

---

# AMPLIACIÓ DE CAPACITATS DE LA XARXA GARLANET

---

**Autor:** Marc Blanca Tabeni

**Director:** Joan Manuel Marquès

**Ponent:** Felix Freitag

**Tutor:** Joaquim Deulofeu Aymar

Especialitat **Tecnologies de la Informació**

Facultat d'Informàtica de Barcelona (**FIB**)

Universitat Politècnica de Catalunya (**UPC**)

curs 2019-20 Q2

UNIVERSITAT POLITÈCNICA DE CATALUNYA  
BARCELONATECH

Facultat d'Informàtica de Barcelona





# Índex

<b>0. Resum introductori</b>	<b>5</b>
<b>1. Formulació del problema</b>	<b>7</b>
1.1. Situació global . . . . .	7
1.2. Context del projecte . . . . .	7
1.3. Projecte en el marc de l'empresa . . . . .	8
1.4. Termes i conceptes de l'estudi . . . . .	9
1.5. Justificació i solució al problema . . . . .	9
1.6. Solucions vigents al mercat . . . . .	10
1.7. Per a qui és Garlanet . . . . .	12
<b>2. Planificació i objectius</b>	<b>12</b>
2.1. Inicial . . . . .	14
2.2. Final . . . . .	18
2.3. Valoració del projecte . . . . .	21
2.4. Control de gestió . . . . .	25
2.5. Objectius . . . . .	25
<b>3. Metodologia i rigor</b>	<b>26</b>
3.1. Eines i metodologia . . . . .	26
3.1.1. Metodologia . . . . .	26
3.1.2. Eines . . . . .	27
<b>4. Garlanet</b>	<b>29</b>
4.1. Disseny i arquitectura de Garlanet . . . . .	29
4.2. Microserveis . . . . .	31
<b>5. Avaluació del sistema</b>	<b>33</b>
5.1. Components . . . . .	34
5.2. Funcionament . . . . .	36
5.3. Descripció tests . . . . .	39
5.3.1. Usuaris . . . . .	39
5.3.2. Repositoris . . . . .	42
5.4. Execució de tests . . . . .	43
5.5. Resultats tests . . . . .	47

5.5.1. Local . . . . .	49
5.5.2. Remot . . . . .	55
<b>6. Ampliació</b>	<b>61</b>
6.1. Descripció implementació . . . . .	61
6.2. Tests de funcionament . . . . .	66
6.2.1. Sessions de consistència . . . . .	66
<b>7. Sostenibilitat i compromís social</b>	<b>68</b>
7.1. Dimensió Ambiental . . . . .	69
7.2. Dimensió Econòmica . . . . .	70
7.3. Dimensió Social . . . . .	71
7.4. Reflexió personal . . . . .	72
<b>8. Conclusions</b>	<b>72</b>
8.1. Tests . . . . .	72
8.2. Millores . . . . .	73
8.3. Personal . . . . .	74
<b>Bibliografia</b>	<b>75</b>

## 0. RESUM INTRODUCTORI

Generació de diferents escenaris per avaluar el funcionament i el comportament del sistema Garlanet, una xarxa social de tipus microblogging, s'han definit diferents tipus de comportaments, més agressius i més passius amb el sistema per veure com afecten aquests perfils al conjunt de components que formen la xarxa.

Per una altra banda, Garlanet també vol implementar la capacitat de xat, tant la comunicació un-a-un com la comunicació molts-a-molts, en aquest projecte es mostra una part que es va implementar que és necessària per poder oferir també aquesta funcionalitat de xat.

Primer de tot veurem la justificació del TFG, per això s'ha de justificar Garlanet sencer donat que el projecte és una avaluació hem que justificar que el que estem avaluant té sentit i és una solució interessant, per tant farem un context mundial, i anirem acotant fins a trobar-nos en el context on se situa Garlanet i veurem solucions semblants.

Una vegada justificat Garlanet, veurem en què consisteix el TFG, la planificació del projecte i el que es vol aconseguir amb aquest, també es veurà quines eines s'empren per complir aquests objectius i la manera amb la qual es treballarà.

Una vegada tinguem clar el que es vol fer, veurem una estructura general de Garlanet, per així poder entendre just després en què consisteix l'avaluació del sistema que s'ha fet i la posterior ampliació de funcionalitats.

Per últim es farà un anàlisi de sostenibilitat i unes reflexions del TFG i personals.

Testing of different scenario to evaluate the functioning and behaviour of the Garlanet system, a social network, it's been defined different types of behaviours, aggressive and passive with the system to observe how the system responds and behaves against them.

On other hand, Garlanet needs a chat-type implementation, either one-on-one chat and also group chat, here its been implemented just a portion of the implementation to extend the capacities of Garlanet towards chat implementation.

First of all we will see the TFG justification, to do that we need to justificate Garlanet as a

good solution, to prove that the evaluation of it its worth to do. I'll start giving a bigger picture of the situation and then approaching to Garlanet environment.

Once that's been done, we are going to scope-in to the TFG, the planification, what the objectives are, and the tools, methods that it's been use to archive them.

Finally, there will be a sustainability analysis and reflections from the work done and personal opinion.

## **1. FORMULACIÓ DEL PROBLEMA**

### **1.1. Situació global**

Actualment és un fet mundial l'expansió de l'ús de les xarxes socials i comporta una de les activitats diàries de l'ésser humà allà on hi ha accés a les tecnologies per poder fer ús d'aquestes. Xarxes com Facebook, Twitter, Instagram... Que són les més conegudes semblen imbatibles en el mercat amb el complet domini quant a microblogging. Encara que hi existeixen altres aplicacions com Whatsapp o Telegram, les xarxes socials mencionades també incorporen la funcionalitat de xat, ja que a partir d'un comentari publicat a la xarxa de microblogging pot derivar a una conversació on és més còmode portar-la a terme en un servei de xat que no pas en un de microblogging.

Per tant ens trobem en un escenari on hi ha grans potències que difícilment es poden tombar, ja que la gent, els usuaris, d'aquestes xarxes socials hi són acostumats al funcionament d'aquestes xarxes. Per tant, perquè invertir en el projecte de Garlanet si el mercat de les xarxes socials està dominat per aquesta competència? Veiem els punts interessants.

És un fet que donem molta més informació de la que creiem a les nostres xarxes socials i que informació com a tal no és únicament la foto que pugem o el missatge on diem que hem estat el cap de setmana de viatge, hi ha molta més informació a part d'aquesta que empreses com les que hem vist a l'apartat anterior recaptin i generen informació d'alta privacitat que es venuda a empreses de tercers interessats en la nostra activitat.

Per tant el problema a resoldre que veiem és la centralització dels serveis tenint la capacitat de poder canviar els termes i condicions del sistema o tenir el control sobre el servei que donen i el funcionament de les màquines, com per exemple podent apagar-les i deixant sense servei a tothom.

### **1.2. Context del projecte**

Per poder parlar sobre el projecte és imprescindible descriure l'escenari on ens trobem per poder comprendre idees proposades i decisions preses tenint en compte els diferents factors que defineixen aquest escenari. En aquest cas encara s'ha de remarcar més aquesta idea, ja que és el cas d'una ampliació de característiques d'un projecte i per tant abans de parlar

sobre el mateix projecte veurem la base i el context on ens trobem.

### 1.3. Projecte en el marc de l'empresa

L'empresa on s'està portant a terme el projecte anomenat Garlanet[1] és a la Universitat Oberta de Catalunya (UOC)[2], com a universitat tenen molts projectes de diferents disciplines, entre aquest mar de projectes en la disciplina de Tecnologia i comunicació es troba Garlanet. Garlanet és un projecte que està sent desenvolupat per membres de dos grups de recerca que són ICSO[3] i KISON[4] de la UOC.

*Internet Computing Systems Optimization*(ICSO) es centra en dues línies de recerca interconnectades:

1. L'ús d'algorismes i solucions de programari per a assistir en la presa de decisions complexes en les àrees del transport i la logística, la producció, el posicionament a temps real i les ciutats intel·ligents.
2. El desenvolupament de sistemes informàtics distribuïts i paral·lels a diferents escales (grups petits, clústers o internet), a més del disseny de sistemes i aplicacions col·laboratius.

*K-riptography and Information Security for Open Networks* (KISON) per una altra banda amb l'aparició de nous serveis com localització de moviment fins a transaccions econòmiques electròniques ha portat a creació de noves tecnologies per millorar la seguretat dels usuaris i de les empreses. KISON es centra en dues línies:

1. Seguretat i privacitat de les xarxes obertes: Aquesta línia de recerca aborda el desenvolupament de mecanismes més adequats per a millorar la seguretat i la privacitat de diversos àmbits, com les xarxes d'igual a igual, les xarxes ad hoc, les xarxes RFID i de sensors, i les xarxes socials en línia, entre d'altres.
2. Seguretat i privacitat dels continguts multimèdia: Aquesta recerca consisteix a dissenyar tècniques adequades per a protegir la informació multimèdia que es transmet en xarxes obertes, en relació amb la privacitat dels usuaris d'aquests continguts.



## 1.4. Termes i conceptes de l'estudi

Ja feta la contextualització del projecte i definit l'ecosistema on es desenvolupa, es definiran un seguit de termes i conceptes que es creuen necessaris que defineixen punts importants en el projecte i que és fonamental conèixer el seu significat.

### ■ **Microbloging**

Microbloging és un terme que s'utilitza per definir un tipus de comunicació entre usuaris d'una xarxa amb missatges curts[5]: un usuari A publica un missatge i els únics usuaris de tota la xarxa que podran veure aquests missatges que l'usuari A pugui seran els que A hagi autoritzat per veure mitjançant el protocol que la xarxa implementi si és el cas de tenir un perfil privat, o per defecte qualsevol usuari de la xarxa podrà veure aquests missatges, en cas que el perfil de l'usuari A sigui públic.

### ■ **Comunitat**

Segons el Diccionari de la llengua Catalana[6] de l'Institut d'Estudis Catalans:  
Comunitat: *f* [LC] [AN] [SO] Grup social que es caracteritza generalment per un vincle territorial i de convivència o per una afinitat d'interessos i de conviccions ideològiques. En aquest cas Garlanet és format per aquesta comunitat amb afinitat d'interessos que com veurem un dels punts més característics de la xarxa.

### ■ **Seguretat**

Quan es menciona el terme seguretat en informàtica es pot aplicar a moltes disciplines i portada a terme amb diferents mètodes. En aquest cas es parla de la seguretat de la informació, referint-se a la privadesa d'aquesta permetent l'accés a aquesta únicament als agents que tenen permís per poder accedir a ella.

## 1.5. Justificació i solució al problema

Com hem vist la centralització i la privadesa són dos aspectes que són bastant qüestionables en les xarxes socials competidores i que per tant són els dos pilars sobre els quals Garlanet es manté i els tractarem en aquest punt.

Respecte al primer dels aspectes, la centralització, com a resposta això parlem directament dels sistemes descentralitzats. El gran avantatge que tenen aquests sistemes és la dificultat de poder fer el seguiment a les dades que circulen per la xarxa, permetent a cada usuari, o participant d'aquesta comunitat, crear el seu espai i unir-lo amb la resta de la comunitat. És probable que molts d'aquests usuaris no el puguin tenir sempre en funcionament per tant

hauran de registrar-se en servidors oferts per altres persones havent-hi acceptat els termes i condicions d'aquests servidors, el problema de seguretat hi segueix sent però és menor que en el cas del servidor centralitzat, ja que no depenem d'un únic servei però el que tenim és una infinitat de termes i condicions d'ús.

L'altre punt a tractar és la privadesa, en tots dos escenaris, centralitzat o en comunitat, existeix el problema on una persona que vulgui extreure informació pugui connectar-se, en cas del model centralitzat als servidors, directament a aquests o al port d'entrada, escenari que sembla més "fàcil" de donar-se, però això no es resol amb un sistema descentralitzat, ja que aquesta operació que es fa en els sistemes centralitzats es pot repetir a múltiples fonts d'informació i així acabant recol·lectant informació i extreien dades sensibles. Per tant queda clar que s'ha d'establir un model on no es vinculin relacions entre el qui envia el missatge i el que el rep, igual que estadístiques dels usuaris.

Per tant sembla interessant aquesta combinació entre la implementació d'un sistema descentralitzat, on no tenim els problemes d'una centralització, amb l'addició d'un model de seguretat que no vincula als diferents usuaris que es comuniquen.

Seguint amb els punts febles de les xarxes socials actuals, Garlanet es constitueix per aquests mateixos punts fent-los els seus propis punts forts. El que es du a terme és dividir el servei en operacions de poca càrrega que s'executen en un sistema descentralitzat que està format per ordinadors que han sigut contribuïts de manera voluntària a la comunitat per alguns dels participants, per tant l'activitat dels usuaris no pot ser seguida, o almenys és molt més difícil que no pas en un sistema centralitzat. A més, informació privada o dades sobre el nostre perfil, estan guardades de forma segura de manera criptogràfica i és únicament llegible pels usuaris que nosaltres hi autoritzem per tant no hi existeix el problema de donar informació d'un mateix perquè un altre la guardi i pugui accedir-hi a ella il·legítimament.

## 1.6. Solucions vigents al mercat

Ara que hem vist on Garlanet es centra veiem les millors solucions similars que hi ha actualment al mercat.[7]

- Diaspora

Els servidors són independents entre ells i no tenen cap node central, la informació no

es ven a tercers i pots escollir en quin "pod", servidor, l'allotges i escollint amb quins comparteixes aquesta informació.[8]

- Minds

És una xarxa descentralitzada *open-source* que permet monetitzar les contribucions a la xarxa dependent del tipus de contribució.[9]

- Mastodon

La més semblant a Twitter de les mencionades, tipus federació són diferents comunitats que juntes formen la xarxa, amb eines i codis de conducta amb moderadors dispersats per tenir un servei més personal.[10]

- Sola

Una xarxa que implementa una AI, fent que no hagi de seguir a ningú, i que segons interaccions mostra més informació que pot estar relacionada amb aquesta i que pot interessar a l'usuari. Immune a bloqueig i censura a més divideix els ingressos d'anuncis, ingressos d'usuaris i associacions amb tots els usuaris.[11]

- Manyverse

Guarda la informació de l'usuari en el dispositiu propi i sincronitza la informació amb la plataforma Scuttlebutt[12], així permet ser usada sense connexió a la xarxa. La informació es pot sincronitzar directament entre dispositius via Bluetooth. Altres característiques són: sense publicitat, sense centres de dades, sense xarxa, sense cookies, sense possible seguiment de dades, sense costos prèmium i sense notificacions de publicitat.[13]

Vistes les solucions actuals podem dir que Garlanet, encara que comparteix algunes idees i filosofies amb algunes d'aquestes solucions, és una solució prou íntegra per a veure-la com una altra solució al problema de la centralització i a la seguretat de les nostres dades. Per tant Garlanet és un projecte interessant que proposa una filosofia, que tot i que comparteix alguns aspectes amb solucions que ja hi existeixen, té un caràcter propi.

Una vegada s'ha presentat un escenari global, xarxes socials, dins d'aquest món ens hem situat on hi som nosaltres, la UOC, i una vegada avaluat el que ja hi existeix i quines són les diferents característiques d'aquestes propostes, s'ha definit Garlanet.

El TFG és una avaluació d'aquest projecte i per la justificació del d'aquest era important justificar Garlanet com a projecte interessant donat que si no ho fos, la seva avaluació tampoc

ho seria. A més d'aquesta avaluació s'ha fet una ampliació de capacitats, que correspon a la gestió de dades interna per implementar la comunicació un-a-un donat que Garlanet actualment només disposa de la capacitat de microblogging.

### **1.7. Per a qui és Garlanet**

Com ja s'ha definit Garlanet és una xarxa descentralitzada i es formada íntegrament pels usuaris que la conformen, ja que són els que li donen ús a aquesta i també els que proporcionen recursos a la xarxa de manera voluntària.

Pensem que empreses sense ànim de lucre poden estar molt interessades a utilitzar Garlanet, ja que segueixen la mateixa filosofia de voluntarietat i a més sense perdre privacitat.

Per una altra banda creiem que empreses més grans podrien arribar a voler desplegar la seva pròpia xarxa Garlanet, ja que degut a l'enfocament de autoorganització i autoprovisió requereix una estructura mínima i per tant el baix cost per desplegar aquesta estructura és un bon al·licient.

Per últim també creiem que moltes persones particulars i altre tipus d'empreses i comunitats poden estar interessades a formar part de Garlanet.

## **2. PLANIFICACIÓ I OBJECTIUS**

A continuació es farà un llistat de les tasques que conformen el projecte i una descripció d'aquestes, però abans d'això es defineixen unes variables per poder generar un context on situar-les.

El projecte es porta a terme en una empresa, per tant hi ha un contracte el qual té una extensió major que supera el dia de lectura del TFG, aquest temps no s'inclou al grup de tasques, ja que encara que serà temps dedicat al projecte no entra en el període, però s'ha volgut mencionar, definit entre la data d'inici i la data de finalització, el dia de lectura.

La data d'inici del projecte per una altra banda sí que és la mateixa que el començament del contracte amb l'empresa i és fitada el 23 d'octubre de 2019, la data de finalització a l'estar

previst fer la defensa del projecte al torn de Juny-Juliol ens posem en el pitjor cas i establim el primer dia d'aquest torn com a data de finalització, 26 de juny de 2020.

Per tant passen 250 dies que només tenint en compte els dies lectius, encara que alguns caps de setmana s'hagi treballat no es tenen en compte, ja que aquestes hores és recol·loquen en l'horari entre setmana, retallem fins a 150 dies.

En dividim el temps de treball en dues parts, un primer interval de temps que comprèn els primers 41 dies on només es dediquen 3 hores al dia i l'interval restant de temps, 139 dies, 7 hores, on queden incloses les hores dels caps de setmana, per tant tenim un resultat de 41 dies a 3 hores per dia i 117 dies a 7 hores per dia el que resulta en 942 hores.

Abans de començar a definir els diferents Sprints que componen el projecte es vol fer una menció a la tasca de Gestió del Projecte que compren diferents subtasques que algunes només es donen durant un temps i altres que es van repetint al llarg del projecte. La documentació del treball fet i les decisions preses, reunions amb el director del projecte sobre documentació i termes de les diferents iteracions són tasques que es desenvolupen al llarg del projecte, i per una altra banda, l'abast del projecte, planificació del projecte al llarg del temps i altres temes com el pressupost del projecte i el projecte dins del marc de la sostenibilitat són tasques que es duen a terme una única vegada.

El que fem és dividir el projecte en iteracions/sprints i cada sprint està format per un conjunt de tasques que també segueixen la metodologia iterativa i per tant són també input de la metodologia Agile: que compon les tasques d'1. Documentació (que volem fer), 2. Disseny (com ho fem), 3. Reunió (comprovació tasca 1 i 2), 4. Implementació, 5. Tests (que funcioni i el comportament sigui l'esperat).

Per tant tenim que el projecte es divideix en dues branques, independents una de l'altre, la primera és una avaluació de comportament del sistema i una segona part d'ampliació de capacitats. Ara es presenta les dues planificacions, una inicial que es va fer abans de començar el treball i una final amb una aproximació del que s'ha fet.

## 2.1. Inicial

A continuació es definiran els diferents sprints i les tasques que formen cadascun d'aquests:

### ■ Sprint-0

Abans de començar amb els objectius es defineix una iteració en la qual es dur a terme un primer estudi de l'estat actual del projecte i es fan unes proves, molt primitives per veure el flux de dades entre els diferents components del projecte i així tenir una idea del projecte a l'hora de poder començar a fer iteracions. Per aquesta iteració s'estima una inversió de 10 hores.

### ■ Sprint-1

Al primer Sprint es porta a terme la generació de comportaments d'usuaris segons un estudi i també es genera el comportament de les màquines que es donen voluntàriament, ja que som conscients que no podem assumir que estaran sempre en funcionament.

**S1-T0.** Estudi sobre distribucions i lògica de: quantitat i moments d'enviament de missatges, quantitat de seguidors i seguits i altres factors de comportament de l'usuari secundaris.

**S1-T1.** Els usuaris tenen la capacitat de registrar-se i posterior login i de seguir a altres.

**S1-T2.** Els usuaris poden enviar missatges i rebre els missatges únicament de les persones que segueixen.

**S1-T3.** Els usuaris són capaços de poder adjuntar fitxers als missatges.

**S1-T4.** Els usuaris interaccionen amb els fitxers que hi són adjunts als missatges rebuts.

**S1-T5.** Distribucions que generen el comportament de les màquines, que està definit per, probabilitat d'apagar-se i probabilitat de tornar a estar en funcionament.

### ■ Sprint-2

El segon sprint compren la generació de scripts i posada en marxa del sistema de microblogging amb els comportaments generats.

**S2-T0.** Estudi sobre els scripts i els diferents components que hem de posar en marxa.

**S2-T1.** Generació dels scripts per poder posar en marxa els tests.

**S2-T2.** Posada en marxa els tests i abstracció de dades i cerca de possibles errors.

### ■ Sprint-3

Aquest sprint està encara per definir, ja que cal encara fer un estudi sobre com implementar el xat a sobre del sistema ja existent per tal de fer els menors canvis en l'estructura actual i aprofitar el màxim el funcionament del sistema que ja hi és implementat.

**S3-T1.** Establir paràmetres que ens permetin definir les relacions entre persones que volen comunicar-se.

**S3-T2.** Modificacions al sistema actual de Garlanet per incloure el servei de xat aprofitant el màxim possible l'estructura actual del sistema.

**S3-T3.** Definir l'API que des de l'usuari ens permeti comunicar-nos amb el sistema per poder dur a terme el comportament definit.

**S3-T4.** L'usuari és capaç de poder establir una comunicació un-a-un (conversació user-user).

**S3-T5.** L'usuari és capaç de poder establir una comunicació molts-a-molts (conversació en grup).

### ■ Sprint-4

A partir de les funcionalitats que hem establert al split anterior, en definim un comportament de l'usuari que ens permeti estudiar les diferents funcionalitats del servei xat per poder comprovar el funcionament i estudiar el comportament del sistema en diferents situacions.

**S4-T1.** Fer que l'usuari generi els diferents paràmetres necessaris per poder establir relacions un-a-un.

**S4-T2.** L'usuari és capaç de poder tenir una conversació un-a-un.

**S4-T3.** Fer que l'usuari generi els diferents paràmetres necessaris per poder establir relacions molts-a-molts.

**S4-T4.** L'usuari és capaç de poder tenir una conversació molts-a-molts.

### ■ Sprint-5

El cinquè sprint consisteix en la generació de scripts que en gran part aprofitarem dels anteriors generats, ja que els components del sistema són els mateixos i posada en

marxa el sistema únicament de xat.

**S5-T0.** Estudi sobre els scripts i els diferents components que hem de posar en marxa.

**S5-T1.** Generació dels scripts per poder posar en marxa els tests.

**S5-T2.** Posada en marxa els tests i abstracció de dades i cerca de possibles errors.

#### ■ Sprint-6

Com a últim split es generen diferents tests que posen en marxa tant el servei de micro-blogging com el de xat.

**S6-T1.** Generació dels scripts per poder posar en marxa el sistema sencer.

**S6-T2.** Posada en marxa els tests i abstracció de dades.

	codi	Nom descriptiu	Temps estimat (h)	Dependències	Recursos
	DOC	Documentació i reunions	250		Ordinador i paper
<b>SPRINT0</b>			<b>18</b>		
	S0-T1	Contexte	18	-	Eclipse
<b>SPRINT1</b>			<b>220</b>		
	S1-T0	Estudi comportaments	40	T0 < T2,T3,T4,T5	PDF,Internet,paper i boli
	S1-T1	User logs	20	T1 < T2,T3,T4	Eclipse,PDF,Internet
	S1-T2	User msgs	50	T2 < T3,T4	Eclipse,PDF,Internet
	S1-T3	Msg amb adjunts	50	T3 < T4	Eclipse,PDF,Internet
	S1-T4	Interacció amb msg	40		Eclipse,PDF,Internet
	S1-T5	Comportament servidors	20		Eclipse,PDF,Internet
<b>SPRINT2</b>			<b>42</b>		
	S2-T0	Estudi components	8	T0 < T1	Eclipse,PDF,Internet, paper i boli
	S2-T1	Gen. scripts	14	T1 < T2	Eclipse,PDF,Internet
	S2-T2	Exec. tests	20		Xarxa local i Guifinet
<b>SPRINT3</b>			<b>200</b>		
	S3-T1	Parametrització	30	T1 < T2,T3	Eclipse,PDF,Internet
	S3-T2	Modificació Garlanet	40	T2 < T4,T5	Eclipse,PDF,Internet
	S3-T3	Definició API	50	T3 < T4,T5	Eclipse,PDF,Internet
	S3-T4	un-a-un conversació	35		Eclipse,PDF,Internet
	S3-T5	molts-a-molts conversació	45		Eclipse,PDF,Internet
<b>SPRINT4</b>			<b>120</b>		
	S4-T1	Gen. paràmetres	20	T1 < T2	Eclipse,PDF,Internet
	S4-T2	Gen. un-a-un	40		Eclipse,PDF,Internet
	S4-T3	Gen. paràmetres	20	T3 < T4	Eclipse,PDF,Internet
	S4-T4	Gen. molts-a-molts	40		Eclipse,PDF,Internet
<b>SPRINT5</b>			<b>42</b>		
	S5-T0	Estudi components	8	T0 < T1	Eclipse,PDF,Internet, paper i boli
	S5-T1	Gen. scripts	14	T1 < T2	Eclipse,PDF,Internet
	S5-T2	Exec. tests	20		Xarxa local i Guifinet
<b>SPRINT6</b>			<b>50</b>		
	S6-T1	Gen. scripts conjunts	10	T1 < T2	Eclipse,PDF,Internet
	S6-T2	Exec. tests	40		Xarxa local i Guifinet

**Figura 1:** Taula tasques inicial.



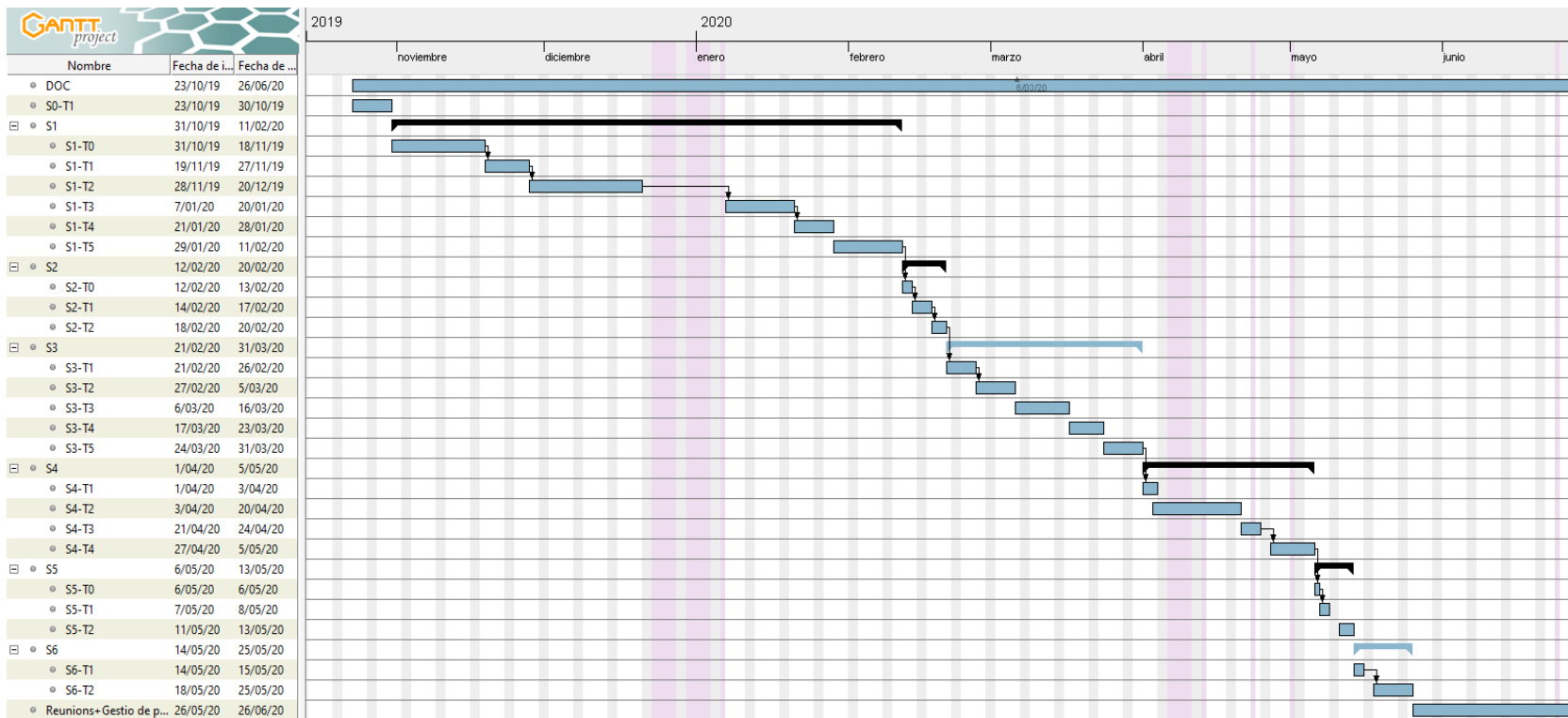


Figura 2: Gantt inicial.

## 2.2. Final

*"La part que pot portar més treball pot ser més imprevisible és el xat de molts-a-molts... En cas que no es pogués finalitzar la implementació del molts-a-molts tindríem el servei un-a-un i fariem les proves únicament amb aquesta implementació... El fet que hi surtin imprevistos per tant com a molt podríem tenir perill únicament amb el servei xat grupal, molts-a-molts."*

Són fragments de la planificació que es va fer a la fita inicial, on es parla sobre els mètodes per adaptar-se a possibles canvis que es podrien haver que fer a la planificació donat el cas que fos necessari, s'ha donat el cas que es necessita fer aquestes modificacions al projecte a conseqüència dels següents motius.

S'ha donat que la magnitud del SPRINT1 era bastant més gran que no pas com s'havia definit donat que quan s'anava desenvolupant es trobaven més comportaments a definir, donant-se el cas fins i tot de poder considerar-se adaptar el projecte a únicament l'estudi del sistema sent robust sense l'ampliació de xat.

Tot i això s'ha volgut implementar una part de la funció de xat un-a-un, encara que la funció molts-a-molts per falta de temps no es podrà portar a terme. Per tant el treball final consisteix en la branca d'avaluació de Garlanet i una part de la implementació una-a-un corresponent a la funcionalitat de xat, que corresponen als SPRINTS 0-2.

S-6 i S-5 queden fora de la planificació donat que són posades en marxa del comportament del xat, com que només s'ha implementat una part d'aquesta funcionalitat només es parlarà dels tests de la part implementada. Per últim els SPRINTS 3-4 queden de la següent manera:

- **Sprint-3**

- S3-T0.** Estudi de l'estructura dels microserveis.

- S3-T1.** Implementació de les diferents parts que componen un microservei.

- S3-T2.** Implementació d'una classe auxiliar per poder fer tests de l'implementat anteriorment.

- **Sprint-4**

- Comprèn els tests per validar la implementació de l'sprint anterior.

- S4-T1.** Tests dels diferents mètodes que componen el microservei.

- S4-T2.** Tests de les sessions de consistència.

	codi	Nom descriptiu	Temps estimat (h)	Dependencies	Recursos
	DOC	Documentació i reunions	250		Ordinador i paper
<b>SPRINT0</b>			<b>20</b>		
	S0-T1	Contexte	20	-	Eclipse
<b>SPRINT1</b>			<b>424</b>		
	S1-T0	Estudi comportaments	80	T0 < T2,T3,T4,T5	PDF,Internet,paper i boli
	S1-T1	User logs	40	T1 < T2,T3,T4	Eclipse,PDF,Internet
	S1-T2	User msgs	160	T2 < T3,T4	Eclipse,PDF,Internet
	S1-T3	Msg amb adjunts	40	T3 < T4	Eclipse,PDF,Internet
	S1-T4	Interacció amb msg	24		Eclipse,PDF,Internet
	S1-T5	Comportament servidors	80		Eclipse,PDF,Internet
<b>SPRINT2</b>			<b>92</b>		
	S2-T0	Estudi components	24	T0 < T1	Eclipse,PDF,Internet, paper i boli
	S2-T1	Gen. scripts	52	T1 < T2	Eclipse,PDF,Internet
	S2-T2	Exec. tests	16		Xarxa local i Guifinet
<b>SPRINT3</b>			<b>108</b>		
	S3-T0	Estudi estructura μserveis	8	T0 < T1,T2	Eclipse,PDF,Internet
	S3-T1	Modificacio Garlanet	40		Eclipse,PDF,Internet
	S3-T2	Implementació μservei	60		Eclipse,PDF,Internet
<b>SPRINT4</b>			<b>48</b>		
	S4-T1	Test mètodes	24	T1 < T2	Eclipse,PDF,Internet
	S4-T2	Test sessió consistència	24		Eclipse,PDF,Internet

**Figura 3:** Taula tasques final.

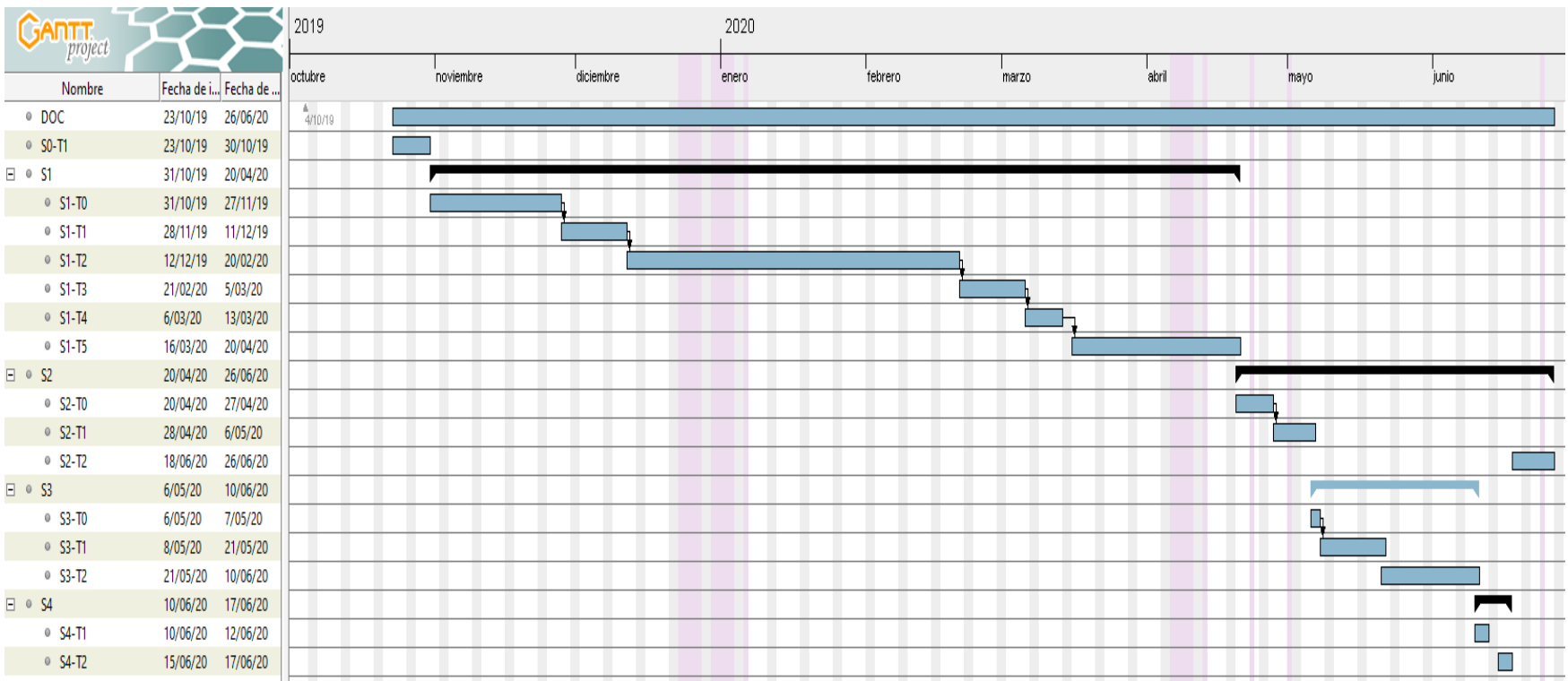


Figura 4: Gantt final.

### 2.3. Valoració del projecte

Els diferents costos els classificarem en tres grups:

#### 1. Costos per activitat (CPA)

Els costos per activitat conformen els costos de personal per cada activitat, el cas de les activitats.

El projecte ha estat desenvolupat únicament per un estudiant en pràctiques amb un sou fix però per fer una estimació més realista dividim els diferents perfils que s'han portat a terme, ja que cadascun d'aquest te un sou diferent que seran aproximacions. El càlcul és fet a partir de sous anuals dividits fins a trobar els €/h per fer més fàcil l'aplicació per cada tasca. S'han estimat jornades de 6 hores/dia amb 24 dies treballats al mes durant 12 mesos.

- Cap de projecte sou de 20 €/h, encarregat de tasques de gestió del projecte.
- Arquitecte de software sou de 22 €/h, encarregat del disseny.
- Desenvolupador sou de 17 €/h, encarregat de la implementació.
- Tester sou de 15 €/h, encarregat de portar a terme els tests.

	codi	Nom descriptiu	Cap de projecte	Arquitecte de software	Desenvolupador	Tester	Total (h)
	DOC	Documentació i reunions	250	-	-	-	250
<b>SPRINT0</b>			<b>250</b>				<b>18</b>
	S0-T1	Contexte	18	-	-	-	18
<b>SPRINT1</b>			<b>18</b>	<b>78</b>	<b>142</b>		<b>220</b>
	S1-T0	Estudi comportaments	-	40	-	-	40
	S1-T1	User logs	-	5	15	-	20
	S1-T2	User msgs	-	10	40	-	50
	S1-T3	Msg amb adjunts	-	10	40	-	50
	S1-T4	Interacció amb msg	-	8	32	-	40
	S1-T5	Comportament servidors	-	5	15	-	20
<b>SPRINT2</b>						<b>42</b>	<b>42</b>
	S2-T0	Estudi components	-	-	-	8	8
	S2-T1	Gen. scripts	-	-	-	14	14
	S2-T2	Exec. tests	-	-	-	20	20
<b>SPRINT3</b>				<b>64</b>	<b>136</b>		<b>200</b>
	S3-T1	Parametrizació	-	30	-	-	30
	S3-T2	Modificació Garlanet	-	8	32	-	40
	S3-T3	Definició API	-	10	40	-	50
	S3-T4	un-a-un conversació	-	7	28	-	35
	S3-T5	molts-a-molts conversació	-	9	36	-	45
<b>SPRINT4</b>				<b>30</b>	<b>90</b>		<b>120</b>
	S4-T1	Gen. paràmetres	-	5	15	-	20
	S4-T2	Gen. un-a-un	-	10	30	-	40
	S4-T3	Gen. paràmetres	-	5	15	-	20
	S4-T4	Gen. molts-a-molts	-	10	30	-	40
<b>SPRINT5</b>						<b>42</b>	<b>42</b>
	S5-T0	Estudi components	-	-	-	8	8
	S5-T1	Gen. scripts	-	-	-	14	14
	S5-T2	Exec. tests	-	-	-	20	20
<b>SPRINT6</b>						<b>50</b>	<b>50</b>
	S6-T1	Gen. scripts conjunts	-	-	-	10	10
	S6-T2	Exec. tests	-	-	-	40	40

**Figura 5:** Taula d'hores dedicades a cada activitat per rol.

L'arquitecte de software té una tasca per cada sprint on es fa una reflexió del sprint en general, però s'ha decidit que de les tasques de desenvolupador extreu unes hores en les quals també es feia d'arquitecte, ja que a l'hora de fer una tasca en concret també es feia el pas previ d'arquitecte. No s'ha fet el mateix amb el tester, ja que el nivell de test és per funció pràcticament i és molt difícil de quantificar. Per tant tenim per cada rol, el sou per hora i el temps de treball el que ens resulta a un preu total de:

Rol	Sou per hora	Temps de treball (h)	Preu total
Cap de projecte	20	268	5360
Arquitecte de software	22	172	3784
Desenvolupador	17	368	6256
Tester	13	134	1742
<b>Total</b>	<b>72</b>	<b>942</b>	<b>17142</b>

**Figura 6:** Càlcul de sou total de cada rol.

Per tant el CPA és de 17.142 €.

## 2. Costos genèrics (CG)

En aquest grup s'agrupen els costos calculats de manera global que tenen presència en totes o algunes de les activitats. En tenim en compte diferents tipus de costos genèrics:

El primer de tot és el software hardware que s'ha utilitzat per portar a terme el projecte. El software que s'ha utilitzat és totalment gratuït, treballant amb Eclipse, Google Drive, Overleaf i Gantt. Per part del hardware s'estima que els dos ordinadors amb els quals es treballarà tant a casa com a l'empresa té un cost d'uns 1500 € incloent els perifèrics necessaris entre les dues màquines.

Els desplegaments per poder fer els tests els fem a Guifinet, com no sabem els costos de les altres màquines el que fem és estimar que totes les màquines que s'utilitzen tenen les mateixes prestacions que la que tenim a l'oficina. Com volem fer tests simulant 1000 usuaris, estimem que cada màquina pot tenir 4 usuaris contem 250 màquines, i pels *repositories* on es guarden les dades estimem que per aquest nombre d'usuaris en necessitem 100 màquines més. Abans de fer els tests en Guifinet els farem en 3 servidors de la UOC, per tant englobem els tests en 354 màquines. Per tant tenim 2 màquines on s'han donat les hores dels rols de cap, arquitecte i desenvolupador i 4 quan estem fent tests.

Aquestes 354 màquines únicament les necessitem per fer els tests, llavors per no disparar el preu el que fem és allotjar-les, amb un cost de 60 €/mes per un servidor dedicat semblant tenim que 0,1 € l'hora, multiplicat pel temps de test són 13,4 € per màquina el que ens dona multiplicat per totes les màquines, 4793 € en total, que sumat a les màquines de l'oficina i de casa són 6.923 € en hardware.

Per últim tenim costos indirectes:

Per una part tenim els costos materials que recull, llapis, goma, bolígraf, fulls i altres elements d'oficina que els recollim en 30 €.

Un altre element és el desplaçament fins a les oficines, que es farà en transport públic, sent un període de 250 dies, són 3 targetes T-JOVE que són 105, 80 i 80 euros. Per tant

265 € en desplaçament.

Per últim ens queda l'espai i l'electricitat consumida. Per tenir un valor de referència, el centre d'empreses Bosc Llarg on dediquen un espai per poder treballar i amb llum, connexió a internet i espai per poder fer reunions, té el cost de 300 €/mes que traduït al període del projecte són 9 mesos, per tant 2.700 €.

En un total de Costos Genèrics tenim  $6923 + 30 + 265 + 2.700 = 9.918$  €

### 3. Contingència (C)

El nivell de contingència és un marge de seguretat del total dels costos de CPA i CG.

En definim un 10% de marge de contingència, per tant en aplicar-se als Costos Per Activitat i als Costos Genèrics ens resulta la suma total de:

$$C = (17.142 + 9.918) * 0,1 = 2.706 \text{ €}$$

### 4. Imprevistos (I)

Per últim agrupem els diferents imprevistos que no estan en el Gantt i que s'afegeixen al cost total segons el percentatge que té de passar.

Els problemes, riscos vistos són casuístiques de fases posteriors a aquesta del projecte per tant no tenen variacions en el pressupost, els obstacles vistos eren desviacions del temps, per tant el que pot passar és que els testos no s'arribin a fer per tant l'allotjament que fem de les màquines no amortitzaríem aquests diners, creiem que la probabilitat que això passi és d'un 5% per tant si perdem els sprints 5 i 6 són 92 hores que a 0,1 €/hora són 9,2 €. Podríem contemplar el fet que l'ordinador s'espatllés, en aquest cas no perdríem cap dada del projecte, ja que tenim el projecte a GitLab, per tant l'únic cost que tenim és el de la pròpia màquina, que creiem que és molt poc probable que passi, un 1%, el que comportaria un afegit de  $650 * 0,01 = 6,5$  €.



Per tant tenim:

$$I = 9,2 + 6,5 = 15,7 \text{ €}$$

Per tant el cost total del projecte sumant els quatre totals és de:

$$\text{TOTAL} = \text{CPA} + \text{CG} + \text{C} + \text{I} = 17.142 + 9.915 + 2.706 + 15,7 = 29.778,7\text{€}$$

## 2.4. Control de gestió

Per poder fer un control el més acurat possible el que es fa és tenir un control sobre les diferents tasques que es porten a terme i una vegada acaba una tasca es té en compte el cost que hem estimat ara i el que realment ha costat, en cas que el segon sigui més alt que el primer haurem de descomptar aquesta diferència dels costos de contingència. Aquest càlcul de cost real que ha acabat costant té en compte tant les hores dedicades (CPA) i una part aproximada dels costos genèrics (CG) a més de sumar algun dels imprevistos en cas que succeeixi algun d'ells.

## 2.5. Objectius

Aquests són els objectius que es van plantejar a la fita inicial del projecte:

1. Generar diferents escenaris per poder testejar i comprovar el correcte funcionament de la xarxa Garlanet oferint el servei de microblogging. Aquests escenaris queden caracteritzats per diferents paràmetres que defineixen els comportaments dels diferents actius que hi participen en la xarxa.
  - a) Queda un subobjectiu clar amb la generació d'aquests diferents escenaris per fer la prova del microservei i és la comprensió del sistema de microblogging, tant per saber com funciona i quina és la filosofia del projecte, tant com per saber com funciona per dins i així tenir una comprensió profunda del projecte.
2. El segon objectiu és la implantació del servei xat a Garlanet. Gràcies a la comprensió adquirida amb el primer objectiu i subobjectiu la idea és implementar el servei de xat no només amb la mateixa filosofia sinó que reaprofitant la quantitat més gran de codi possible, com a bona pràctica.

- a) Com a conseqüència de l'objectiu 2, apareix aquest subobjectiu que podria ser-hi un objectiu per si mateix però en dependre de l'objectiu dos es va decidir definir-lo com a subobjectiu i és proporcionar una GUI a l'usuari per poder utilitzar el servei de xat.

Com ja s'ha esmentat, l'objectiu 2 no s'ha pogut complir íntegrament, donat que només s'ha implementat una part del funcionament a causa de falta de temps donat que l'objectiu 1 va resultar ser molt més robust que no pas es va contemplar al principi del projecte.

Tot i que la idea inicial del projecte era establir l'objectiu 2 com a part principal del projecte, es va trobar amb molt més treball del que s'esperava per poder complir l'objectiu 2, per tant es va fer un replantejament i es fa focalitzar més en realitzar un bon estudi del sistema que no pas en expandir-lo afegint-hi noves funcionalitats.

### **3. METODOLOGIA I RIGOR**

#### **3.1. Eines i metodologia**

##### **3.1.1. Metodologia**

Per dur a terme el projecte s'està seguint una metodologia iterativa la qual té un procés iteratiu on cada iteració està formada per 4 fases:

1. Anàlisi

Com a primera etapa hem de tenir clar que és el que volem o el que necessitem, per tant hem de fer una primera reflexió sobre quin és l'objectiu i organitzar-nos.

2. Disseny

Ja sabent el que volem el següent pas és definir com podem obtenir-ho i modelar una estructura per poder aconseguir-ho tenint en compte els recursos dels quals disposem.

3. Implementació

Un punt tenim clar que és el que volem i una estructura amb un camí de treball definit el que cal és començar a caminar-lo, en aquest punt podem trobar-nos amb errors en la fase de disseny sobretot que ens faran fer un pas enrere i tornar a la fase anterior i acabar de corregir per poder seguir avançant en la fase d'implementació.

#### 4. Testing

Com a fase final per comprovar que el que hem implementat funciona i ho fa com ha de fer-ho, i no hi trobem cap escenari que no hàgim contemplat. En cas que hi tinguem escenaris que no corresponguin amb el que hauria de passar hem de tornar enrere a la fase de disseny o implementació depenent de l'error que hàgim comès.

Igual que el propi projecte Garlanet, en dividim aquest cicle iteratiu el més curt possible per poder identificar els errors que hem comès en les fases de disseny o implementació a més que també en facilita molt la fase de test quan fem parts més curtes de codi no només per elaborar aquests tests sinó que hi tenim menys variables on pot estar l'error.

#### 3.1.2. Eines

##### ■ Sistema Operatiu

Per portar a terme el projecte no hi existia un requisit estricte respecte a l'elecció del sistema operatiu donat que el software amb el qual es treballa no posa restriccions de compatibilitat. Per tant les opcions amb les quals treballa amb més comoditat són Windows i Ubuntu, es porta a terme el projecte en dues màquines, la màquina que se situa a l'oficina es va configurar una arrencada dual de tots dos sistemes operatius per motius externs al projecte però es treballa amb Ubuntu donat que l'arrencada és més ràpida i la màquina personal ja hi tenia instal·lat Windows i com no hi ha problemes de compatibilitat amb el software que s'utilitza per portar a terme el projecte es treballa amb aquest sistema operatiu.

##### ■ Serveis de Google

###### • Gmail

Per poder treballar en un projecte que implica treballar amb altres persones la comunicació és molt important i es troba a totes les fases aquesta necessitat de comunicació, com a eina de comunicació quan físicament no es podia fer el debat o consulta respecte a algun punt del projecte amb Gmail es porta a terme la comunicació mitjançant el correu electrònic que proporciona l'empresa.

###### • Meet i Hangouts

Donada la situació de la pandèmia, ha arribat un punt en el qual no podíem fer les reunions i treballar a distància, correu és una solució fàcil i ràpida per tractar punts en concret però a mesura que el projecte avança hi ha aspectes amb els

quals mitjançant el correu era bastant difícil de comunicar i transmetre la idea que es vol, per tant per fer trucades aprofitant el servei que ens proporciona el compte de Google que té l'empresa s'ha utilitzat les dues eines de trucades que proporciona amb el compte.

- Java 1.8

Com ja s'ha explicat el projecte és una comprovació de capacitats i una extensió d'un projecte ja en marxa, desenvolupat amb Java, i per tant no ha sigut una elecció el fet de treballar amb aquest llenguatge de programació. Donat que hi ha una implementació de sistema de seguretat hi ha *features* amb les quals és necessari treballar amb la versió 1.8 de Java.

- Eclipse

En desenvolupar el projecte en Java, Eclipse és una de les solucions per excel·lència, donat que porta una gran quantitat de complements a més donant la possibilitat de poder treballar en diferents sistemes operatius. Un altre punt a favor és que ja s'ha treballat amb aquest entorn i per tant un extra de comoditat en treballar-hi amb un projecte que ja està en marxa coneixent diferents complements perquè l'entorn de treball no sigui un obstacle. Per últim, el fet de tenir la consola en el mateix IDE fa molt més àgil i còmode la transició cíclica entre fases de desenvolupament i test del funcionament.

- Gitlab

Com ja s'ha comentat el que es du a terme és una ampliació de capacitats d'una xarxa ja existent i el codi ja estava sent compartit mitjançant Gitlab per tant no hi ha possibilitat a decidir si utilitzar Gitlab o una altra solució, ja que no és pas òptim el fer una còpia del conjunt de projectes que formen Garlanet a un altre servei que no sigui Gitlab així que la solució per poder compartir el codi de manera fàcil amb la resta d'actius en el projecte és Gitlab.

- Overleaf

Decisions preses, documents recollint el funcionament de funcions o classes, o qualsevol altra decisió presa que requereix ser documentada, tal com la pròpia memòria del projecte, s'ha decidit utilitzar l'editor de text en línia Overleaf. El principal avantatge és tenir els documents al núvol i poder accedir-hi des de qualsevol lloc i com a diferència entre Overleaf i Google Drive, altra solució semblant en aquest cas, la disponibilitat de plantilles de disseny i altres detalls de disseny a l'hora de generar documents.

- DB Browser SQLITE Amb una interfície simple i senzilla d'utilitzar és el software gratuït i de codi obert que s'ha escollit per comprovar que les insercions SQL s'estiguin fent de manera correcta, únicament s'utilitza com a visualitzador donat que la creació i modificacions es fan des del codi i el que ens interessa és veure com aquests canvis s'estan fent de manera correcta.

## 4. GARLANET

A continuació s'explicarà que és Garlanet i quins components la formen i s'aprofundirà en els diferents components que són més importants de profunditzar per poder entendre més endavant el que es pretén fer al TFG. Referenciar tota la informació sobre Garlanet al document *Joan Manuel Marques, Helena Rifà-Pous, Samia Oukemeni. From False-Free to Privacy-Oriented Communitarian Microblogging Social Networks. (2020. Pendent de revisió)*[1] on s'explica molt més detinguda i de manera més profunda que és Garlanet i com funcionen les diferents parts que la componen.

Ja s'han esmentat les diferents idees sobre les quals Garlanet se sosté i es construeix, per tant ara veurem una definició de Garlanet al complet. Garlanet ofereix un servei de xarxa social de microblogging escalable, resistent a fallades i no depèn de cap proveïdor extern. Els usuaris de Garlanet poden crear perfils i utilitzar aquests serveis, entrar a la xarxa com a contribuïdors al sistema d'emmagatzematge o totes dues variants. Una vegada feta una descripció de Garlanet veiem a continuació com està format Garlanet.

### 4.1. Disseny i arquitectura de Garlanet

- Directory Service

El Directory Service s'encarrega de decidir i resoldre la ubicació de les dades. No recull cap informació, únicament els usuaris que estan registrats al sistema, que pugui comprometre la privacitat dels usuaris. Està format per tres subcomponents, dels quals únicament ens interessa el Directory Service.

Aquest conté un mapatge d'ID's de microserveis i els servidors. També és el Directory Service qui està a càrrec de garantir que per cada ID de microservei hi ha el nombre de rèpliques indicat als paràmetres. Com ja hem vist Garlanet està formada per

servidors que proporciona la pròpia comunitat i per tant són màquines heterogènies que no sempre estaran oferint servei, per tant hi ha mètodes eficients per saber on guardar les diferents rèpliques dels microserveis que tenen com a input percentatge de temps connexió i desconnexió, la seva capacitat, ocupació amplada de banda i altres.

Com es pot veure tenir un Directory Service centralitzat és compatible amb la qualitat de descentralització i la privacitat del sistema, aquest únicament resol noms i manté la llista de servidors per tant no conté informació sobre els usuaris o les seves identitats. Tota la lògica respecte d'aquests microserveis que s'han esmentat hi és pràcticament tota al component Application que se situa a l'ordinador dels usuaris, mentre que els servidors interactuen amb els usuaris sense la intervenció del Directory Service.

- Repository

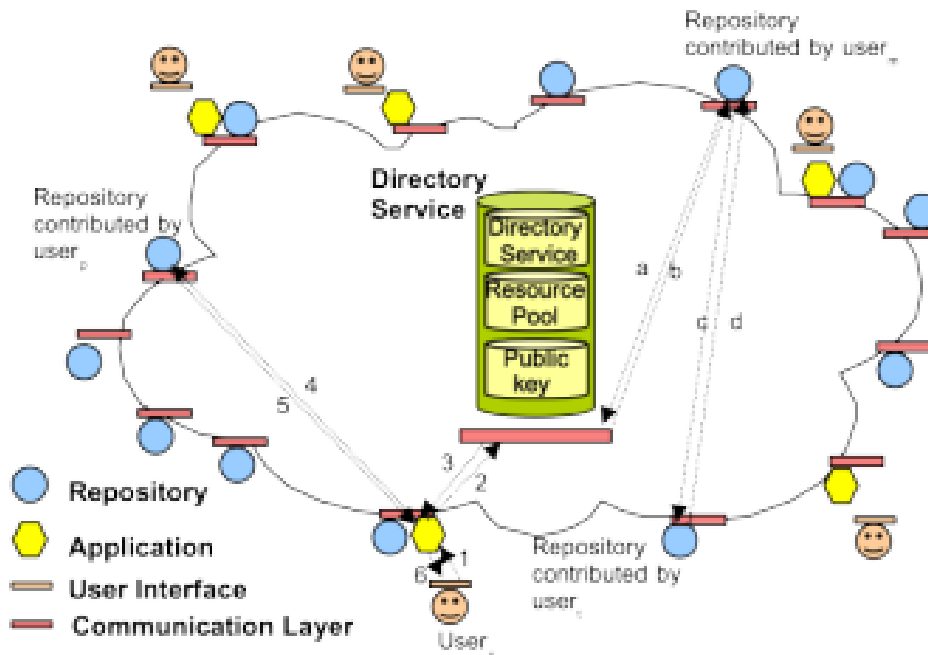
Els servidors, anomenats *repositories* en Garlanet, s'encarreguen d'emmagatzemar informació. Com són màquines de la pròpia comunitat no es pot garantir que aquestes estiguin en permanent servei per tant per garantir la disponibilitat de les dades, aquestes es repliquen en diferents *repositories* que és supervisat, com ja hem vist, pel Directory Service. Les dades de les quals parlem són, des dels microserveis fins a fitxers adjunts dels missatges i informació del perfil. Per propagar aquesta informació es fan sessions de consistència seguint un model que no cal veure per entendre el treball fet al TFG. Per poder contribuir amb la funcionalitat d'emmagatzemament a Garlanet, els usuaris hi han d'instal·lar aquests *repositories* a la màquina que volen posar en servei i connectar-la amb la comunitat.

- Application

És la part que proporciona les funcionalitats de microblogging a l'usuari. Cada usuari ha d'instal·lar el component d'Application al dispositiu conjuntament amb la interfície de l'usuari. Llavors l'usuari pot enviar i rebre missatges utilitzant la interfície d'usuari interactuant amb el component d'Application, en ser dos components separats, qualsevol desenvolupador pot implementar una interfície i connectar-la amb l'Application de Garlanet i fer-la servir.

Per agrupar els diferents components que hem vist i la informació de cadascun, es mostra una captura que agrupa els diferents components que s'han explicat per ajudar donant una

versió gràfica del que correspon a un desplegament de Garlanet.



**Figura 7:** Exemple de un desplegament de Garlanet

## 4.2. Microserveis

És un aspecte que ha sortit en veure que és Garlanet i com està formada, i identifica una idea que defineix molt Garlanet que és la divisió del treball en parts més petites, és la part que s'ha implementat per fer un pas cap a la implementació de la comunicació un-a-un i definitivament ha sortit el concepte i per tant cal almenys fer una definició que pugui fer una idea de que es tracten per ajudar a entendre el que és Garlanet.

Les funcions de Garlanet, consisteixen en un grup de microserveis, cadascun d'aquests tracta amb una de les funcionalitats per cada usuari de manera que el pes de càlcul és més baix que no pas si tot es tractés conjuntament.

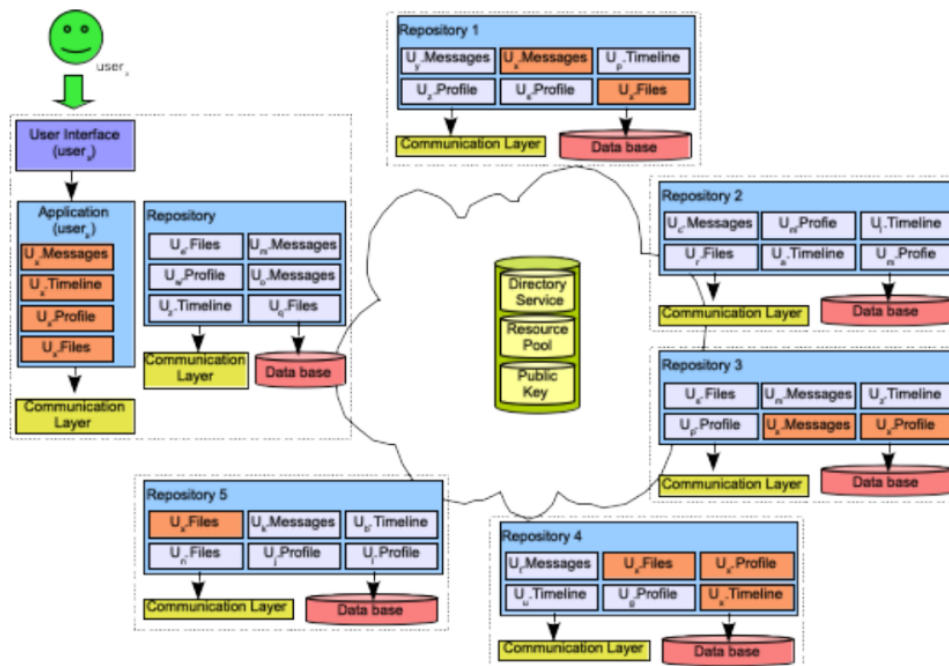
Com s'ha mencionat cada usuari conté un grup de microserveis, veiem quins són i una breu descripció del treball que fa cadascun d'aquests, donat un usuari A tenim:

- Messages  
Conté tots els missatges que ha enviat l'usuari A.
- Timeline  
Conté tots els missatges que ha enviat ell mateix, igual que Messages, però també els missatges que ha rebut de tots els usuaris que segueix A.
- Profile  
Conté la informació del perfil de l'usuari A.
- Files  
Conté tots els fitxers que s'han enviat mitjançant missatges del propi usuari A.

Els microserveis d'un usuari estan distribuïts de manera independent, el que vol dir que si un *repository* manté una rèplica del microservei Messages de l'usuari A, no ha de mantenir una rèplica dels altres tres microserveis.

Una altra vegada, per representar gràficament el que acabem d'explicar tenim la següent figura, l'usuari que es veu, té instal·lat tant la part d'usuari de Garlanet com la de contribuïdor a aquesta mateixa, per tant disposa d'Application i la seva GUI per un costat i el *repository* instal·lat. En taronja, es veu com a Application té el seu pack de microserveis Messages, Timeline, Profile i Files; a més, al *repository* que té instal·lat, conté rèpliques de microserveis d'altres usuaris de la xarxa, informació a la qual aquest usuari no té accés, a menys que el segueixi, per principi de seguretat. A més de l'usuari que tenim hi ha uns altres *repositories* que contenen rèpliques de diferents microserveis entre els quals es troben els microserveis de l'usuari, i com hem dit abans el fet de contenir una rèplica d'un microservei no implica tenir rèpliques de la resta dels microserveis d'un mateix usuari.





**Figura 8:** Exemple distribució de microserveis

Per últim, respecte a la seguretat a Garlanet fer un apunt sobre com funciona la privacitat i la seguretat, Garlanet és una xarxa social de microblogging que preserva la privacitat, aprofitant l'estructura descentralitzada per proporcionar als usuaris una major protecció de la privacitat i control més directe sobre les dades i qui té accés a elles a més està dissenyat per proporcionar comunicació segura. Com no és un fonament que necessitem conèixer profundament pel TFG amb aquesta definició es dona per definit en què consisteix la seguretat a Garlanet.

Per tant ja tenim una visió global de com funciona Garlanet en un desplegament real i com treballa i gestiona la informació, ara ja podem profunditzar en el treball que s'ha fet i veure els diferents components amb els quals s'han treballat i com s'ha simulat el comportament tant dels usuaris com dels *repositories* per avaluar el funcionament del sistema.

## 5. AVALUACIÓ DEL SISTEMA

La part d'avaluació i posada en marxa dels tests, tot el treball s'ha fet en un projecte independent de Garlanet, on en aquest projecte s'emula els diferents comportaments dels components que hi participin en l'experiment en qüestió, que funcionen conjuntament amb

la resta del sistema que es desplega de manera normal com si fos un desplegament real.

Abans de veure els diferents experiments que s'han definit es presentarà els diferents components que no hem vist donat que no formen part del desplegament real de Garlanet donat que les funcions que desenvolupen són, generació de comportaments, posada en marxa de l'experiment, generar un entorn per desenvolupar aquest, i altres funcions que veurem a continuació. Primer es nomenaran els diferents components i explicant quina és la seva funció dins de l'experiment i després es farà una descripció d'una posada en marxa, per últim veurem els experiments que s'han definit i en quins entorns es fan i els resultats que s'han obtingut.

## 5.1. Components

- TestServer

És el primer component a posar en marxa per poder començar a definir l'experiment, el Test Server, s'encarrega de rebre les peticions dels diferents components que volen comunicar-se per primera vegada amb l'experiment en el qual hi participen, atén aquestes peticions i segons l'identificador que s'hagi rebut com a paràmetre es dóna el port en que el TestServerExperimentManager corresponent estigui escoltant les peticions. Per tant el que fa és redirigir les peticions a l'experiment corresponent, així permet tenir diferents experiments funcionant alhora.

- TestServerExperimentManager

Com indica el seu nom, és la classe que gestiona l'experiment, això vol dir que, tot component que formi part de l'experiment ha de comunicar-se amb ell per adquirir un dels comportaments que s'han generat format pels paràmetres que necessiti depenent del tipus de component que sigui, usuari o *repository*. No és l'única funció que desenvolupa, també recull algunes altres funcions que veurem més endavant quan veiem el funcionament d'un experiment.

- GenerateUserBehaviour

Una classe destinada íntegrament al càlcul i definició de les relacions dels usuaris entre ells, el que es tradueix en quants usuaris seguim i quants ens segueixen, a més de la quantitat de missatges que enviarà cada usuari al dia de mitja.

- **Send Arguments**

Encarregat de llegir els diferents paràmetres de `parameters.properties`, en pocs casos es fan càlculs donat que com s'ha definit gran part de la generació dels comportaments i particularitats dels experiments es calculen al `TestServerExperimentManager` o al `GenerateUserBehaviour`, i envia aquests paràmetres al `TestServer`, com per exemple l'identificador de l'experiment que es vol fer.
- **Parameters.properties**

Un fitxer de text el qual recull els diferents arguments que defineixen el comportament dels diferents components de l'experiment i altres dades com la duració de l'experiment. Alguns d'aquests paràmetres són utilitzats per definir comportaments i d'altres són variables estàtiques que per tots els experiments són les mateixes i són els "usuaris" d'aquestes variables qui accedeixen donat que no es necessiten per cap càlcul ni definició d'un cert comportament que calgui ser calculat en un punt intermedi.
- **RepositoryManager**

Encarregat de donar el comportament que s'espera dels *repositories* que es posaran a disposició de la xarxa en un desplegament real, aquest component és l'encarregat de posar en marxa i aturar un *repository* seguint el comportament que hagi rebut.
- **GarlanetUserEmulator**

Encarregat de simular els usuaris, aquest component genera l'activitat que pot generar un usuari amb les característiques que té Garlanet actualment, que són registrar-se al sistema, seguir a altres usuaris, rebre els missatges d'aquests, veure els missatges adjunts en aquests si és que hi ha i enviar missatges, aquest últim comportament és monitorat completament per un thread que genera la classe, `MsgSend`.
- **MsgSend**

Donat que es defineix un nombre de missatges a enviar al dia, s'ha definit en un thread a part la generació d'activitat que comporta l'enviament de missatges, també s'inclou el contingut dels missatges, que són un seguit de paràmetres per controlar els missatges que s'envien i en cas de tenir un missatge adjunt es genera una imatge aleatòria per adjuntar al missatge.

- NUserEmulator

Donat que l'objectiu és arribar a fer tests amb centenars d'usuaris, es va implementar una classe que executés usuaris anomenada NUserEmulator en forma de threads, aquesta classe el que fa és executar N usuaris, segons el paràmetre que li passem, aquests s'executen de manera seqüencial per no saturar el sistema volent registrar tots els usuaris de cop, aquí és on es calcula el temps que l'usuari després de fer register i login ha d'esperar abans de començar a generar activitat fins que tots els usuaris s'hagin registrat.

## 5.2. Funcionament

Una vegada hem vist els diferents components que s'afegeixen al desplegament de Garlanet per poder fer tests ara veurem l'orde de posada en marxa i com interactuen entre ells.

Independentment de si estem fent tests o no el Directory Service és una peça clau que sempre ha d'estar en funcionament, com ja hem vist és el component encarregat de direccionar les comunicacions entre usuaris i *repositories*.

En segon lloc el que encenem és un TestServer, és l'administrador dels diferents tests que es volen fer, per tant ha d'estar encès abans de posar cap altre component en marxa.

Ara sí que podem començar a generar els comportaments dels diferents components que hi participaran a l'experiment, Send Arguments s'encarrega d'enviar els diferents arguments al TestServer, aquest al rebre aquesta petició el que fa és crear una instància de Test Server Experiment Manager i és aquest qui tracta amb els arguments.

Test Server Experiment Manager és qui interpreta els arguments que ha rebut, Test Server no, i conjuntament amb GenerateUserBehaviour que és l'encarregat de fer els càlculs necessaris per generar els comportaments que els retorna al Test Server Experiment Manager i és aquest qui els guarda fins que algun usuari es comunica amb ell.

En aquest punt el sistema està preparat, ara únicament el que cal és engegar els diferents *repositories* i usuaris que hi participen.

Primer posem en marxa els *repositories* que vulguem, el comportament dels *repositories* és únicament engegar-se i apagar-se per simular el comportament per tant a diferència dels usuaris el seu comportament no conté càlculs, únicament se selecciona un tipus de comportament dels que hi ha definits, més endavant veurem quins són.

Donat un temps perquè els *repositories* tinguin un estat estable, el que implica que el Directory Service els coneix comencem a posar en marxa usuaris.

Aquests usuaris fan el mateix que els *repositories*, aquests envien un missatge al TestServer indicant a quin test hi participen i aquest els redirigeix al TSEM corresponent per obtenir un dels comportaments que s'han definit.

Una vegada els usuaris es registren, comencen a generar activitat igual que els *repositories*. Ara veurem l'ordre dins d'aquesta activitat de l'usuari

- Registre i login

Com és lògic el primer que fa l'usuari és registrar-se i fer login, com faríem en qualsevol xarxa social, aquests usuaris no es desconnecten per tant es fa un únic registre i login. Just després el que fa l'usuari és donar la seva carta d'amic.<sup>3</sup>aquest és un fitxer que necessita un usuari per poder seguir i veure l'activitat d'aquest usuari.

- Relacionar-se i obtenir informació

Un dels dos processos d'activitat que fa l'usuari contínuament fins que s'acaba l'experiment, és mirar a quants usuaris ha de seguir, segons el comportament que se li ha assignat, i mirar si algun d'aquests ha enviat algun missatge.

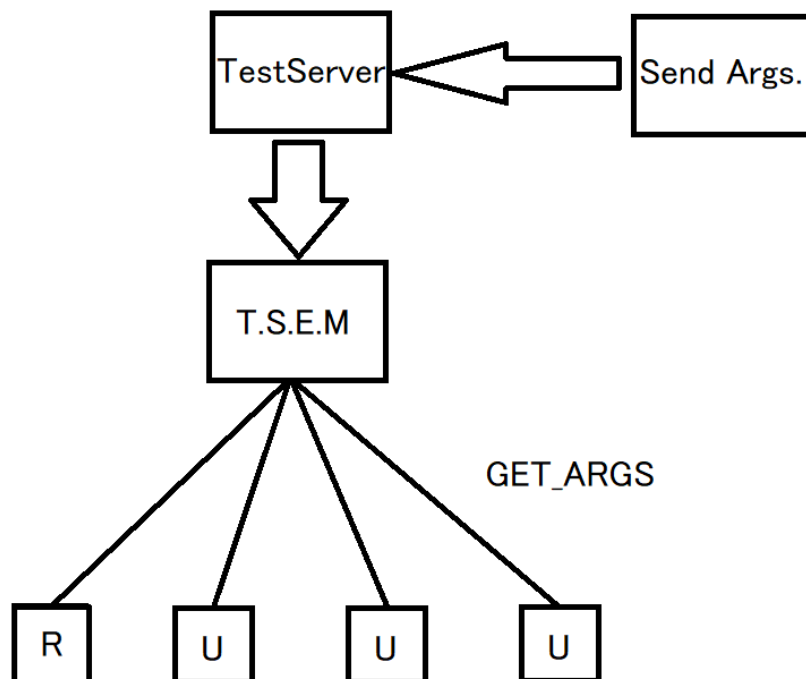
Aquesta última comprovació el que fem és una consulta al nostre microservei Timeline, per tant aquí hi veurem els missatges que hem enviat nosaltres nous o els que han enviat les persones a les quals seguim i ho afegim a la resta de missatges que ja hem vist, seguidament el que fem és interactuar amb els missatges que tenen arxius adjunts, en cas que hàgim de fer-ho.

- Enviar missatges

L'altre procés que genera activitat és l'enviament de missatges, en un thread apart

cada dia que passa es generen, segons el nombre de missatges que hàgim d'enviar, un nombre aproximat a aquests missatges mitjançant una funció amb mitjana aquest paràmetre. Aquests missatges es distribueixen en un temps de 16 hores, donat que s'han donat 8 hores que és el temps que l'usuari descansa. Durant la resta els missatges es distribueixen de manera aleatòria, podent haver d'enviar en un mateix moment més d'un missatge si es donés el cas. En cas que el missatge hagi de tenir un fitxer adjunt, es genera una imatge de manera aleatòria i s'adjunta al missatge.

Una vegada el test acaba, els usuaris acaben la seva execució i manualment s'han d'aturar els altres components per fer una altre execució, si volem fer un altre test es podria fer, donat que es crearia una altra instància de TSEM quan arribés un missatge de Send Arguments a Test Server amb els mateixos paràmetres o diferents, els *repositories* els tindríem que apagar i tornar a executar, donat que en el període de registre dels usuaris aquests han de mantenir-se encesos per assegurar que el registre es fa correctament, no cal que estiguin tots els *repositories* encesos des de principi però els que estiguin si han de mantenir-se en funcionament.



**Figura 9:** Estructura i primeres comunicacions dels components que formen part dels tests. L'exemple consta d'un *repository* i tres usuaris.

### 5.3. Descripció tests

Hem vist Garlanet, la seva estructura i el seu funcionament sobretot les parts que ens interessen per entendre la seva avaluació. Hem vist els diferents components que formen Garlanet, la seva posada en marxa i el seu funcionament, per tant ara el que veurem és el que dóna forma als diferents tests, els paràmetres que defineixen als usuaris i als *repositories*.

Gran part d'aquests comportaments han sigut definits per TECHNICAL REPORT Àngel Ollé Blázquez. Director: Joan Manuel Marquès Puig que a la vegada són dades que s'han extret de l'estudi de Barracudalabs, "*Barracuda Labs 2010 Annual Security Report*" de Barracuda Labs del 2010", que defineixen aquests comportaments, característiques i parametritzacions per definir l'activitat dels usuaris, per part dels *repositories* no hi ha dades històriques d'aquests *repositories*, per tant s'han agafat unes diferents distribucions que ja hi havien declarades amb anterioritat, documentació de Garlanet.

#### 5.3.1. Usuaris

- Name

Nom de l'usuari, diferent de l'identificador que té en el sistema, aquest nom només ens serveix per identificar els diferents usuaris donat que els generem a partir d'un nombre que coneixem.

- Followers

El primer que es defineix pel que fa a relacions amb altres usuaris és la quantitat de seguidors, que segueixen els següents percentatges:

Nombre de followers	Nivell de probabilitat(%)
0	11
1-9	39
10-99	33
>100	17

**Figura 10:** Percentatge de cada categoria de followers, taula extreta de TECHNICAL REPORT Àngel Ollé Blázquez amb director Joan Manuel Marquès Puig.

- Missatges

Una vegada tenim el nombre de seguidors que tenim adjudiquem un nombre de

missatges a enviar cada dia, es passa aquest nombre de missatges que ha d'enviar l'usuari cada dia, aquest paràmetre s'utilitza com a mitja per definir una corba on aquest valor és la mitja de missatges que envia l'usuari amb el pas del temps. La següent taula mostra la distribució amb el que es generen aquests missatges. La taula es llegeix de la següent manera, el 98,18% dels usuaris que tenen 0 followers envien menys d'1 missatge al dia, un altre cas seria que el 13,64% dels usuaris que tenen entre 10 i 99 seguidors envien entre 1 i 4 missatges al dia.

		Categoria followers			
		0	1 – 9	10 – 99	100+
Categoria nivells d'activitat (nombre de missatges/dia) estudi Barracuda	<1	98,18%	94,62%	75,45%	40%
	1 – 4	1,82%	3,08%	13,64%	29,41%
	5 – 9	0%	1,03%	6,06%	17,65%
	10 – 99	0%	1,03%	4,55%	11,76%
	>100	0%	0,26%	0,3%	1,18%

**Figura 11:** Relació entre nombre de seguidors i quantitat d'activitat, taula extreta de TECHNICAL REPORT Àngel Ollé Blázquez amb director Joan Manuel Marquès Puig.

- Following

Igual que amb els seguidors, es genera la quantitat de persones a seguir.

Nombre de following	Nivell de probabilitat(%)
0	16
1-9	27
10-99	40
>100	17

**Figura 12:** Distribució de persones a seguir.

Aquestes definicions de comportaments i posterior relació entre persones que demanden seguidors i persones que demanden seguir a persones no són exactes, donat que són dos variables que no tenen relació entre elles i són percentatges que varien per tant no es respectaran els valors que es generen al principi, amb vista d'això



el que es va fer és que en cas que hàgim de seguir a més persones de les que hi ha disponibles, es reduirà aquest valor fins al màxim de persones a les quals es pot seguir, fent generacions d'aquests escenaris es va veure que l'únic grup afectat era el grup que vol seguir a >100 usuaris, i funcionava correctament, amb marge d'error de dècimes o poques unitats, a partir dels 300-400 usuaris.

- Interacció

Aquest valor és la probabilitat d'interaccionar amb els missatges, que tenen fitxers adjunts, que ens arriben del timeline, hi ha dos valors un pels missatges que es veuen per primera vegada i un altre valor pels missatges que ja hem vist amb anterioritat.

Interacció amb missatges nous = 30 %

Interacció amb missatges antics = 5 %

- Imatges per missatge

Un valor que defineix els percentatges que tenen els missatges de contenir des de 0 fins a 4 imatges.

Quantitat de blocs	Probabilitat(%)	Tamany en (kB)	Dimensions
1	27	250	250x250
2	40	750	435x435
3	27	1250	560x560
4	3	1750	660x660
(5-7)	2	3000	870x870
(8-10)	1	4500	1060x1060

**Figura 13:** Taula de diferents tamanyes d'imatge.

- Tamany de les imatges

Garlanet divideix les imatges en blocs, per tant el que vàrem definir són grandàries d'imatges per poder tenir imatges dels diferents blocs.

Nombre imatges	Nivell de probabilitat(%)		
	Poc	Normal	Alt
0	67	50	46
1	27	41	40
2	3	6	18
3	2	2	4
4	1	1	2

**Figura 14:** Taula percentatges de nombre d'imatges per missatge.

- Durada del dia

Per fer els tests més ràpids s'ha definit un temps de simulació, aquest valor en cas de fer grans desplegaments s'ha definit a 4 hores cada dia, en cas de fer dies curts per comprovar que tot funciona correctament és del valor de pocs minuts.

- Temps de descans

No és real que l'usuari estigui les 24 hores del dia generant activitat, per tant vàrem decidir que les últimes 8 hores l'usuari no genera activitat i l'únic que fa és rebre missatges d'altres usuaris.

Temps de descans de l'usuari = 8 hores reals.

### 5.3.2. Repositories

Per part dels repositories el nombre de paràmetres és únicament de 3 valors:

- Qualitat del repository

Defineix la qualitat del repository.

- Probabilitat de connexió

Probabilitat de connectar-se i oferir servei o de mantenir-se oferint servei, en el nostre cas és la mateixa.

- Probabilitat de desconnexió

Ídem al cas anterior, és la probabilitat de desconnectar el repository o mantenir-se desconnectat.

Aquests tres paràmetres es recullen en la taula següent.

Qualitat	Comportament	
	Repository ON	Repository OFF
Alta	90	10
Normal	60	40
Baixa	10	91

**Figura 15:** Taula qualitats de *repositories*.

- Període d'avaluació

Hem vist les probabilitats de connexió i desconnexió, aquestes probabilitats s'avaluen cada cert temps, aquest temps ve donat pel període d'avaluació juntament amb el rang d'interval.

- Rang de l'interval

Com el seu nom indica, aquest paràmetre defineix el rang de valors que pot tenir el període d'avaluació.

Aquest rang de valors es va definir per tots els *repositories* com a 1 hora de període d'avaluació i un rang de valors de 30 minuts.

- Percentatges de *repositories*

Per últim cal definir quins són els diferents percentatges de cada qualitat de *repository*, s'han definit diferents escenaris.

	High(%)	Medium(%)	Low(%)
Scenario 1	5	20	75
Scenario 2	5	40	55
Scenario 3	5	55	40
Scenario 4	10	45	45
Scenario 5	10	50	40
Scenario 6	20	20	60
Scenario 7	20	40	40
Scenario 8	70	15	15

**Figura 16:** Taula distribucions per tipus de *repositories*.

## 5.4. Execució de tests

S'ha vist que teníem diferents escenaris on fer els tests, que es divideixen en local, servidors de la UOC i un desplegament a Guifinet.

Comencem amb els tests en local, en principi aquests tests estaven pensats per fer tests de pocs usuaris de 2-10 aproximadament donat que serien per comprovar que es fa tot com s'ha de fer i trobar el màxim de bugs possibles, per tant les proves que s'han anat fent no tenen sentit que figurin donat que no es va guardar els resultats donat que únicament es feien tests d'uns minuts per veure que tot funcionava correctament.

Una vegada tot sembla anar bé, passem als tests en els servidors de la UOC, on podem fer desplegaments amb més usuaris i més repositories podent provar els comportaments que s'han definit donat que amb els tests en local no es podien portar a terme donat que no hi participaven els suficients usuaris.

Un cop haguéssim fet el pas de fer proves en local a fer proves en remot i tot sigues funcionant correctament donaríem l'últim pas i fer les proves a gran escala.

Aquest era el pla que es tenia en un principi, però malauradament no s'ha pogut portar a terme. Un canvi que es pot veure a la planificació final respecte a la inicial va ser que els tests s'han hagut de fer al final de tot en una única setmana, el que implica dues coses.

El primer problema que ens ha portat aquest canvi de planificació és el no poder fer tests en tots els escenaris que es volien fer, donat que per fer els tests a Guifinet s'havia de fer una preparació la qual podia portar més temps del que tenia, a més hauria de fer alguns canvis als scripts, per tant, eren riscos que amb el poc temps que tenia no els volia córrer.

El segon problema és la falta de dades, s'han fet tests en local i en remot amb els servidors de la UOC però personalment no crec que aquests tests puguin decidir si Garlanet està llesta per poder llaçar-se oferint la funció de microblogging.

Un altre problema són els repositories, hi ha algun problema amb el RepositoryManager que no engega i apaga correctament el repository que ha de donar comportament amb els paràmetres, donat que va ser l'últim que es va fer, es va haver de canviar a la part de la implementació del microservei Control per motius de l'empresa i aquesta part no es va validar, donat que era una part que estava mig feta on únicament calia definir els comportaments no es va posar en marxa i no em vaig donar compte que tot i que el comportament era correcte els repositories no es posaven en marxa.

Respecte al nombre d'usuaris quedava reduït respecte al que es volia desplegar, donat que comptàvem en fer els tests en una xarxa ampla i finalment he hagut de fer els tests amb 3 servidors i les màquines locals de casa. Encara i així s'esperava poder fer tests amb pocs centenars d'usuaris per poder utilitzar el comportament que es va definir, però em vaig trobar que aproximadament a partir d'un cert nombre d'usuaris que rondava els 70, aquests deixaven de registrar-se correctament, una altra vegada, sense temps de reacció es va decidir reduir els tests a 50 usuaris per assegurar que aquests funcionessin, donat que és l'objectiu que es vol aconseguir, tenir Garlanet desplegat i veure que els usuaris despleguin la seva activitat, tot i que era una peça important la qualitat dels repositoris com afectava a les diferents peticions dels usuaris.

Es creu que del nombre d'usuaris que participen a Garlanet, entre un 5% o de manera optimista fins a un 10% aportaran un repository a la xarxa, per tant en el nostre cas per executar els tests posarem 5 repositories en marxa, encara que es podrien haver fet amb el 5% donat que aquests repositories no fallen, a menys de manera que nosaltres puguem controlar.

Recollint els diferents que m'he trobat els tests que s'han pogut arribar a fer són:

- Tests en local amb 50 usuaris.
- Tests en remot amb 50 usuaris.

Application té *features* de Java 8, per tant aquest projecte utilitza aquesta versió, el problema que això ens porta és que els servidors de la UOC estan en Java 7 i per tant no es podien posar en marxa usuaris als servidors, aquest inconvenient es va tornar a favor i així tenim la "màxima" distància entre els repositories i els usuaris, distribuint els repositories en els tres servidors de la UOC i els usuaris executant-se des de les màquines de casa.

Un altre punt a tenir en compte és la generació d'activitat dels usuaris i les relacions entre aquests, com s'ha dit l'engranatge que es va muntar per relacionar els diferents usuaris extret de les fonts mencionades estava pensat per fer tests amb centenars d'usuaris, finalment no els hem pogut fer així per tant s'ha hagut de fer un seguit de modificacions per fer tests amb sentit però allunyant-nos el mínim possible del comportament que vàrem partir d'un principi.

Per tant les taules que hem vist abans queden modificades de la següent manera:

Nombre de following	Nivell de probabilitat(%)
0	10
1-5	27
5-15	40
15-50	17

**Figura 17:** Taula usuaris que seguim. Dades extretes de TECHNI-CAL REPORT Àngel Ollé Blázquez amb director Joan Manuel Marquès Puig.

Nombre de followers	Nivell de probabilitat(%)
0	11
1-5	39
5-15	33
15-50	17

**Figura 18:** Taula seguidors. Dades extretes de TECHNI-CAL REPORT Àngel Ollé Blázquez amb director Joan Manuel Marquès Puig.

		Categoria followers			
		0	1-5	5-15	15-50
Categoria nivells d'activitat (nombre de missatges/dia) estudi Barracuda	<1	98,18%	94,62%	75,45%	40%
	1-5	1,82%	3,08%	13,64%	29,41%
	5-10	0%	1,03%	6,06%	17,65%
	10-15	0%	1,03%	4,55%	11,76%
	15-20	0%	0,26%	0,3%	1,18%

**Figura 19:** Taula adaptació d'activitat als tests que es faran.

Com es pot veure, s'ha pres mesures de contingència per fer els tests d'alguna manera significatius, donat que no es poden fer tests amb la quantitat d'usuaris que es va plantejar en un principi, s'hi ha optat per carregar els usuaris amb activitat, donat que tenim pocs usuaris el que es vol és almenys que aquests usuaris carreguin bastant el sistema. No és l'única mesura que s'ha pres, a més d'incrementar la quantitat de missatges a enviar i ajustar el nombre de seguidors i seguits, s'ha decidit fer tests on els dies durin 30 minuts, per això el màxim de missatges a enviar no arriba a 30, per no tenir el cas d'enviar més missatges

que minuts, comportament que no es creu real tampoc. Els usuaris que envien entre 15 i 20 missatges al dia és tan baix que com a molt apareix un, tenint en compte que no tenim ni 100 usuaris, les probabilitats que apareixen en la taula són encara menors, encara que un comportament de 15-20 missatges no és gaire usual tampoc, per això es recull en tan poc percentatge.

## 5.5. Resultats tests

Estudiarem dels tests els següents punts:

- Distribució dels seguidors

S'ha comentat, la generació de seguidors i la generació de persones que ens segueixen són dues variables totalment independents, per tant no tenen cap relació, a més, els percentatges de les diferents categories són diferents el que implica encara més possible d'esacord entre aquestes dues variables tenint més usuaris per seguir que usuaris que demanen seguidors. El que ens porta a variacions entre els percentatges que generem al principi i les relacions que acaben quedant, veurem com queden aquestes distribucions de seguidors donat que és el factor que varia i està fet de tal manera que el nombre de persones a les quals seguim es modifiqui per quadrar amb el valor del seguit per això no es comprova aquests percentatges i sí els de seguidors, per últim l'usuari no sap quantes persones el segueixen, no ho necessita, la dada realcionada amb els seguidors és la quantitat de missatges a enviar que és causada per la quantitat de seguidors que tenim.

- Persones seguides

Encara que no estigui evaluada la quantitat d'activitat que es rep dels altres usuaris, almenys s'ha volgut estudiar a quantes de les persones de les quals hem de seguir realment seguim en la pràctica, no seguir a alguns dels usuaris pot donar-se per problemes de connexió a l'intentar rebre el seu identificador, que l'usuari no s'hagi registrat o algun altre problema, per tant també es farà recompte de les persones que seguim, i que per tant veiem la seva activitat, respecte al que havíem adjudicat.

- Registre

A la primera fase de l'experiment es fa el registre de tots els usuaris, és possible que en aquest temps alguns dels usuaris fallin a registrar-se, sobrecàrrega, pèrdua d'algun missatge o altres casos... Per tant es comprova quants d'aquests 50 usuaris que hem

engegat s'han registrat correctament i esperen a poder fer l'activitat que els hi pertoca.

- Missatges

Per últim avaluarem els missatges que enviem, això inclou el nombre d'intents que hem hagut que fer per enviar el missatge en cas que s'envii, i en cas que no s'hagi enviat quin percentatge són respecte al total de missatges que teníem per enviar.

- Resum tests

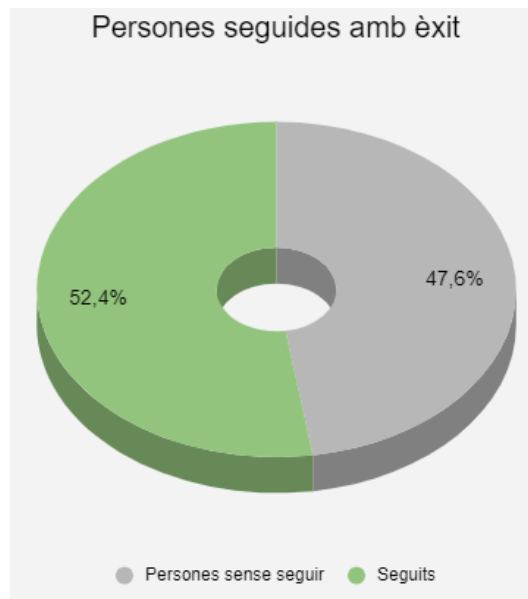
Per tancar tota la informació respecte als tests es fa una mitjana per usuari de més informació respecte els missatges, s'ha fet una mitja de les diferents variables que s'han avaluat i comptant els usuaris que s'han executat correctament s'ha generat un comportament mitjà per cada test, de les variables les quals no es fan gràfics.

S'han fet 10 tests locals i 5 remots, d'aquests veurem l'abstracció de dades de 3 locals i 3 remots, estudiant els paràmetres que s'acaven de definir fent una representació gràfica dels diferents comportaments dins de cada paràmetre.

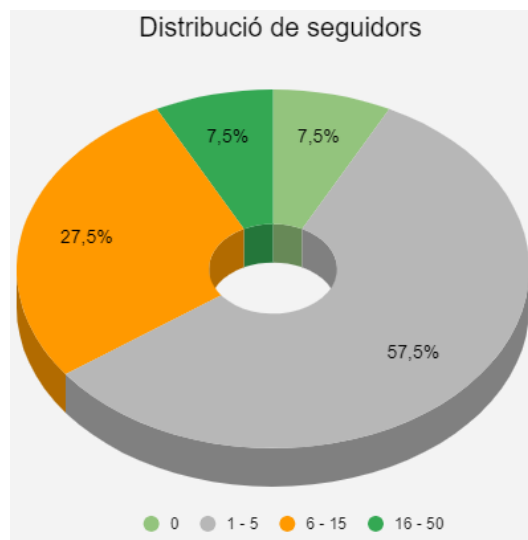


### 5.5.1. Local

- Test 1



**Figura 20:** Gràfic percentatge de persone seguides amb èxit.



**Figura 21:** Gràfic distribució d'usuaris respecte als usuaris que segueixen.

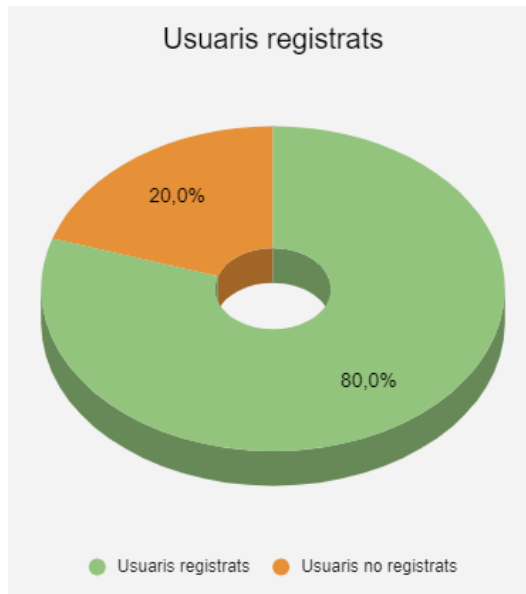
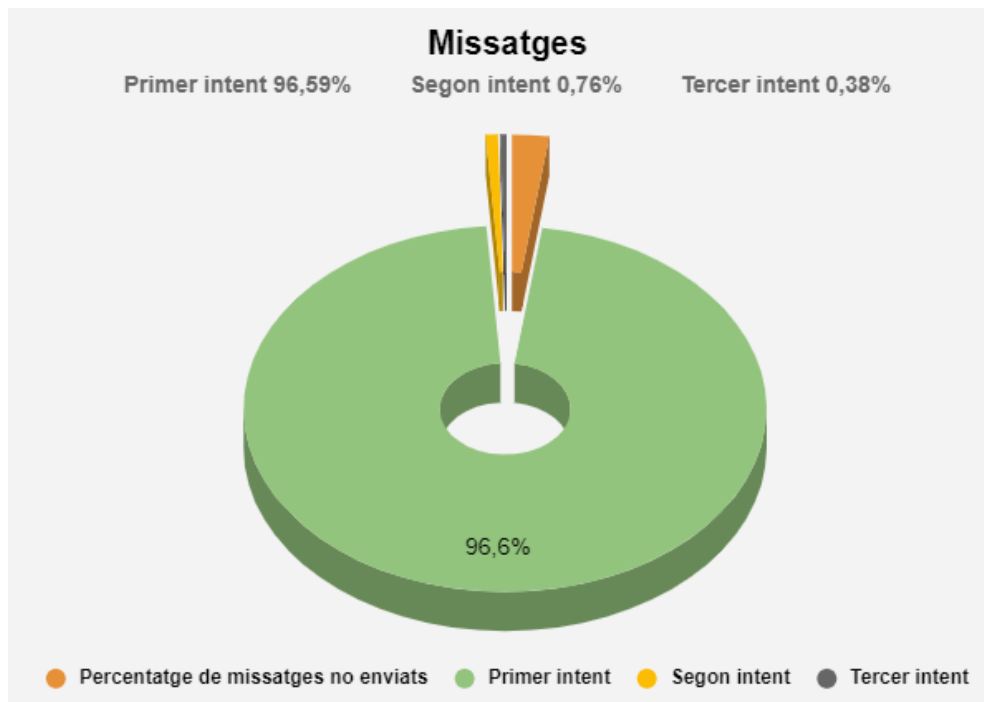


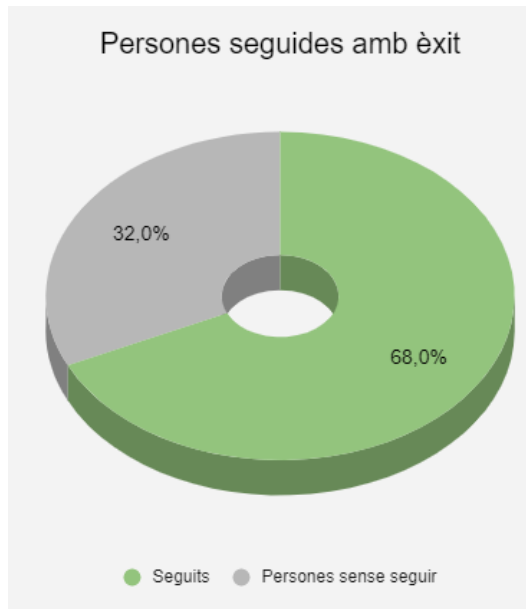
Figura 22: Gràfic percentatge d'usuaris registrats correctament.



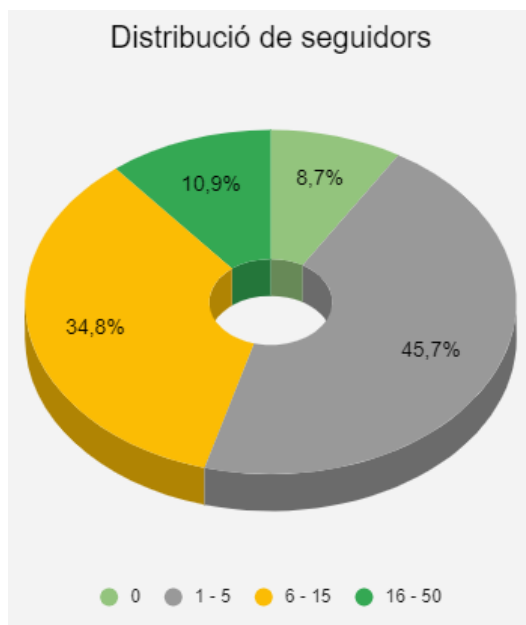
Missatges amb fixers	1 fixer	2 fixers	3 fixers	4 fixers	Missatges sense fixers	Temps en enviar (ms)
	5,05	3,08	1,68	0,20	0,10	1,55
						18.380,88

Figura 23: Gràfic amb missatges enviats i taula amb continguts.

■ Test 2



**Figura 24:** Gràfic percentatge de persone seguides amb èxit.



**Figura 25:** Gràfic distribució d'usuaris respecte als usuaris que segueixen.

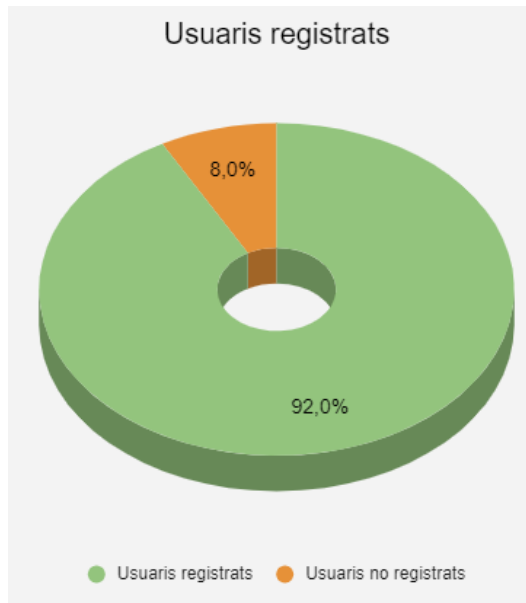
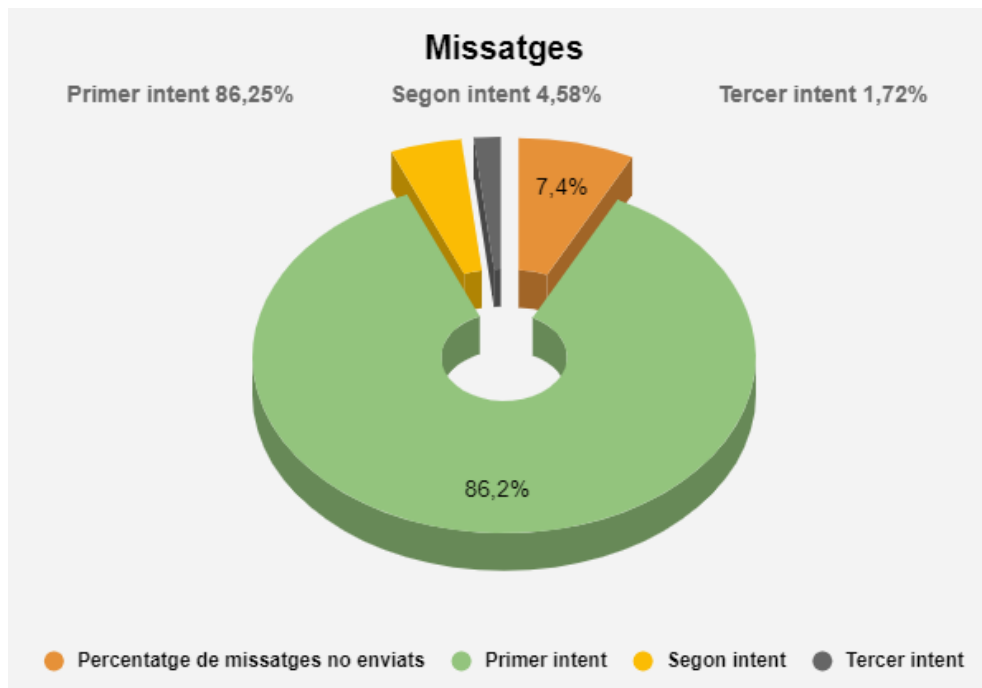


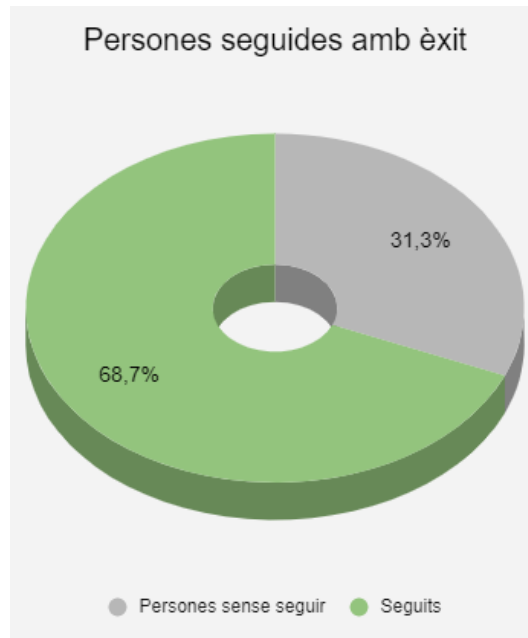
Figura 26: Gràfic percentatge d'usuaris registrats correctament.



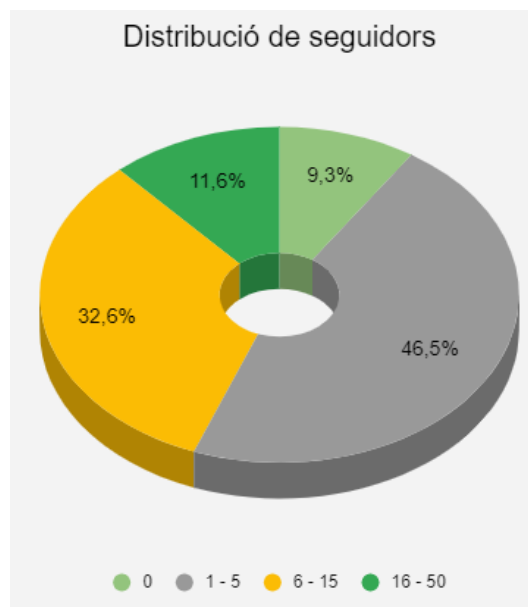
Missatges amb fixers	1 fixer	2 fixers	3 fixers	4 fixers	Missatges sense fixers	Temps en enviar (ms)
	5,96	3,98	1,53	0,27	0,18	1,80
						14.971,91

Figura 27: Gràfic amb missatges enviats i taula amb continguts.

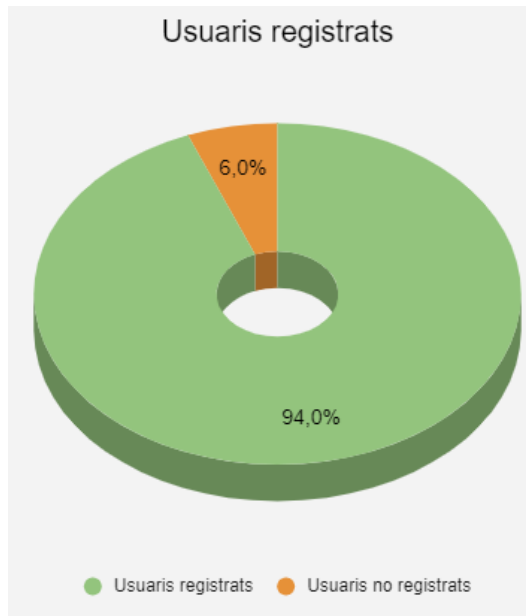
■ Test 3



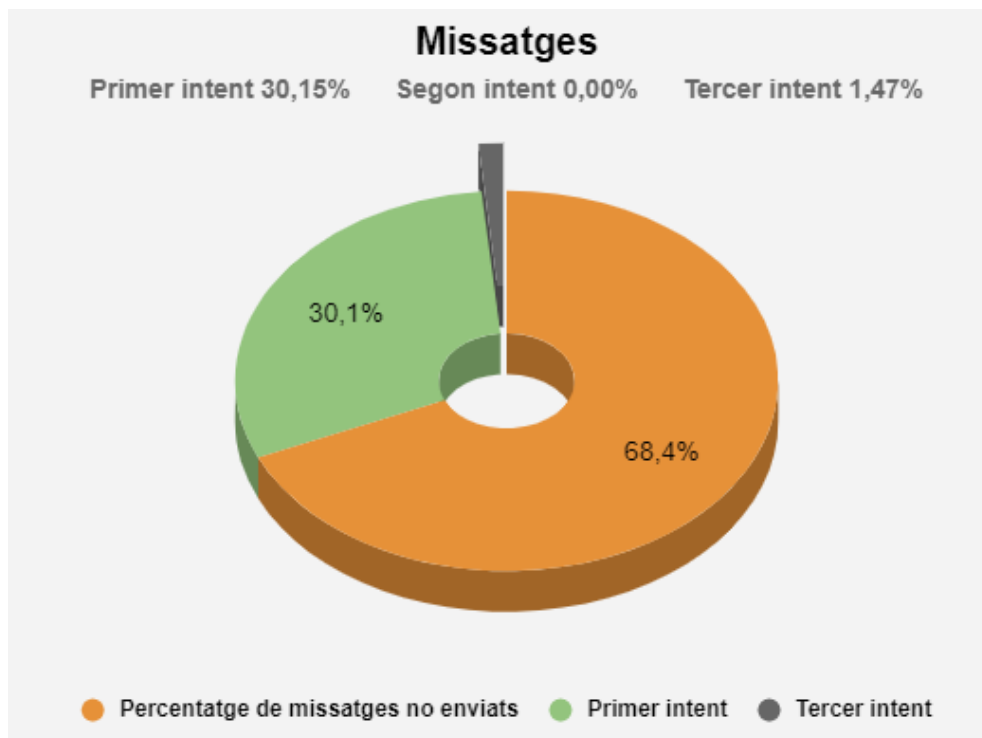
**Figura 28:** Gràfic percentatge de persone seguides amb èxit.



**Figura 29:** Gràfic distribució d'usuaris respecte als usuaris que segueixen.



**Figura 30:** Gràfic percentatge d'usuaris registrats correctament.

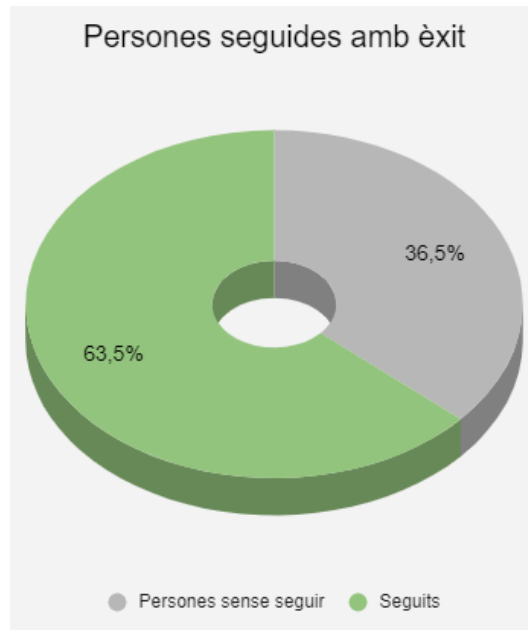


Missatges amb fitxers	1 fitxer	2 fitxers	3 fitxers	4 fitxers	Missatges sense fitxers	Temps en enviar (ms)
	2,55	1,53	0,77	0,13	0,13	0,34
						4.626,98

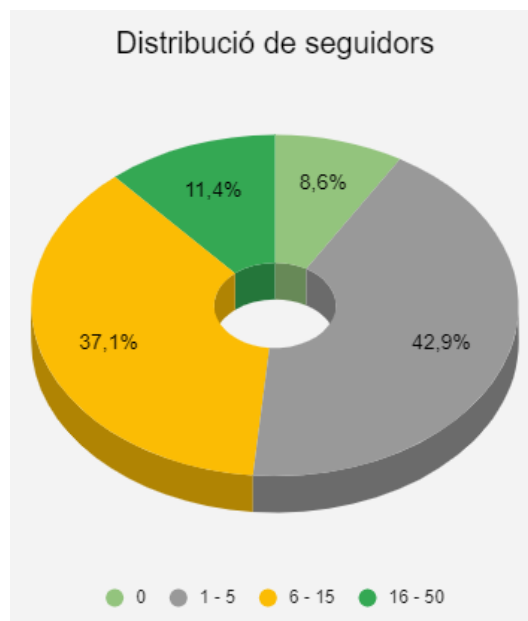
**Figura 31:** Gràfic amb missatges enviats i taula amb continguts.

### 5.5.2. Remot

- Test 1



**Figura 32:** Gràfic percentatge de persone seguides amb èxit.



**Figura 33:** Gràfic distribució d'usuaris respecte als usuaris que segueixen.

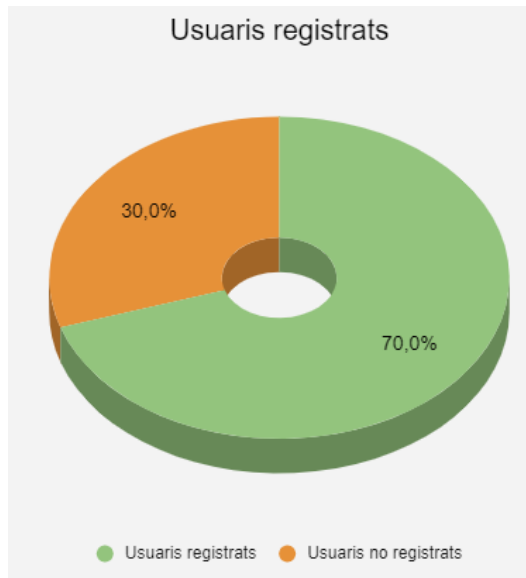
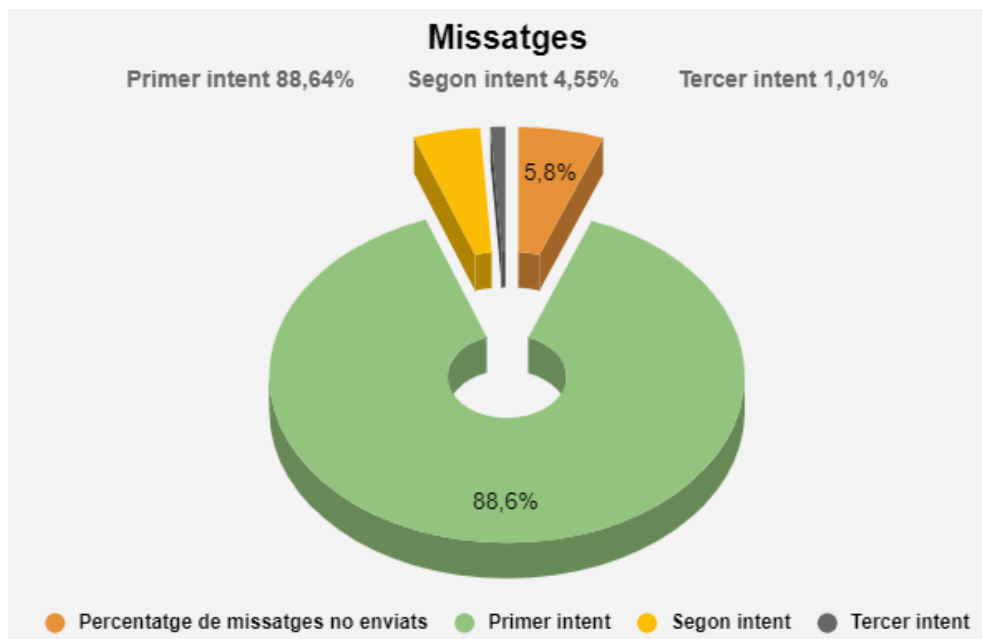


Figura 34: Gràfic percentatge d'usuaris registrats correctament.

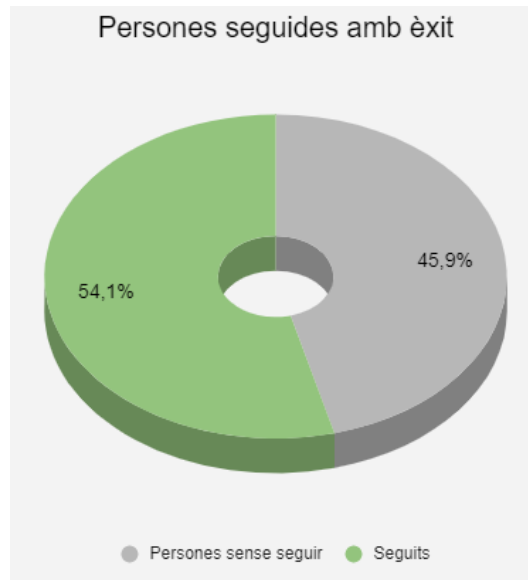


Missatges amb fixers	1 fixer	2 fixers	3 fixers	4 fixers	Missatges sense fixers	Temps en enviar (ms)
	8,03	5,18	2,68	0,22	0,18	3,29
						22.737,49

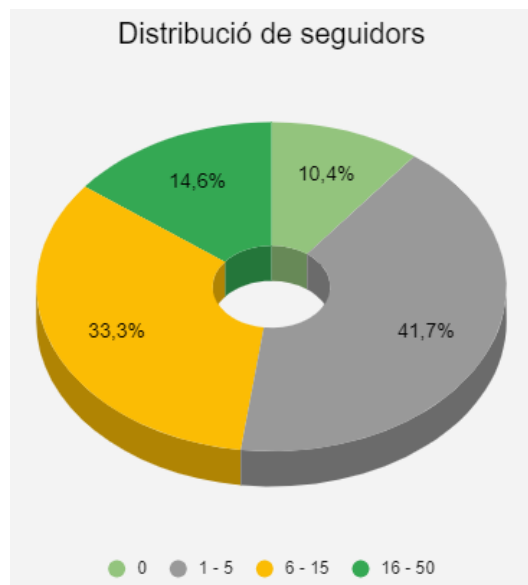
Figura 35: Gràfic amb missatges enviats i taula amb continguts.



■ Test 2



**Figura 36:** Gràfic percentatge de persone seguides amb èxit.



**Figura 37:** Gràfic distribució d'usuaris respecte als usuaris que segueixen.

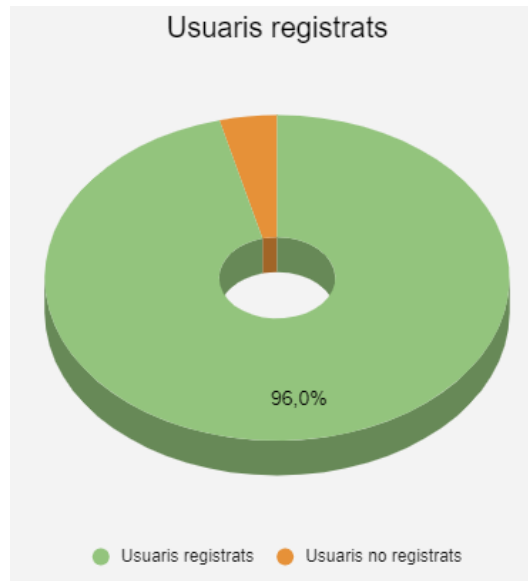
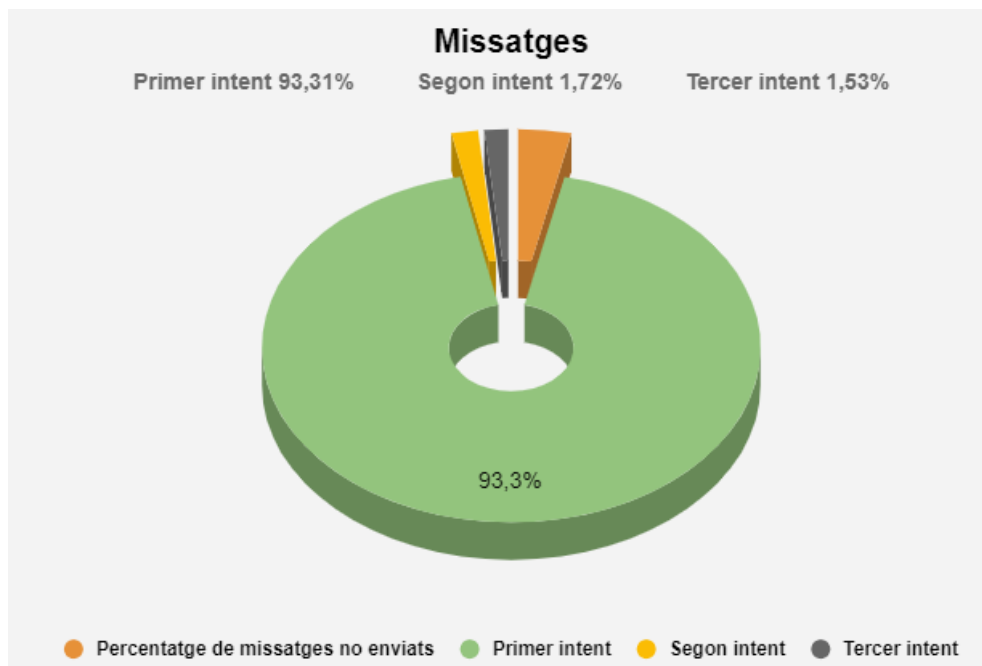


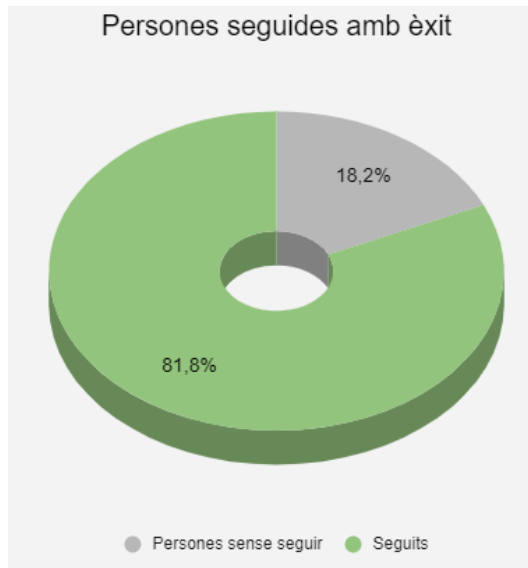
Figura 38: Gràfic percentatge d'usuaris registrats correctament.



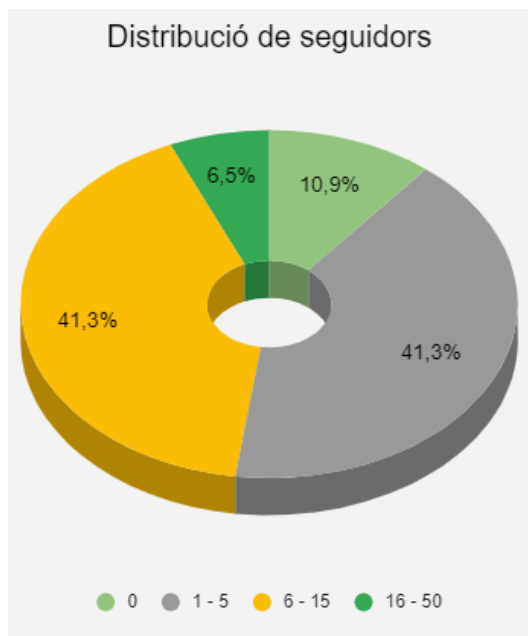
Missatges amb fixers	1 fixer	2 fixers	3 fixers	4 fixers	Missatges sense fixers	Temps en enviar (ms)
	7,96	4,71	2,92	0,17	0,17	2,94
						12.050,61

Figura 39: Gràfic amb missatges enviats i taula amb continguts.

■ Test 3



**Figura 40:** Gràfic percentatge de persone seguides amb èxit.



**Figura 41:** Gràfic distribució d'usuaris respecte als usuaris que segueixen.

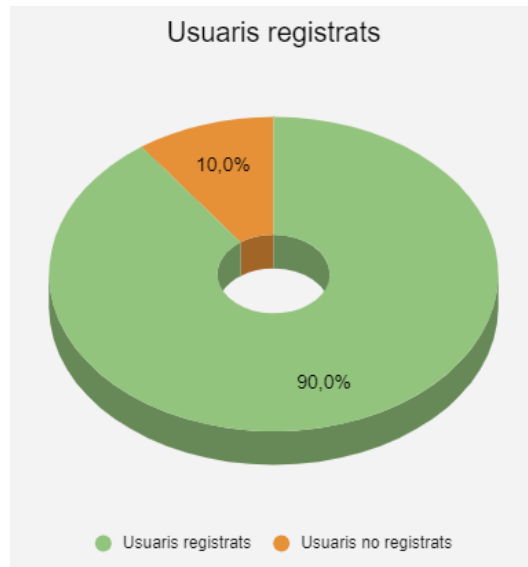
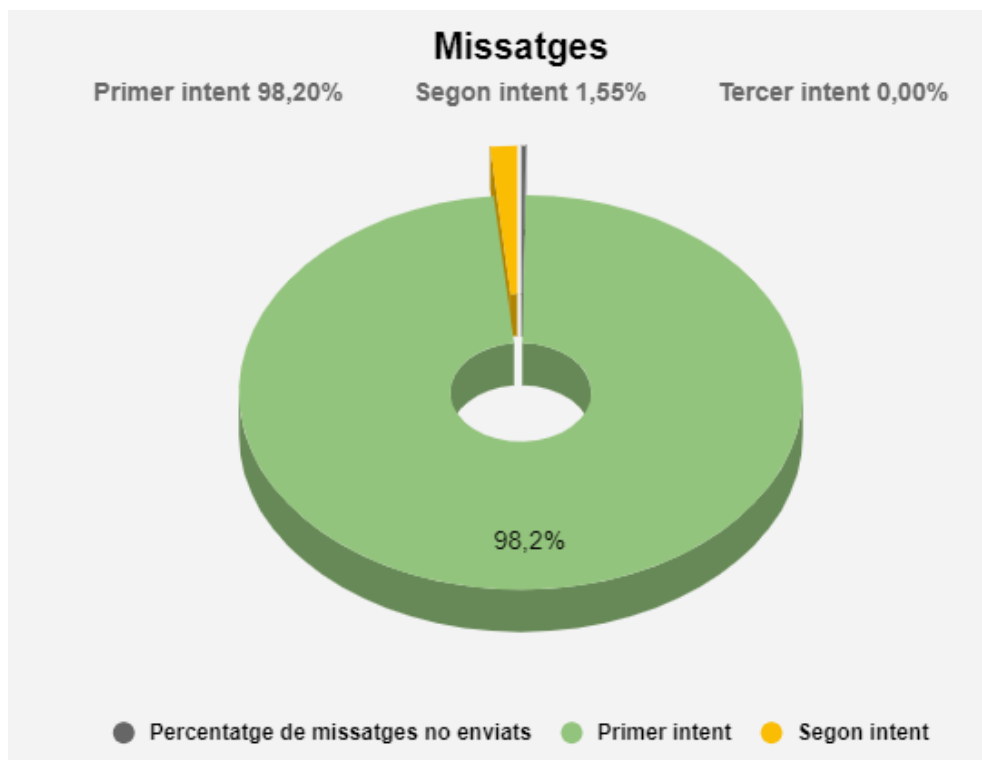


Figura 42: Gràfic percentatge d'usuaris registrats correctament.



Missatges amb fixers	1 fixer	2 fixers	3 fixers	4 fixers	Missatges sense fixers	Temps en enviar (ms)
	11,42	6,80	3,64	0,69	0,30	5,82
						7.031,02

Figura 43: Gràfic amb missatges enviats i taula amb continguts.

## 6. AMPLIACIÓ

### 6.1. Descripció implementació

Respecte a l'ampliació de capacitats de Garlanet, camí a la implementació de la comunicació un-a-un, algunes de les parts del codi que ja hi eren implementades per la funcionalitat de microblogging han sigut modificades aprofitant la major part possible per no tenir que reprogramar comportaments similars. Tot i que en aquesta part es parlarà únicament de la part que s'ha creat des de zero i per tant és contingut del que es pot parlar com a treball fet, a aquesta part s'anomena el microservei Control.

Aquesta part que s'ha implementat és un servei de control de dades que circulen internament en el sistema, que consisteix en la informació que necessita un usuari A d'un altre usuari B si aquest vol tenir una comunicació directa amb ell o si vol comunicar-se amb un grup on B és l'administrador, aquest segon cas no hi és implantat finalment, però alguns dels camps sí que són comuns.

La taula del Control emmagatzema com ja s'ha comentat valors que necessita un usuari per establir una connexió amb un altre, s'ha pensat que pot ser útil mantenir valors anteriors per tal de poder recuperar valors previs, per tant el que es fa és indexar els valors per temps on s'ha guardat aquest valor, per tant podem tenir valors d'una mateixa clau, un mateix camp, però amb instants de temps diferents. Una vegada ja hem fet una introducció a les dades amb les quals tractem ja es pot veure quins mètodes implementa el microservei Control.

- PUT

Mètode d'inserció de dades a la BD, que està format per tres passos:

1. putRequest

Es demana fer el put, en aquesta acció a línies generals el que fem és reservar el timestamp per inserir el valor que volem. En cas que el timestamp que hem passat sigui inferior al màxim de la taula, l'operació put es cancel·la i es retorna el valor màxim de la taula. És un INSERT a la taula.

2. putApply

Es porta a terme el propi insert a la base de dades afegint-hi el valor a emmagatzemar. És un UPDATE a la taula.

### 3. putConfirm

Es confirma que s'ha fet l'operació anterior correctament seguint uns paràmetres externs a Control que regula i valida el número de rèpliques que es té de la informació. És un UPDATE a la taula.

Per seguir el recorregut dels diferents puts, es fa amb una variable que s'afegeix al putRequest que és un id de l'operació put que engloba els tres passos.

Un altre variable de control per als tres passos és l'estat en que es troba el put, que pren com a valor les tres fases de l'operació, REQUEST, APPLY, CONFIRM, sent aquest últim utilitzat en altres mètodes per obtenir valors o quan busquem valors posteriors o inferiors de la dada que volem donat que són els valors de la taula que estan confirmats o podem dir que són vàlids.

- GET

Operació simple, que donat una key, s'obté el valor més actualitzat, com s'ha dit es pot donar que tinguem diferents entrades amb la mateixa key però els valors seran diferents.

- GET\_IF\_UPDATED

En aquest cas és un get que a més de passar la key que volem, passem el timestamp del valor que tenim, per així comprovar que el valor que tenim és el més actual o hi ha una versió més recent, en cas que tinguem el valor més actualitzat no es retorna res.

- GET\_TIMESTAMP

Una altre versió diferent del GET, en aquest cas no passem la key que volem obtenir, ni cap dels altres anteriors, en aquest cas passem el timestamp del qual volem obtenir informació.

- GET\_PREVIOUS

Com hem vist, guardem diferents valors d'un mateix camp, que es pot anar actualitzant al llarg del temps, per què això tingui sentit, aquesta funció ha d'estar, aquest get consisteix en passar una clau i un instant de temps, llavors el que fem és demanar el

valor que contingui la taula amb un instant de temps anterior al que s'ha passat, aquest valor és el més actualitzat dels anteriors al timestamp que passem.

- **GET\_TIMESTAMP\_CONFIRMED**

Com hem vist les diferents instàncies que té una taula poden tenir estats "inconsistentes", el que vol dir que pot ser que encara el PUT estigui fet a mitges i encara no tingui el valor o no el tingui confirmat, per obtenir els instants de temps que tenen valors que podríem dir vàlids, tenim aquesta funció que obtenim els timestamps de les entrades que tenen l'estat CONFIRMED, per tant que són vàlides.

- **GET\_TIMESTAMP\_PURGED**

Hi ha un quart estat que encara no hem vist i és l'estat PURGED. Que seguint el mateix comportament del mètode que acabem de veure obtenim els instants de temps que tenen una entrada amb aquest estat.

- **PURGE**

Aquest mètode té dues parts:

1. Una primera part on es borra totes les entrades que no siguin CONFIRMED que tinguin un valor CONFIRMED amb un timestamp més actual, que es doni aquest cas vol dir que alguna operació d'UPDATE o INSERT a la taula s'ha quedat a mitges i per tant aquesta entrada ja no serveix a més que ja tenim un valor més nou, que pot ser que sigui el mateix donat que el primer intent de PUT ha fallat i s'ha tornat a fer un altre.
2. Una segona part és una neteja de valors antics, guardem valors antics per poder accedir-hi a ells, però arriba un punt on no té sentit guardar valors tan antics i per tant perquè la taula no creixi infinitament fem una neteja per tenir un màxim de valors d'una mateixa key.

- **KEYS**

Una consulta per obtenir les diferents keys que hi ha a la taula, les taules guarden diferents valors de diferents instants de temps d'una mateixa key, en aquest mètode

obtenim les diferents keys que hi ha a la taula, únicament obtenim les keys per tant sabem que estem guardant però no el valor ni l'instant de temps.

- **GET\_KEYS\_TIMESTAMP**

Amb un comportament semblant a KEYS, aquest mètode obtenim tant les keys com el timestamp de cada entrada.

- **GET\_IF\_UPDATED**

Tenim una segona versió d'aquest mètode, tot i que en aquest cas és diferent. Passem com argument una llista formada per parells de key, timestamp, el que fem en aquesta versió és retornar una subllista d'aquesta que únicament contindrà els parells de key, timestamp que segueixin la següent condició: que la taula on s'ha fet la consulta contingui una instància amb estat CONFIRMED que tingui un timestamp més gran que el que passem com a paràmetre.

- **GET\_HIGHEST\_TIMESTAMP**

Mètode que simplement obté el timestamp més alt de la taula, mètode simple per saber si el valor que volem fer INSERT a la taula es pot fer o és un valor deprecated donat que ja hi ha valors amb un timestamp més alt que no pas el que tenim nosaltres, també s'utilitza a les sessions de consistència, segons el resultat d'aquesta consulta es prenen unes decisions o unes altres depenent de l'operació que estiguem fent.

- **Sessions de consistència**

Abans de veure els dos últims mètodes veiem com són les sessions de consistència donat que tots dos són mètodes que formen part del comportament d'aquesta cal explicar primer el perquè s'utilitzen per així entendre-ho millor.

Una sessió de consistència es iniciada per una rèplica del microservei Control, que anomenarem A, i una altra rèplica del microservei rep aquest missatge de sessió de consistència que anomenarem B, aquesta assignació es fa de forma externa al microservei per tant no hi entrarem. El primer missatge A-B és un timestamp, aquest timestamp és el màxim timestamp de la BD de la rèplica A, llavors B consulta quin és el seu màxim timestamp i a partir d'aquí es divideix en tres possibilitats:



1. La primera possibilitat és que el timestamp de A sigui major que el de B, per tant la rèplica B respon amb el seu timestamp perquè la rèplica A li retorni i tanqui la comunicació amb la informació que li falta a la rèplica B, que correspon a les instàncies que tenen timestamp entre el timestamp de B i el timestamp de A.
2. La segona possibilitat és que el timestamp de B sigui menor que el de B, en aquest cas B ja coneix la informació necessària per actualitzar la rèplica A, almenys amb la informació que ell té, per tant retorna aquesta informació que li falta a la rèplica A per tenir almenys la mateixa informació que la rèplica B i així tancant la comunicació.
3. Per últim el cas en què tots dos timestamps siguin el mateix vol dir que les rèpliques tenen la mateixa informació, en aquest cas la rèplica B retorna nul i acaba la comunicació.

Aquest és el comportament de les sessions de consistència, així és com es manté a les diferents rèpliques actualitzades tot i que com es pot veure es pot donar que hi hagi moments on la informació que hi ha a les diferents rèpliques no fos la mateixa aquest marge de temps serà més gran o més petit depenent de cada quant programem les sessions de consistència, aquestes sessions es fan quan no hi ha activitat a cap dels dos servidors que contenen una rèplica del microservei.

- UPDATE\_CONTROL

Aquest mètode és el que s'invoca quan nosaltres, una rèplica de Control rebem una llista amb la informació actualitzada, i per tant el que fem és actualitzar la nostra BD amb la informació que hem obtingut de l'altra rèplica, com hem vist, l'obtenció d'aquesta informació no ens assegura que tinguem la informació més actualitzada.

- GET\_UPDATED\_CONTROL\_DATA

Per últim aquest mètode és el que cridem a les sessions de consistència quan és el nostre timestamp el més gran de les dues rèpliques del microservei i per tant el que fem és accedir-hi a la nostra BD per agafar la informació que li falta a l'altra rèplica per donar-li la informació que li falta respecte al que nosaltres sabem

## 6.2. Tests de funcionament

Donat que no s'ha posat aquesta part en funcionament amb la resta del projecte, el que s'ha fet per comprovar que funcionés correctament és implementar els mètodes que invoquin a aquests, generant diferents escenaris que s'ha cregut que recullen totes les possibilitats i així estudiar el comportament i veure si es correspon amb el que esperem que succeeixi per quan aquesta part s'implementi amb la resta del sistema sapiguem que es comportarà com volem, aquestes proves s'han fet, quan la taula es buida, quan no hi hagués resultat a retornar i el cas que tot funcione correctament sense cap escenari excepcional.

El cas de les sessions de consistència s'ha fet un estudi diferent, per així estudiar les diferents vies que pot prendre.

### 6.2.1. Sessions de consistència

Ja hem vist anteriorment com funcionen les sessions de consistència per tant hi identifiquem directament els diferents escenaris que es poden donar i com es comporten totes dues rèpliques on una rèplica A envia una sessió de consistència a una rèplica B:

1. Quan la rèplica A té informació menys actualitzada que no pas la rèplica B, seguint l'algoritme explicat abans, la rèplica B veu que el timestamp que A li ha passat és menor que el seu timestamp màxim i per tant fa un `GET_UPDATED_CONTROL_DATA` i retorna a la rèplica A aquestes dades, seguidament A mitjançant `UPDATE_CONTROL` actualitza la seva taula amb la informació que B li ha passat.
2. El cas mirall, on B és qui té informació menys actualitzada, rep el timestamp de A i es dóna compta que és més gran que no pas el màxim de la seva taula, llavors retorna a A el seu timestamp més alt, i es A qui en aquest cas fa `GET_UPDATED_CONTROL_DATA` per obtenir la informació que li falta a B respecte a la que té A i li envia a B perquè aquest mitjançant `UPDATE_CONTROL` actualitzi la seva taula per tenir la mateixa informació que A.
3. Un altre escenari que hem volgut evitar és on la taula que envia informació sigui tant A com B, eviti les entrades amb estat `PURGED`. Com ja hem vist són entrades que eventualment seran eliminades i per tant no té sentit que gastem esforços en mantenir

rèpliques d'aquestes, entrades amb qualsevol altre estat es creu que sí que han de ser replicades donat que pot ser que el o els passos anteriors a l'insert s'hagin perdut i que canvis d'estat posteriors sí que hi arribin i llavors acabar de fer el PUT. En cas que aquestes entrades no acabin de completar-se s'acabaran purgant, per tant no és un problema.

4. S'han nomenat i descrit els diferents escenaris que succeeixen amb normalitat, per últim veurem que és el que passa quan aquest procés es veu interromput en diferents punts de la seva execució i com es comporta el sistema i si aguanta aquestes excepcions o no. Aquestes rèpliques s'allotgen en els servidors que ja hem vist, per tant es pot donar que a mitja sessió de consistència aquest servidor deixi d'estar operatiu i que s'apagui de cop i volta, per tant estudiem els diferents casos que es poden donar si es dóna el cas que el servidor deixi de funcionar per aquesta raó o qualsevol altre:

- El primer escenari és en el moment on A envia el primer missatge obrint la sessió de consistència, recordar que aquest missatge conté el seu timestamp més recent, A no rep cap resposta, al cap del temps dona per acabat aquesta sessió i inicia una altra sense cap problema.
- Relacionat amb el primer escenari, qualsevol altre punt de l'execució de la sessió de consistència que no sigui en els mètodes de GET\_UPDATED\_CONTROL\_DATA o UPDATE\_CONTROL es resolen de la mateixa manera que el primer, es dóna per finalitzada la sessió de consistència al cap del temps donat que no es rep resposta, en cas que sigui el punt on A envia la informació que li falta a B, un cop A envia aquesta informació es desentén per tant A no s'adonaria que B no s'ha actualitzat, cas que tampoc és un problema.
- Arribem al primer cas on pot ser delicat, estem fent la consulta, GET\_UPDATED\_CONTROL\_DATA, en qualsevol dels dos casos, ja sigui A o B el comportament és el mateix, donat que com estem fent una consulta, el fet que es doni una parada d'execució inesperada no afecta a cap de les dues rèpliques, únicament que no es farà cap actualització on calgués.
- Per últim tenim el cas que pot ser conflictiu, quan estiguem fent UPDATE\_CONTROL, si ja estem fent aquesta operació, vol dir que la connexió amb

l'altra rèplica ja s'ha tancat donat que ja s'ha resolt l'escenari on ens trobàvem, ja fos A o B qui li faltés informació, la comunicació entre totes dues rèpliques ja no hi es activa, per tant només afecta a la rèplica en qüestió que s'estigui actualitzant. I en què l'afecta? En el fet que no podrà tenir la mateixa informació que tenia l'altra rèplica, per tant fins que aquesta rèplica no comenci una sessió nova o sigui objectiu d'una, no actualitzarà el contingut que no ha pogut arribar a actualitzar. Això no és un problema, donat que encara que hagués completat l'actualització no garanteix que tenia la informació més actual, donat que es podia donar el cas que la rèplica que li havia passat la informació també estigues desactualitzada, per tant no és un inconvenient, si torna a rebre una sessió de consistència de la mateixa rèplica llavors en aquesta segona instància, si no sorgeix un altre imprevist, actualitzarà la part de la informació que li faltava.

## 7. SOSTENIBILITAT I COMPROMÍS SOCIAL

Per començar es defineix el que vol dir sostenibilitat i que vol dir que un projecte és sostenible, cita de diccionari de L'institut d'Estudis Catalans: 2 adj. [EG] [LC] Que es pot mantenir indefinidament, especialment sense afectar l'equilibri ecològic. Per tant que un projecte sigui sostenible no únicament vol dir que els recursos que utilitza siguin infinits sinó que hi han altres factors que hem de tenir en compte. Per determinar el nivell de sostenibilitat del projecte es fan tres divisions on s'estudiarà en cada divisió uns factors en concret.

Sostenibilitat **social**, factors que determinen si el projecte fomenta les relacions entre individus i fomentant el creixement social i respecte al benestar social, creant llocs de treball i protegint la seguretat i salut de les persones i reduint la pobresa.

Sostenibilitat **econòmica**, factors que controlen que aprofitem els recursos dels quals disposem al màxim i que utilitzem, per tant sent molt estrictes amb l'eficiència amb que tractem els recursos, inputs, del nostre projecte. Cuidant en particular també els escassos per garantir un futur per aquests recursos.

Sostenibilitat **ambiental** principalment consisteix a aconseguir un creixement satisfent les necessitats socials, econòmiques i ambientals sense comprometre les futures generacions. I

no és únicament que aquestes tecnologies siguin renovables sinó que també siguin netes, ja que no únicament s'ha de preocupar per deixar recursos a les següents generacions sinó que també hem de deixar el planeta sense danys irreparables.

Per tant hem de tenir en compte en els nostres productes els diferents paràmetres dels tres àmbits de la sostenibilitat amb idees creatives per utilitzar tecnologies sostenibles i aplicar mesurador per veure que tan sostenible és, així com tracta la justícia social, equitat i diversitat, com afecta el bé comú de la societat i la reutilització de recursos i com aplicar l'economia circular per reduir al màxim els indicadors d'impacte i altres conseqüències i ampliant el més possible els diferents impactes positius a la societat podent utilitzant solucions d'altres productes sostenibles.

## **7.1. Dimensió Ambiental**

És la comunitat de Garlanet la que de manera directa ajuda a reduir l'impacte ambiental evitant que s'hagi de construir un espai que reculli els servidors del sistema, proporcionant les màquines que disposen, si volen. No es creu que en l'àmbit de recursos utilitzats es pugui fer una reducció que porti a un impacte menor en el medi ambient lo suficientment notable com per poder-se quantificar, únicament podríem dir que el temps que es va dedicar a fer tests podrien haver sigut menors per evitar tenir en funcionament les màquines.

Per evitar haver de construir un edifici per poder allotjar els servidors, la solució que s'utilitza és reacondicionar llocs que estiguin en desús i que recullin les característiques que ha de tenir un espai on tenir servidors.

En el nostre cas no ens cal tenir un espai on recollir els servidors, ja que el sistema està format per diferents equips de la pròpia comunitat i per tant no sabem on els col·loquen i quines característiques tenen, per tant tampoc sabem si el maquinari on el nostre sistema està corrent és respectuós o no amb el medi ambient.

Els recursos que es poden arribar a utilitzar durant la vida útil del projecte són la maquinària per mantenir el sistema en funcionament, per tant identifiquem aquestes màquines juntament amb els recursos que es necessitin per mantenir aquestes màquines en funcionament i l'espai físic on se situen. Per tant són recursos que en un principi ja s'estaven utilitzant tot i que tant en el cas de la maquinària, que al estar més temps treballant o tenint una càrrega de

treball major que abans tingui més probabilitats de tenir una fallada de hardware/software i es necessiti reparació o compra d'una nova, i en el cas de la llum que s'elevi el consum d'aquest recurs donat que la deixem més estona encesa que de costum per aportar els recursos a la xarxa. Llavors l'impacte d'aquestes dues variables depèn molt del que es fagi amb la maquinaria, o part d'aquesta, un cop no serveixi i del proveïdor de llum, que són factors externs a nosaltres.

No es creu que poguem substituir a empreses que utilitzen una estratègia centralitzada o sistemes molt més robust que impliquen molta més maquinaria, per tant no es creu que es pugui reduir l'empremta ecològica, però el que sí que podem dir és que una solució molt més sostenible amb el medi ambient.

Es creu que l'únic cas que es podria donar és que el fet de aportar les màquines a Garlanet accelerés el deteriorament de aquestes i que donés més empremta a generar més basura tecnològica.

## **7.2. Dimensió Econòmica**

Com s'ha comptabilitzat a l'apartat d' Identificació i estimació dels costos, s'han definit els diferents perfils de treballadors i les diferents quantitats d'hores que desenvolupen cadascun d'ells, també s'ha estimat altres tipus de costos com els materials d'oficina com els de hardware i els costos d'espai per treballar, llum i connexió a internet. No s'han pres decisions per reduir aquests costos, donat que són els costos necessaris per portar a terme el projecte.

Quant a la solució actual a aquests problemes el que portem com a millora de Garlanet és el fet de necessitar tant poca infraestructura per poder posar en marxa el sistema i a més no hem d'invertir en construcció de cap tipus d'espai per guardar els servidors, ja que són els mateixos usuaris els que proporcionen la poca infraestructura que es necessita. No es pot monetitzar els costos que això comporta, donat que no sabem quanta maquinaria hi haurà donat que no la aportem nosaltres, si no els usuaris, i no sabem cada quant aquesta maquinaria s'ha de renovar o recondicionar, l'únic que podríem fer per reduir aquest cost és que es necessites encara menys infraestructura, però no es creu que les millores que es puguin arribar a fer siguin lo suficientment importants com per poder reduir de forma quantificable la infraestructura necessària.

Un escenari que es podria donar es que cap usuari de la xarxa o un nombre insuficient aportés recursos a la xarxa el que fes que no poguem sostenir el nombre d'usuaris que volguesin utilitzar la xarxa.

### **7.3. Dimensió Social**

A nivell personal el projecte em donarà una experiència en projectes de gran escala i la continuïtat de treballar en un projecte a llarg termini. Altres factors també generals respecte a projectes és la gestió de diferents tasques de gran quantitat d'hores al llarg del temps i capacitat d'expressar i d'entendre idees tant en la part de plantejament, disseny i implementació. Per últim adquirir la capacitat d'entendre grans quantitats de codi, ja fet i a partir d'una base construir-hi sobre d'aquesta.

S'ha après coses com la reorganització de un projecte de llarg termini i tenir que prendre decisions, no únicament sobre desenvolupament intern del projecte, sinó fer un anàlisi del que havíem plantejat de fita inicial i segons ha avançat el projecte com han anat afectant diferents variables, algunes que es tenien presents i d'altres que no.

A nivell social Garlanet és una solució diferent, ja que el mateix sistema està format per maquinari que els mateixos usuaris hi proporcionen de manera voluntària, com a solució més o menys semblant que hi ha actualment és l'emmagatzematge d'informació al mateix dispositiu per poder fer consultes sense connexió o passar aquesta informació per mitjans de comunicació de curta distància, però no té el mateix enfocament, únicament el fet d'utilitzar part dels dispositius de la mateixa comunitat.

No es creu que existeixi una necessitat real, però com tota solució tecnològica a un problema creiem que tenim un enfocament on la nostra solució sigui l'adequada i la més profitosa. Associacions sense ànime de lucre són comunitats que poden estar molt interessades en la nostra solució, per una altra banda empreses de gran calibre poden estar interessades a desplegar la seva pròpia xarxa de Garlanet, ja que amb poca infraestructura es poden donar ells mateixos el servei sense dependre d'uns servidors externs.

Un cas que es podria donar és la monopolització dels recursos, sent una única persona o

grup de persones qui aporta la major part dels recursos i per tant arribar a centralitzar gran part de la informació, i intentar treure'n profit d'aquesta monopolització d'alguna manera. Tot i que aquesta situació, en cas que portes algun risc en quant a la idea de Garlanet es podria intervenir impedit que aquestes màquines segueixin donant servei, tot i que podria ser un problema si aquestes han recollit tanta informació que no es pugués replicar, en cas que no hi sigues, en altres màquines.

#### **7.4. Reflexió personal**

Com a reflexió personal em motiven molt els projectes i reptes que tenen una idea o finalitat d'aportar un bé a la societat i poder millorar la qualitat de vida de les persones sense afectar a la qualitat d'altres. A més també ha de portar un bé a mi com a persona i també de manera econòmica a l'empresa, per tant no desapropitar els recursos i fent les coses amb cura i amb una planificació prèvia i la manera que crec que s'ha de treballar. Per últim en l'àmbit ambiental no hi he estat involucrat en casos que hi pugués tenir un impacte ambientalment però crec que hi soc bastant conscient dels impactes que te no respectar el medi ambient i la gestió de recursos gràcies a diferents sessions informatives que s'han fet al llarg del grau.

## **8. CONCLUSIONS**

### **8.1. Tests**

Farem una avaluació per cada paràmetre que s'ha decidit evaluar per veure, si el valor és òptim en cas d'evaluar el sistema i comparar aquest valor amb el que s'ha obtingut fent un test en remot, donat que les comunicacions entre processos en local és molt més ràpida que amb components que es troben fora de la nostre xarxa local.

- Persones seguides amb èxit.

És un valor que s'ha d'evaluar tenint en compte les diferents causes que pot donar que una persona no segueixi a una altre, l'únic cas que podem parametritzar és el percentatge de usuaris que s'han registrat, donat que si un usuari que seguim no s'ha registrat no el podem seguir, revistant parells de valors podem dir que contra més aprop del 100% estigui el percentatge d'usuari que s'han registrat amb èxit major és el



percentatge de persones que seguim respecte al total.

- Registrats

Un valor que com s'acava de veure te molta relació en quant a extraure dades amb les persones seguides, el nombre de registrats és bastant alt, idealment no hauria de fallar cap registre donat que es va fer els tests entrant els usuaris en onades de manera seqüencial evitant que el sistema colapsés, tot i així s'han donat alguns casos.

- Distribució de seguidors.

Ja s'ha parlat sobre les distribucions dels seguidors i seguits, en aquest cas es volia veure quanta variació ens trobem en els percentatges respecte a les distribucions que establim en un principi. Els valors s'han tingut que modificar respecte al que estava pensat i per tant l'accentuació de la variança entre el que hi havia establert i el que queda és major que és el que s'esperava.

- Missatges.

Una dada molt positiva, tots els tests han tingut un molt bon rendiment en quant a enviament de missatges, el Test Local 4, va arribar un punt en el que el sistema va entrar en un estat de inconsistència i donat aixó hi ha l'alt percentatge de missatges que no es van enviar, tot i aixó es va voler incloure en els tests donat que el sistema es va recuperar i es van enviar uns missatges més abans de finalitzar el test.

## 8.2. Millores

- Més tests.

Degut al canvi de planificació els tests es van tenir que fer en un període de temps on no va donar temps a poder fer la quantitat de tests que es voldria fer, tot i així durant el desenvolupament del TFG es van anar fent tests per provar les funcionalitats, pero sense resultats amb els que comparar.

- Grandària dels tests.

No només va ser un problema el temps donat que els tests que es van fer es van tenir

que fer amb pocs usuaris donat que per alguna raó el sistema no aguantava tenir més usuaris i cap usuari a partir d'un cert número d'usuaris deixava de registrar els usuaris. Més probablement, el problema del registre dels usuaris podria ser el límit d'usuaris per màquina, donat que als servidors no podem executar usuaris per problemes de versions de software els usuaris es van tenir que executar tots en local i la resta del sistema als servidors, potser reduint el nombre d'usuaris per màquina haguéssin funcionat els tests amb més usuaris.

- RepositoryManager

En quant a comparació amb un desplegament real de Garlanet aquesta és la major diferència, els tests que s'han fet han sigut amb *repositories* que eren perfectes, sent màquines que ofereix la pròpia comunitat s'ha d'estar preparat perquè els *repositories* deixin de funcionar i tornin a arrencar-se de manera inesperada, això no s'ha pogut avaluar.

- Altres avaluacions.

Altres valors que podrien haber estat interessants d'avaluar són, la relació entre fallades dels *repositories* i el nivell d'activitat com afecten aquestes dues variables, relació entre la fallada d'un post i el pes del missatge o missatges que ens arriben.

### 8.3. Personal

Com a conclusió personal respecte al TFG, ha sigut un treball que ha tingut molts canvis respecte el que es va plantejar i es volia fer al principi fins el que s'ha obtingut finalment. La reflexió que faig respecte això és la capacitat de adaptar-se als diferents canvis i inconvenients que han anat sorgint per poder presentar un TFG i no en la diferència entre el que es volia fer i el que s'ha acabat fent.

Altres reflexions com l'aprenentatge que s'ha adquirit al llarg d'aquest temps no només en programar més bé en adquirir la forma i filosofia d'un projecte ja en marxa i fer un treball a partir d'aquest, això ha sigut la part on més he après, la comprensió d'un sistema i afegir-hi part de codi seguint la filosofia de la resta del sistema.

## BIBLIOGRAFIA

[1]Pàgina web Garlanet. [No disponible actualment] [Consulta: 26 Jun. 2020]

<<http://dpcs.uoc.edu/projects/garlanet/>>

[2]Pàgina web de la UOC. [En línia] [Consulta: 19 Feb. 2020]

<<https://www.uoc.edu/portal/es/index.html>>

[3]Grup de recerca ICSO. [En línia] [Consulta: 24 Feb. 2020]

<<http://transfer.rdi.uoc.edu/ca/grup/internet-computing-systems-optimization>>

[4]Grup de recerca KISON [En línia] [Consulta: 24 Feb. 2020]

<[https://www.uoc.edu/portal/ca/in3/recerca/grups/kriptography\\_and\\_information](https://www.uoc.edu/portal/ca/in3/recerca/grups/kriptography_and_information)>

[5]Entrada de Microblogging a Wikipedia. [En línia] [Consulta: 23 Feb. 2020]

<<https://es.wikipedia.org/wiki/Microblogging>>

[6]Diccionari Català. [En línia] [Consulta: 22 Feb. 2020]

<<https://dlc.iec.cat/>>

[7]TechRepublic. [En línia] [Consulta: 23 Feb. 2020]

<<https://www.techrepublic.com/article/top-5-decentralized-social-networks/>>

[8]Diaspora. [En línia] [Consulta: 23 Feb. 2020]

<<https://diasporafoundation.org/>>

[9]Minds. [En línia] [Consulta: 23 Feb. 2020]

<<https://wefunder.com/minds>>

[10]Mastodon. [En línia] [Consulta: 23 Feb. 2020]

<<https://joinmastodon.org/>>

[11]Sola. [En línia] [Consulta: 23 Feb. 2020]

<<https://sola.ai/https://sola.ai/>>

[12]Scuttlebut. [En línia] [Consulta: 23 Feb. 2020]

<<https://scuttlebutt.nz/>>

[13]Manyverse. [En línia] [Consulta: 23 Feb. 2020]

<<https://www.manyverse.se/>>