iniciar un xat amb ells.

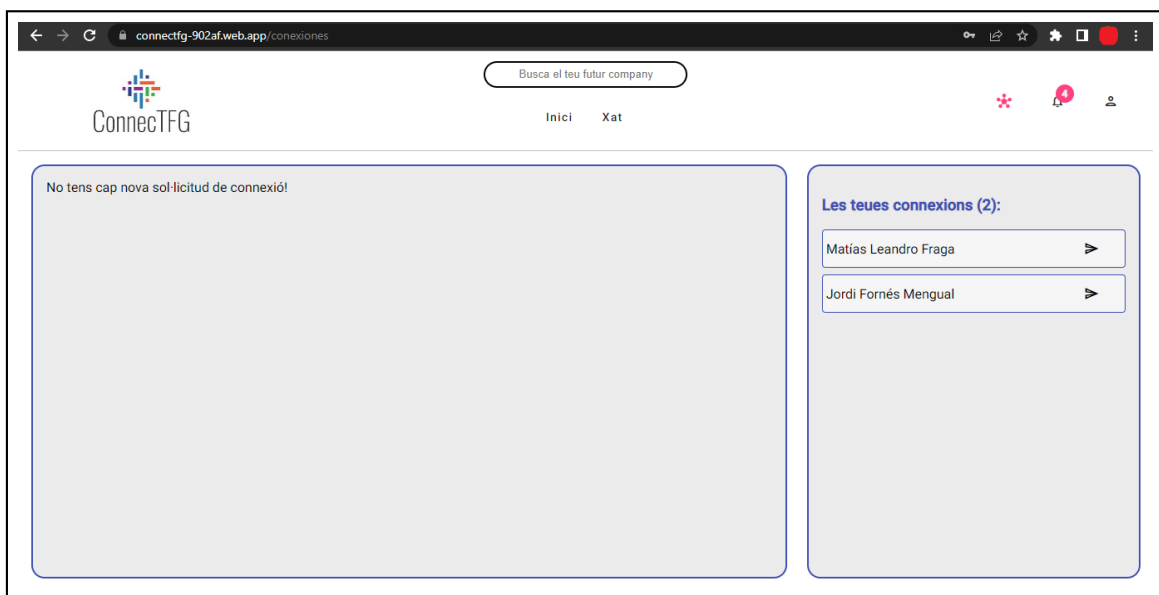
Una vegada s'ha desplegat de forma exitosa tant el *frontend* en Firebase Hosting com el backend en Render, els usuaris finals ja poden gaudir d'una aplicació completament funcional.

Primerament, una vegada un estudiant s'ha postulat a la teua publicació, es rep una notificació i es representa en la finestra de connexions com a una nova sol·licitud de connexió, com es pot observar en la figura 6.8.



**Figura 6.8:** Exemple d'una sol·licitud de connexió

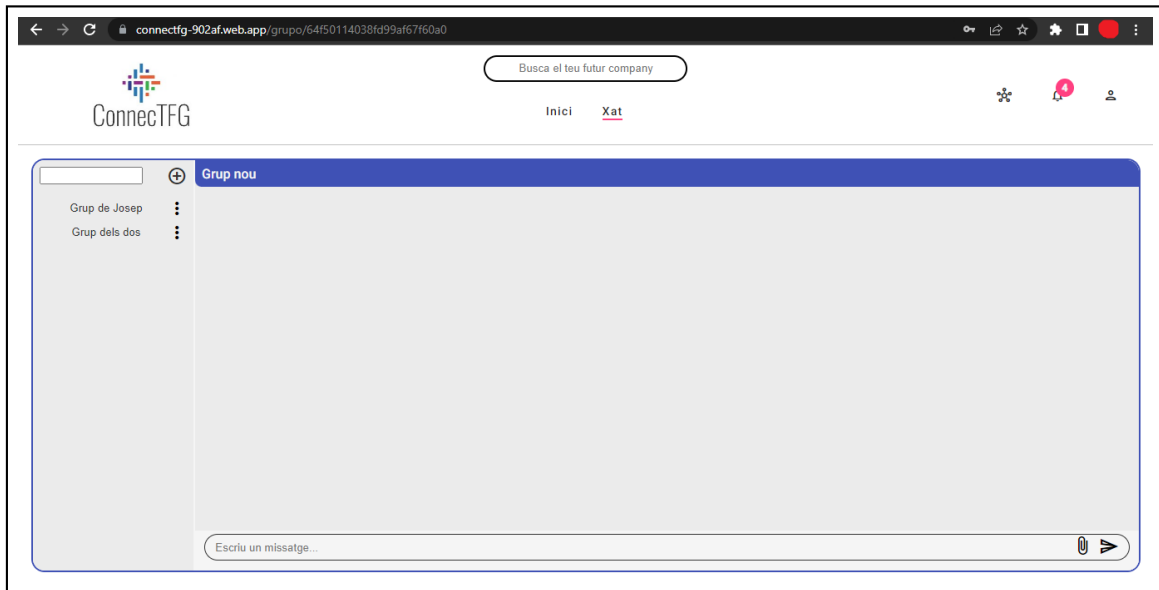
Una vegada es rep una sol·licitud, es pot acceptar (✓ verd) o rebutjar (X roja). En el cas d'acceptar-la, l'estudiant passa a formar part de les teues connexions, com es pot veure en al figura 6.9.



**Figura 6.9:** Finestra de connexions

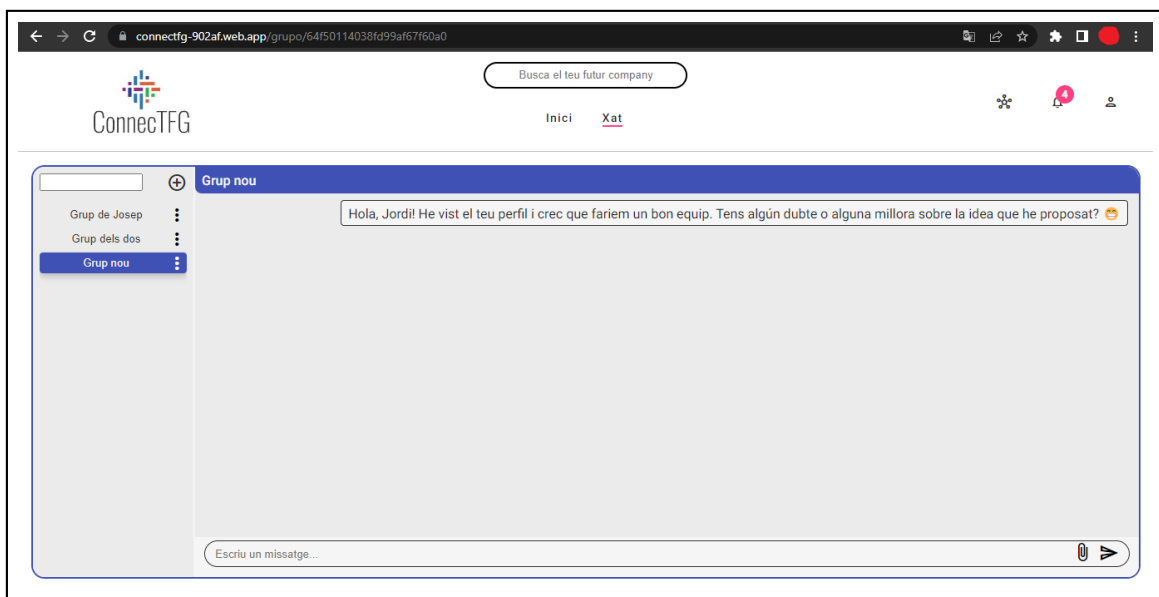
A més, si es prem la icona d'enviar missatge que es mostra a la dreta del nom, serà dirigit al xat amb l'usuari o, si no existeix, el crearà amb el nom 'Grup nou', com s'il·lustra a la figura 6.10.





**Figura 6.10:** Exemple de creació d'un xat

En últim lloc, tal com es mostra en la figura 6.11, s'envia un missatge a la nova connexió per tal d'establir una conversa i poder decidir si treballar junts o no.



**Figura 6.11:** Exemple d'iniciar la conversa

Aquesta interacció és un exemple de com ConnectFGG pot facilitar la creació de xarxes d'estudiants universitaris, permetent-los connectar-se amb persones amb interessos i objectius acadèmics similars ràpidament, presentant una interfície intuïtiva i atractiva.



---

---

# CAPÍTOL 7

## Conclusions i treballs futurs

---

Aquest capítol conclou el treball, establint la relació entre el projecte i els estudis realitzats. També es discuteixen les oportunitats de futur per a ConneCTFG i es considera com podria seguir desenvolupant-se.

### 7.1 Conclusions

---

El desenvolupament de ConneCTFG ha estat un gran èxit, ja que ha suposat un avanç significatiu en la millora dels futurs TFG, amb una aplicació innovadora que facilita la creació d'equips multidisciplinaris entre els estudiants. A través d'aquesta xarxa social, els estudiants tenen l'oportunitat de col·laborar entre ells, compartint recursos i coneixements, a més de tenir una experiència més pròxima al que serà el seu futur laboral.

Un dels principals reptes al que ens vam haver d'enfrontar va ser la creació d'un sistema de missatgeria en temps real integrat en la plataforma. Com ja s'ha vist en la secció 5.3.1, al ser la primera vegada que treballàvem amb *websockets* en aquest àmbit, i no aconseguíem que el xat s'actualitzara al enviar/rebre algun missatge, vam haver de configurar un patró observador que connectara el *frontend* i el *backend*, el qual, degut a la falta d'experiència en aquest context, va requerir una gran inversió de temps.

Tot i això, s'han aconseguit tots els subobjectius que conformen l'objectiu principal, la creació d'una plataforma digital que permetia connectar estudiants i trobar els companys idonis de TFG, des del desenvolupament d'una interfície d'usuari atractiva e intuïtiva fins a l'implementació d'una metodologia àgil Kanban, gràcies a l'ajuda de Worki.

En conclusió, durant aquests 4 mesos de treball ens hem enfrontat a multitud de reptes, fent-nos evolucionar com a programadors i com a persones; però, mal-

grat totes les adversitats, hem sabut sobreposar-nos i acomplir els objectius que ens havíem proposat inicialment.

## 7.2 Relació amb els estudis

---

El desenvolupament de ConnecTFG és el resultat de l'aplicació pràctica i l'aprenentatge adquirit a través dels estudis d'Enginyeria Informàtica. Les habilitats en programació, disseny de base de dades, implementació de patrons de disseny, automatització de proves unitàries, entre altres, han sigut essencials per desenvolupar aquesta aplicació.

Durant el recorregut acadèmic, les assignatures cursades han estat vitals per a la formació necessària per afrontar els reptes del desenvolupament de ConnecTFG. Des de les classes d'enginyeria del software (ISW) fins a les de disseny de software (DDS) i gestió de projectes àgils (PIN, PSW), cada assignatura ha contribuït a adquirir les competències requerides per a la creació d'aquesta plataforma.

A més dels coneixements tècnics, el grau en Enginyeria Informàtica també ha fomentat la capacitat de treball col·lectiu, a través d'una gran quantitat de treballs en equip. També, l'ús de la metodologia de desenvolupament Kanban ens ha permès organitzar l'avanç del projecte eficientment.

En conclusió, ConnecTFG és el fruit de la conjunció entre l'aplicació pràctica dels coneixements adquirits en els estudis d'Enginyeria Informàtica i la capacitat de treballar en equip.

## 7.3 Treball futur

---

Malgrat els èxits aconseguits fins ara, ConnecTFG té un ampli marge per a futures millores i desenvolupaments. A continuació, es presenten algunes de les funcionalitats i aspectes que es consideren importants per continuar millorant la plataforma i l'experiència d'usuari:

- Possibilitat de realitzar videotrucades dins del xat.
- Ampliació de l'ús de la plataforma a estudiants d'altres universitats.
- Addició del rol de professor per a tutoritzar idees.
- Abordar les respostes obtingudes a través de l'enquesta proposada per millorar diversos aspectes de ConnecTFG.

- Dissenyar la interfície per ser utilitzable des dels dispositius mòbils.
- Reduir la sobrecàrrega d'informació en el *feed* de notícies per a proporcionar una experiència més agradable i eficient per als usuaris.

Aquestes millores i el desenvolupament continu de ConnectTFG permetran que aquesta eina siga encara més valuosa per als estudiants de qualsevol universitat, millorant la qualitat dels seus Treballs de Fi de Grau i preparant-los millor per al seu futur professional.



# Bibliografía

---

- [1] Infoautonomos, “Análisis DAFO: qué es y ejemplo,” Infoautonomos. <https://www.infoautonomos.com/plan-de-negocio/analisis-dafo/> (accedit 20 Agost, 2023)
- [2] (n.d), “Arquitectura Cliente-Servidor”, Reactive Programming. <https://reactiveprogramming.io/blog/es/estilos-arquitectonicos/cliente-servidor> (accedit 20 Juny, 2023)
- [3] (n.d), “Nodejs - Tutorial Frontend - Backend”, Tutorial Frontend - Backend. <https://ccsw-csd.github.io/tutorial/site/develop/basic/nodejs/> (accedit 20 Juny, 2023)
- [4] C. Fernández, “¿Qué es una Single Page Application o SPA?,” ABA-Mobile, Març, 2023, [Online]. Disponible: <https://abamobile.com/web/que-es-single-page-application-spa/>
- [5] M. Coppola, “¿Qué es Angular? Características y ventajas,” Hubspot, Gener, 2023. [Online]. Disponible: <https://blog.hubspot.es/website/que-es-angular>
- [6] M. J. Gonçalves, “¿Qué es Angular y para qué sirve? - Blog de Hiberus Tecnología”, Blog De Hiberus Tecnología. <https://www.hiberus.com/crecemos-contigo/que-es-angular-y-para-que-sirve/> (accedit 3 Juliol, 2023)
- [7] Rootstack, “Para qué usar las directivas incorporadas en Angular”, Rootstack. <https://rootstack.com/es/blog/para-que-usar-las-directivas-incorporadas-en-angular> (accedit 3 Juliol, 2023)
- [8] (n.d), “Qué es HTML,” DesarrolloWeb.com. <https://desarrolloweb.com/articulos/que-es-html.html> (accedit 5 Juliol, 2023)

- [9] “Conceptos básicos de HTML - Aprende desarrollo web | MDN”, MDN. [https://developer.mozilla.org/es/docs/Learn/Getting\\_started\\_with\\_the\\_web/HTML\\_basics](https://developer.mozilla.org/es/docs/Learn/Getting_started_with_the_web/HTML_basics) (accedit 5 Juliol, 2023)
- [10] Admin, “DOMINICODE.COM”, DOMINICODE.COM. <https://dominicode.com/diferencias-entre-css-vs-scss/> (accedit 6 Juliol, 2023)
- [11] Yumpu.com, “MANUAL DE HTML CSS,” yumpu.com. <https://www.yumpu.com/es/document/view/21683545/manual-de-html-css> (accedit 3 Setembre, 2023)
- [12] (n.d), “Acerca | Node.js”, Node.js. <https://nodejs.org/es/about> (accedit 10 Juliol, 2023)
- [13] J. Lucas, “Qué es NodeJS y para qué sirve”, OpenWebinars.net. <https://openwebinars.net/blog/que-es-nodejs/> (accedit 10 Juliol, 2023)
- [14] Kinsta, “¿Qué es Express.js? Todo lo que Debes Saber,” Kinsta. <https://kinsta.com/es/base-de-conocimiento/que-es-express/> (accedit 11 Juliol, 2023)
- [15] C. Merchán, “Introducción a socket.io,” OpenWebinars.net. <https://openwebinars.net/blog/introduccion-a-socketio/> (accedit 15 Juliol, 2023)
- [16] A. Roldan, “Socket.io: el motor en tiempo real más rápido y fiable para Javascript - Blog Clicko,” Blog Clicko. <https://blog.clicko.es/socket-io-el-motor-en-tiempo-real-mas-rapido-y-fiable-para-javascript/> (accedit 15 Juliol, 2023)
- [17] T. Rédac, “MongoDB: todo sobre la base de datos NoSQL orientada a documentos,” Formation Data Science | DataScientest.com. <https://datascientest.com/es/mongodb-todo-sobre-la-base-de-datos-nosql-orientada-a-documentos> (accedit 15 Juliol, 2023)
- [18] Y. Fernández, “Qué es Github y qué es lo que le ofrece a los desarrolladores,” Xataka, Octubre 2019, [Online]. Disponible: <https://www.xataka.com/basics/que-github-que-que-le-ofrece-a-desarrolladores>
- [19] P. Letelier, “Herramientas para gestión ágil de proyectos de desarrollo de software”, Agility at work. <http://agilismoatwork.blogspot.com/2012/01/herramientas-para-gestion-agil-de.html> (accedit 25 Juliol, 2023)



- [20] (n.d), "Modelado del Dominio," DIAGRAMAS UML. <http://diagramasumlerickolmososati102.weebly.com/modelado-del-dominio.html> (accedit 16 Juliol, 2023)
- [21] (n.d), "Observer", Refactoring Guru. <https://refactoring.guru/es/design-patterns/observer> (accedit 20 Juliol, 2023)
- [22] Digital, "Cómo usar Testing en Angular con Jasmine y Karma," DIGITAL55. <https://digital55.com/blog/como-usar-testing-angular-jasmine-karma/> (accedit 15 Agost, 2023)
- [23] N. Avila, "Aplicaciones multilenguaje en Angular 7 con ngx-translate," Medium, Diciembre, 2021. [Online]. Disponible: <https://medium.com/angular-chile/aplicaciones-multilenguaje-en-angular-7-con-ngx-translate-db8d1e7b380c>
- [24] EcuRed, "Despliegue de Software - EcuRed", EcuRed. [https://www.ecured.cu/Despliegue\\_de\\_Software](https://www.ecured.cu/Despliegue_de_Software) (accedit 20 Agost, 2023)
- [25] V. Unir, "¿Qué es el despliegue de aplicaciones web?", UNIR, Diciembre, 2022. [Online]. Disponible: <https://www.unir.net/ingenieria/revista/despliegue-aplicaciones-web/>
- [26] (n.d), "Cloud Application Hosting for Developers | Render", Render. <https://render.com/> (accedit 20 Agost, 2023)
- [27] (n.d), "¿Qué son los Objetivos de Desarrollo Sostenible?", Sostenibilidad. <https://www.sostenibilidad.com/desarrollo-sostenible/que-son-los-objetivos-de-desarrollo-sostenible/> (accedit 29 Agost, 2023)



---

---

# APÈNDIX A

## Descripció de les UT

---

En aquest annex es descriuen en detall el conjunt de funcionalitats desenvolupades al llarg d'aquests 3 *sprints*, centrant-nos amb les que conformen el sistema de missatgeria i notificacions.

### A.1 *Sprint 1*

---

Ordre	Identificador d'UT	Nom	Estimació (en hores)
5	316	<i>Toolbar</i>	3
15	255	<i>Crear grupo</i>	7
25	224	<i>Añadir participantes al grupo</i>	7
35	147	<i>Enviar mensaje</i>	12
45	146	<i>Recibir mensaje</i>	6
55	223	<i>Cambiar nombre del grupo</i>	3
65	225	<i>Añadir descripción al grupo</i>	3
75	241	<i>Indicador de estado del usuario</i>	10

#### A.1.1. *Toolbar*

Aquesta UT representa el desenvolupament de la barra de navegació que hi ha en la part superior de la pantalla. Esta serveix per a navegar entre el *feed*, el xat, les notificacions, etc.

#### A.1.2. *Cambiar nombre del grupo*

Esta UT li permet qualsevol membre d'un grup canviar-li el nom en cas que ho desitge. Esta funcionalitat es va estimar en 3 hores.

### A.1.3. *Añadir descripción al grupo*

Aquesta UT permet afegir una descripció a un grup a l'hora de crear-lo. La descripció té la finalitat de proporcionar informació bàsica i rellevant sobre el propòsit i objectius del TFG plantejat.

### A.1.4. *Indicador de estado del usuario*

Aquesta funcionalitat permet saber en quin estat es troba qualsevol usuari de la xarxa. Per tal de saber-ho hi hauria que anar al perfil de l'usuari i veure si ix un cercle verd (connectat) o roig (desconnectat).

## A.2 *Sprint 2*

---

Ordre	Identificador d'UT	Nom	Estimació (en hores)
5	256	<i>Aceptar candidato</i>	10
15	257	<i>Rechazar candidato</i>	4
25	145	<i>Recibir notificaciones</i>	20
35	144	<i>Eliminar notificación</i>	4
45	246	<i>Eliminar participantes del grupo</i>	7
55	247	<i>Eliminar grupo</i>	4
65	245	<i>Editar descripción del grupo</i>	3

### A.2.1. *Eliminar participantes del grupo*

Aquesta UT permet eliminar membres d'un grup, sent útil en cas que a algun usuari ja no li interesse la idea o qualsevol altre motiu. No es pot eliminar del grup al creador del grup.

### A.2.2. *Eliminar grupo*

Esta UT permet eliminar un grup, en el cas de ser el creador. Aquesta funcionalitat és útil en cas que finalment no vaja a produir-se la idea de TFG.

### A.2.3. *Editar descripci3n del grup*

Aquesta UT permet que qualsevol membre del grup edite la descripci3n, per tal de donar-li un nou context o qualsevol altre motiu.

## A.3 Sprint 3

Ordre	Identificador d'UT	Nom	Estimaci3n (en hores)
5	361	<i>Rediseño de la interfaz (m3s intuitiva)</i>	5
15	365	<i>Indicar autor del mensaje</i>	7
25	366	<i>Diferenciar los mensajes del usuario de los otros</i>	3
35	362	<i>Visualizaci3n de las conexiones actuales</i>	10
45	239	<i>Enviar archivos en el chat</i>	14
55	162	<i>Internacionalizaci3n</i>	15
65	364	<i>Desplegar la aplicaci3n web</i>	3
75	363	<i>Encuesta usabilidad y an3lisis de resultados</i>	5

### A.3.1. *Rediseño de la interfaz (m3s intuitiva)*

En aquesta UT es va redissenyar la *toolbar* i el flux de navegaci3n. Anteriorment, per tal d'arribar al xat, s'havia de passar per la finestra de connexions. Aç3 s'ha sol·lucionat en aquesta 3ltima versi3n, per tal de donar una millor experi3ncia d'usuari al tenir el xat m3s a mà.

### A.3.2. *Indicar autor del mensaje*

Aquesta UT serveix per a, en el cas de grups amb diverses persones, que es sàpiga qui envia cada missatge i poder respondre-li dirigint-se a ell.

### A.3.3. *Diferenciar los mensajes del usuario de los otros*

En un principi tots els missatges es mostraven a la part dreta del xat, sense diferenciar els missatges de l'usuari dels altres. Aç3 s'ha sol·lucionat en aquesta versi3n, estant els missatges de l'usuari a la dreta i els de la resta de membres a l'esquerra.

#### **A.3.4. *Visualización de las conexiones actuales***

S'ha redissenyat la finestra de connexions per a que, a més de mostrar les sol·licituds de connexions, mostrar també les connexions actuals de l'usuari, permetent iniciar o continuar una conversa des d'ahi mateixa.

#### **A.3.5. *Desplegar la aplicació web***

Tal com s'ha explicat en el capítol 6, una vegada implementats tots els canvis es va procedir a desplegar tant el *frontend* com el *backend* per tal que els usuaris finals pogueren utilitzar l'aplicació.

#### **A.3.6. *Encuesta de usabilidad y análisis de resultados***

Una vegada desplegada l'aplicació es va procedir a realitzar una enquesta de Google Forms amb la intenció de rebre *feedback* per part de potencials usuaris. Aquests resultats van ser analitzats i seran emprats per a realitzar millores en els futurs *sprints*.

---

## APÈNDIX B

### Sobre ODS

---

Grau de relació del treball amb els Objectius de Desenvolupaments Sostenible (ODS)

<b>Objectius de Desenvolupament Sostenible</b>	<b>Alt</b>	<b>Mitjà</b>	<b>Baix</b>	<b>No procedeix</b>
ODS 1. <b>Fi de la pobresa.</b>				X
ODS 2. <b>Fam zero.</b>				X
ODS 3. <b>Salut i Benestar.</b>				X
ODS 4. <b>Educació de qualitat.</b>		X		
ODS 5. <b>Igualtat de gènere.</b>			X	
ODS 6. <b>Aigua neta i sanejament.</b>				X
ODS 7. <b>Energia assequible i no contaminant.</b>				X
ODS 8. <b>Ocupació decent i creixement econòmic.</b>				X
ODS 9. <b>Indústria, innovació e infraestructures.</b>		X		
ODS 10. <b>Reducció de les desigualtats.</b>			X	
ODS 11. <b>Ciutats i comunitats sostenibles.</b>				X
ODS 13. <b>Acció pel clima.</b>				X
ODS 14. <b>Vida submarina.</b>				X
ODS 15. <b>Vida d'ecosistemes terrestres.</b>				X
ODS 16. <b>Pau, justícia e institucions sòlides.</b>				X
ODS 17. <b>Aliances per aconseguir objectius.</b>		X		

Reflexió sobre la relació del TFG amb les ODS i amb les ODS més relacionades.

Les Nacions Unides han establert una sèrie d'Objectius de Desenvolupament Sostenible (ODS) [27], una agenda global amb 17 objectius interconnectats que busquen abordar els reptes més urgents que enfronta el món actual. Aquests ODS ofereixen un marc ampli per a la cooperació internacional i la col·laboració en diferents àrees amb l'objectiu de crear un futur més just, igualitari i sostenible per a tots.

Dins d'aquest context global de desenvolupament sostenible, el projecte ConneCTFG emergeix com una plataforma digital innovadora amb relacions significatives amb diversos ODS. Aquesta plataforma té com a finalitat millorar la col·laboració i l'intercanvi d'idees entre estudiants universitaris durant la creació dels seus TFG.

Específicament, ConneCTFG està intrínsecament lligat a l'Objectiu 4: Educació de Qualitat. Aquesta plataforma digital fomenta la millora de la qualitat educativa mitjançant la creació d'un entorn virtual on els estudiants poden compartir coneixements, intercanviar idees i formar equips multidisciplinaris per millorar els seus TFG. Aquest enfocament contribueix a una educació més rica, interconnectada i diversificada, que pot afavorir la comprensió global i la cooperació enfront dels reptes de la societat actual.

Així mateix, ConneCTFG s'alinea amb l'Objectiu 5: Igualtat de gènere. La plataforma ofereix als estudiants la possibilitat de formar equips sense discriminació de gènere. A més, els projectes ideats en la plataforma poden servir per a conscienciar sobre qüestions relacionades amb la igualtat de gènere.

Un altre ODS rellevant és l'Objectiu 9: Indústria, Innovació i Infraestructura. ConneCTFG encarna la innovació mitjançant l'ús de tecnologies digitals per connectar estudiants i crear un espai per a la creació col·laborativa de coneixement. Aquesta plataforma és una manifestació pràctica de com la tecnologia pot ser utilitzada per millorar l'educació i fomentar la interacció multidisciplinària, enfortint així els fonaments de la indústria i la innovació.

La plataforma també comparteix vincles amb l'Objectiu 10: Reducció de les Desigualtats. ConneCTFG té la capacitat d'ampliar el seu abast més enllà de les fronteres universitàries, permetent la participació de estudiants d'altres institucions acadèmiques. Aquesta ampliació pot contribuir a reduir les desigualtats en l'accés a recursos educatius i oportunitats, promovent així la inclusió i la diversitat.

A més, l'Objectiu 17: Aliances per a aconseguir els objectius, és central per ConneCTFG. La plataforma crea un espai on els estudiants poden formar aliances



i col·laboracions significatives. Aquesta relació interconnectada poden formar aliances i col·laboracions significatives. Aquesta relació interconnectada reflecteix la naturalesa del projecte com a eina de connectivitat i col·laboració, que busca aconseguir metes compartides en l'educació i la millora dels TFGs.

Si bé ConnecTFG té una relació més directa amb alguns ODS, com s'ha esmentat, és important reconèixer que la seua influència també es pot percebre en altres àmbits. A través de la seua capacitat per fomentar la comunicació global, l'intercanvi de coneixements i la creació de comunitats en línia, el projecte contribueix de manera indirecta a la promoció de la pau, la justícia, la igualtat de gènere i la creació de comunitats sostenibles.

En resum, ConnecTFG es converteix en un exemple d'innovació educativa que se suma als esforços globals per a l'assoliment dels Objectius de Desenvolupament Sostenible. La plataforma no només reforça les relacions educatives i professionals dels estudiants, que també ajuda a construir un futur més inclusiu, sostenible i connectat a través de la col·laboració, la tecnologia i l'aprenentatge col·lectiu.