



Devilish: Una plataforma de desenvolupament de programari

Oscar Rubio Garcia
Grau d'Enginyeria Informàtica
Java EE

Albert Grau Perisé
Santi Caballe Llobet

14/06/2017

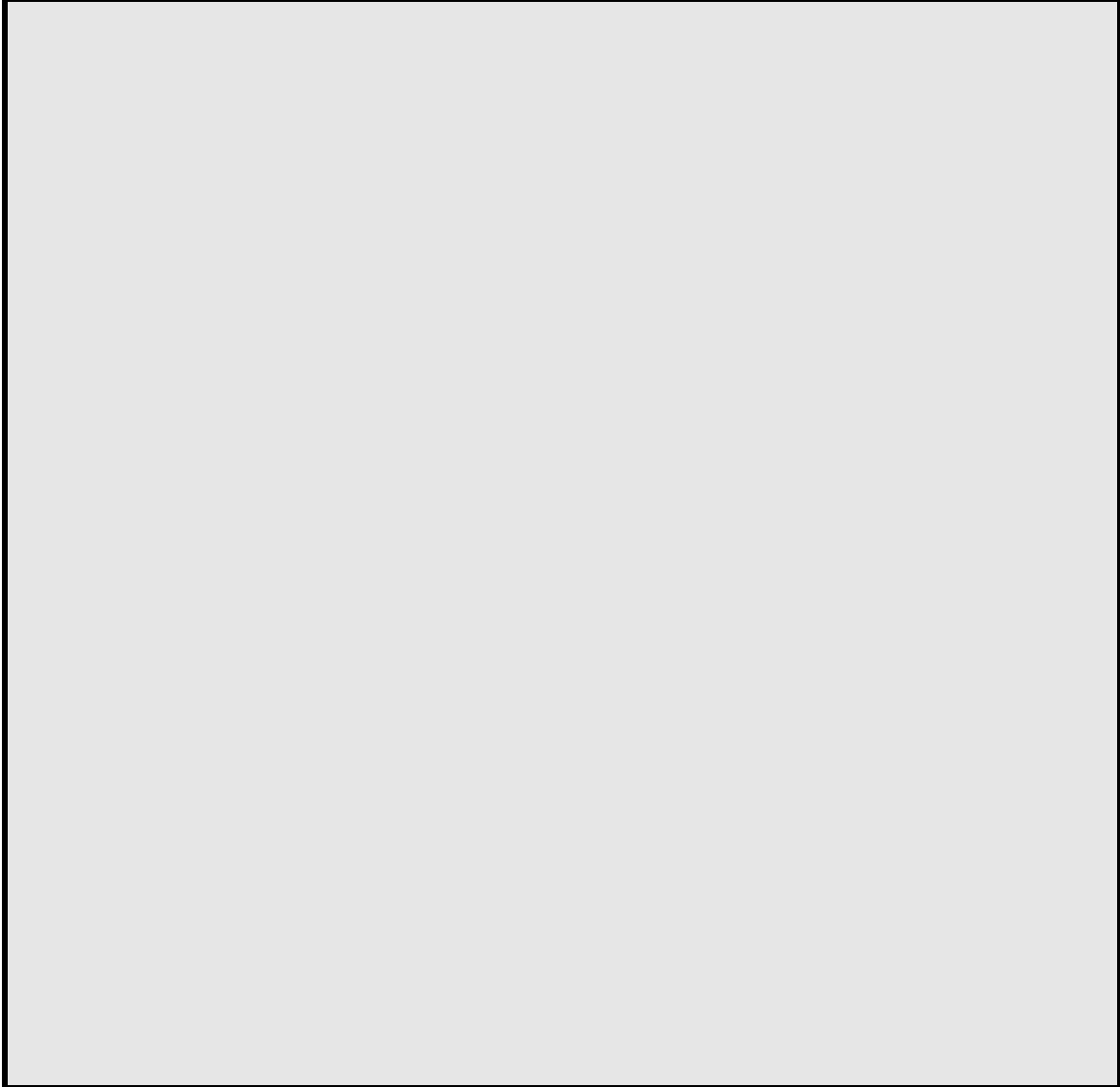


Aquesta obra està subjecta a una llicència de [Reconeixement-NoComercial-SenseObraDerivada 3.0 Espanya de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

FITXA DEL TREBALL FINAL

| | |
|---|--|
| Títol del treball: | <i>Devilish: Una plataforma de desenvolupament de programari</i> |
| Nom de l'autor: | <i>Oscar Rubio Garcia</i> |
| Nom del consultor/a: | <i>Albert Grau Perisé</i> |
| Nom del PRA: | <i>Santi Caballe Llobet</i> |
| Data de lliurament (mm/aaaa): | <i>06/2017</i> |
| Titulació o programa: | <i>Grau d'Enginyeria Informàtica</i> |
| Àrea del Treball Final: | <i>Java EE</i> |
| Idioma del treball: | <i>Català</i> |
| Paraules clau | <i>TFG, devilish, software</i> |
| Resum del Treball (màxim 250 paraules): <i>Amb la finalitat, context d'aplicació, metodologia, resultats i conclusions del treball</i> | |
| <p>Devilish té com a finalitat ser una plataforma gratuïta de desenvolupament de programari, que no depengui de la tecnologia o llenguatge de programació. Pretén ser un conjunt d'eines integrades que millorin els processos i la productivitat dels equips de desenvolupament.</p> <p>D'eines de desenvolupament de programari podem trobar en el mercat una gran varietat i quantitat. Devilish no pretén, ni pot, competir comercialment amb cap d'aquestes, sinó que pretén ser una alternativa gratuïta, ja sigui per a ús personal o per a distribució a tercers.</p> <p>S'ha intentat seguir una planificació marcada per les fites del TFG, on s'ha estudiat diferents propostes existents i s'ha recollit una sèrie de requisits funcionals que s'ha cregut convenient que Devilish tingués. Un cop recollits s'ha fet una sèrie de spikes tecnològics per decidir les tecnologies a fer servir, ja sigui tenint en compte els temps del TFG o les més adients. Arribat al punt de la implementació s'ha intentat seguir una metodologia àgil que ens a permès variar segons els canvis introduïts.</p> <p>El resultat no ha sigut ni de bon tros l'esperat, ja que la idea era tenir un producte mínim viable, amb les mínimes funcionalitats implementades. Però per falta de temps i diferents desviacions i contratemps s'ha quedat molt lluny. Tot i això, com s'explicarà, Devilish, en el context del TFG és simplement la llavor d'un projecte més gran, real, i complet, pel que ha servit per plantar aquesta llavor i fer un primer anàlisi de per on ha d'evolucionar el projecte.</p> | |

Abstract (in English, 250 words or less):



Índex

| | |
|--|----|
| 1. Introducció..... | 1 |
| 1.1 Context i justificació del Treball | 1 |
| 1.2 Objectius del Treball..... | 2 |
| 1.3 Enfocament i mètode seguit | 2 |
| 1.4 Planificació del Treball..... | 2 |
| 1.5 Breu sumari de productes obtinguts | 5 |
| 1.6 Breu descripció dels altres capítols de la memòria | 5 |
| 2. Introducció a Devilish | 6 |
| 2.1 Visió del producte | 6 |
| 3. Abast del producte | 6 |
| 2.1. Producte mínim lliurable | 6 |
| 2.2. Altres funcionalitats | 8 |
| 2.3. Funcionalitats futures | 8 |
| 3. Documentació de requisits | 9 |
| 3.1. Stakeholders | 9 |
| 3.2. Diagrama de casos d'ús | 9 |
| 3.3. Actors | 9 |
| 3.4. Fitxes de casos d'ús | 10 |
| 4. Diagrama del model de entitats..... | 24 |
| 5. Arquitectura | 25 |
| 6. Tecnologies i eines..... | 27 |
| 6.1. Llenguatges..... | 27 |
| 6.2. Plataforma de desenvolupament..... | 27 |
| 6.2.1. Back-end | 27 |
| 6.2.2. Front-end..... | 28 |
| 6.2.3. Apps | 28 |
| 6.3. Bases de dades..... | 28 |
| 6.4. Eines, frameworks i llibreries | 29 |
| 7. Model de pantalles | 31 |
| Pàgina d'inici | 31 |
| Pàgina d'autenticació | 32 |
| Taulell del projecte | 32 |
| 8. Conclusions..... | 33 |

| | |
|-----------------------|----|
| 9. Glossari | 34 |
| 10. Bibliografia..... | 35 |
| 11. Annexos | 36 |

Llista de figures

No se encuentran elementos de tabla de ilustraciones.

1. Introducció

1.1 Context i justificació del Treball

Amb el pas dels anys el desenvolupament de programari ha anat madurant els seus processos i metodologies, han aparegut diferents branques i especialitzacions que tracten diversos aspectes d'aquest que abans no es tenien en compte o directament es desconeixien, o no s'entien com a quelcom separat, i que amb el pas del temps s'han mostrat indispensables, també s'han popularitzat conceptes com ara agile, UX, taulells, historia d'usuari, bones pràctiques, patrons, SOLID, integració continua i molts d'altres.

Tots aquest nous i no tan nous conceptes si s'implementen bé milloren la productivitat dels equips, la qualitat del codi, la gestió dels projectes, la reutilització, etc., i al final això comporta una millora en els beneficis del projecte i en la satisfacció dels equips i, molt important, en la satisfacció dels clients.

Per tot això, les empreses i els equips de desenvolupament de programari han pres consciència de la necessitat d'incorporar moltes d'aquestes pràctiques al seu ideari, però la tasca no és senzilla, aconseguir implementar tot això seria una quimera sense l'ajut d'eines informàtiques especialitzades que ens facilitin la tasca. Existeixen multitud d'eines: en tenim de control de versions, d'anàlisis de qualitat del codi, de gestió del cicle de vida, de gestió documental, testing, etc.

També hi ha productes que integren moltes d'aquestes eines unificant-les en una única plataforma, el que facilita el treball als equips i ajuda a les empreses en la compra e implantació, a més d'afavorir els processos i la integració entre les diferents eines que componen la plataforma.

Durant anys d'experiència desenvolupant programari he tractat amb multitud d'aquestes eines, i moltes vegades m'ha tocat escollir la més apropiada segons les necessitats i requisits del projecte, de l'equip o de l'empresa. Aquest projecte té com a finalitat la creació d'una plataforma que integri diferents d'aquestes eines, o si més no es tracta de crear un producte mínim que sigui l'embrió d'aquesta plataforma, començant per la gestió del cicle de vida.

La motivació ve donada després de treballar amb productes com JIRA o Visual Studio Team Services, productes molt bons, complets i estesos, però que no han satisfet, per una cosa o altra, les necessitats que m'he trobat en la cerca abans esmentada en múltiples ocasions. En el cas de JIRA, per la seva influència del ticketing provinent dels seus orígens HelpDesk juntament amb una complexitat moltes vegades no necessària, mentre que en el cas del VSTS el seu lligam a .NET (tot i que sembla que això està canviant) i a la seva plataforma online (no podem crear el nostre servidor propi) i de pagament quan es tracta d'equips mitjans o grans el fan

poc atractiu en ambients multiplataforma, multilinguatge, etc. Altres eines menys esteses o menys conegudes també s'han valorat, de vegades el preu era l'obstacle, de vegades les funcionalitats ofertes, la discontinuïtat del projecte, etc.

Evidentment la plataforma que es vol crear no serà competidor en funcionalitats d'aquests dos monstres, com ja s'ha dit la funcionalitat principal serà la gestió del cicle de vida. La finalitat és crear una plataforma no lligada a cap tecnologia, que sigui completa o parcialment gratuïta, i que ajudi a empreses i equips en les tasques més bàsiques e importants del desenvolupament de programari, sense afegir complexitats innecessàries, i basada en les pràctiques i metodologies actuals, incorporant el conceptes agile i bones pràctiques en el seu ADN.

1.2 Objectius del Treball

L'objectiu principal és posar en pràctica en un projecte real els coneixements adquirits durant les diferents assignatures relacionades amb l'àrea de Java EE del Grau d'Enginyeria Informàtica, i alhora aconseguir la màxima destresa en l'aplicació d'aquests. Per tal de fer això, la materialització d'aquest objectiu resultarà en la consecució del producte mínim descrit en apartats posteriors i amb la màxima qualitat possible.

Com a objectius secundaris es té la possibilitat de crear un producte que sigui continuat més enllà del projecte i que personalment em serveixi d'eina global per als meus equips. Un pas més enllà, es té com a objectiu que sigui viable per a la seva distribució en el mercat com a plataforma que pugui ajudar a altres empreses que tinguin la necessitat.

1.3 Enfocament i mètode seguit

Amb la consecució del projecte es pretén crear un producte nou, però que alhora estigui molt influït pels referents existents actualment en mercat. És a dir, la idea és agafar totes les funcionalitats que agraden d'altres productes existents eliminant les no desitjades i "millorant" les incorporades.

Per tal de fer aquest procés s'ha estudiat els diferents productes ja existents, com ara Jira, Visual Studio Team Services o d'altres, i a partir d'aquí s'ha creat una llista de funcionalitats que es volen incorporar al nostre producte.

Un cop començada la implementació s'ha decidit seguir una metodologia àgil semblant a Scrum per tal d'adaptar-nos ràpidament als possibles canvis introduïts per la incertesa inicial en quan a tecnologia que teníem i a possibles canvis funcionals que s'han considerat.

1.4 Planificació del Treball

El següent diagrama mostra la planificació del projecte. S'han posat diferents fites segons les diferents entregues i tenint en compte el producte mínim lliurable.

Durant tot el procés de desenvolupament de la plataforma es farà servir una metodologia àgil, més concretament Scrum, o una variació d'aquesta.

Tot i que l'enunciat ens demani la documentació del requisits en fitxes de casos d'ús, es traslladarà també aquestes a històries d'usuari per tal d'adaptar-los a la metodologia utilitzada.

Els processos d'enginyeria de requisits que es donaran inicialment són l'obtenció (en el nostre cas revisió) dels requisits, el seu anàlisi i la documentació d'aquests. Després durant la PAC3 es començarà un procés iteratiu (sprints) que ens portarà a fer la gestió completa de requisits, incidint sobretot en la priorització i estimació al principi de les iteracions, i la validació al final. I en el cas de acabar el producte mínim afegint al backlog els requisits de les funcionalitats extres.

Tot seguit, en la propera pàgina, es mostra el diagrama de Gantt de la planificació inicial del projecte. Aquesta planificació s'ha vist alterada per els inconvenients que han anat sortint durant el transcurs del projecte.

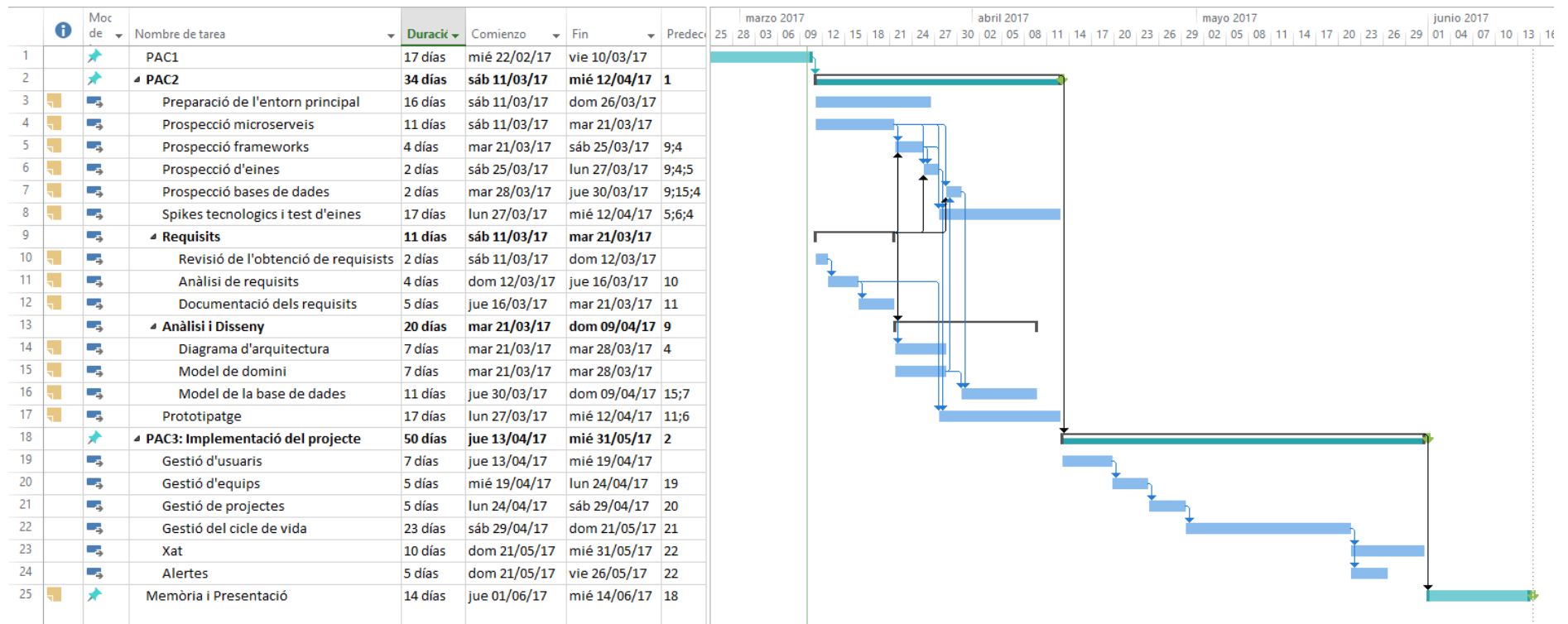


Diagrama de Gantt de la planificació del treball

1.5 Breu sumari de productes obtinguts

Els artefactes obtinguts en aquest projecte són el propi producte desenvolupat, distribuït a través d'un arxiu `.ear`. Aquesta distribució serà publicable en un servidor d'aplicacions i web que sigui capaç de servir components Java EE.

El producte, Devilish, és una aplicació desenvolupada amb Java EE, composta d'una capa de persistència sobre JPA/Hibernate, una capa de negoci sobre EJB, i que disposa d'una capa de presentació web JSF i també d'una capa de serveis REST per a connectar qualsevol altre tipus d'aplicació client, com ara una SPA o una aplicació mòbil.

Les característiques del producte es detallaran amb més detall en els següents apartats.

Com a complement s'adjunta el projecte Eclipse de desenvolupament del producte, és a dir, el codi del producte, i un arxiu de Postman per a testing dels serveis REST.

1.6 Breu descripció dels altres capítols de la memòria

En els següents capítols s'exposarà el projecte amb més detall i de forma més específica.

El capítol "Introducció a Devilish" pretén fer una aproximació general al producte, exposant la formulació de la visió del producte, tot seguit el capítol "Abast del producte" exposarà les funcionalitats de la plataforma, per seguir amb la documentació dels requisits en el capítol "Documentació de requisits".

Els següents capítols ja es tornen una mica més tècnics, començant per el "Diagrama del model d'entitats", on es mostren les entitats i les seves relacions existents en el model de domini del sistema. El capítol "Arquitectura" ens mostra una visió ràpida de l'arquitectura de l'aplicació i de les decisions preses. Continua el capítol "Tecnologies i eines" que fa un repàs a totes les tecnologies i eines que s'han utilitzat o que es pretén utilitzar en les properes fases.

Finalment, acabem amb una mostra reduïda dels prototips de les principals pantalles que es van dissenyar abans de fer la implementació.

2. Introducció a Devilish

Devilish és un projecte que té com a inici el TFG que s'exposa en aquest document, però el desenvolupat aquí només pretén ser la llavor d'un projecte més gran i, al final de tot, d'un projecte real, que pugui arribar a fer-se servir, ja sigui per mi mateix en els meus desenvolupaments o com a distribució a tercers.

Per tant, un cop acabat aquest TFG el desenvolupament continuarà en les condicions que es pugui, ja sigui amb uns timings o uns altres.

Tot el desenvolupat està subjecte a refactoritzacions i canvis de tecnologia o arquitectura en aquesta primera fase, però s'ha intentat fer el millor possible en els temps de que s'ha disposat per a poder aprofitar el màxim possible un cop superat el TFG.

2.1 Visió del producte

Per a empreses i equips de desenvolupament de programari que volen millorar la productivitat en els seus projectes, Devilish és una plataforma de desenvolupament de programari que integra múltiples eines que faciliten els processos i pràctiques implicades, com ara la gestió de projectes i equips, la gestió del cicle de vida, o la comunicació, entre d'altres, i així repercutint en uns millors temps d'entrega i beneficis reportats pel projecte. A diferència de JIRA o VSTS, Devilish és una plataforma gratuïta, no lligada a cap tecnologia o llenguatge concret, orientada purament al desenvolupament de programari, per a tot tipus d'equips i que segueix de manera clara la filosofia agile.

3. Abast del producte

En aquest apartat es descriurà l'abast del projecte, és a dir, els requeriments, tan funcionals com de qualitat, que el projecte incorporarà. Aquesta descripció serà una descripció a alt nivell, encara no s'entrarà en detall, per exemple, definint casos d'ús, tasca que forma part de la següent fase.

2.1. Producte mínim lliurable

El producte mínim lliurable incorpora l'abast mínim que compleix amb la visió del producte i que, en aquest cas, es considera que ha d'entrar per a la consecució del projecte.

Com s'ha comentat en apartats anteriors la "llavor" de la plataforma serà la gestió del cicle de vida del projecte, abraçant les metodologies àgil, i per tant el producte mínim estarà centrat en aquest tipus d'eina.

S'incorporarà, evidentment, temes organitzatius d'equips i projectes, i com a extra un xat senzill per aquesta primera versió.

Per tant, es pot dir que els principals components o blocs de funcionalitat que tindrà la plataforma a implementar són els següents:

- Gestió d'usuaris: Aquest component serà l'encarregat de tot el referent a l'autenticació i autorització de l'aplicació, a la gestió dels usuaris, perfils d'aquests últims, etc.
- Gestió d'equips: La plataforma ha de permetre la creació d'equips i la seva gestió. Aquests equips estaran formats per usuaris de la plataforma, i podran tenir el seu propi espai, crear projectes, etc.
- Gestió de projectes: Aquest component serà el referent a tot el relacionat amb la gestió de projectes.
- Gestió de cicle de vida: Encarregat de tot el referent a la gestió del cicle de vida, i ens permetrà les següents funcionalitats mínimes:
 - Creació de taulells Kanban o Scrum mitjançant plantilles, ja que la intenció és que sigui ampliable en un futur amb altres metodologies.
 - Creació i gestió d'històries d'usuari, tasques, èpiques, etc., depenent de la plantilla escollida. És a dir, dels elements necessaris e implicats en la metodologia.
 - La gestió del backlog, sprints, etc.
 - També s'afegirà una sèrie d'eines de seguiment típiques com ara el burndown chart i alguna més encara per determinar si es creu convenient.
- Xat col·laboratiu: S'ha cregut convenient integrar una eina col·laborativa de xat per als equips de desenvolupament creats en la plataforma. Aquest tipus d'eines com ara Slack o HipChat s'han convertit en essencials per augmentar la productivitat mitjançant una millora en la comunicació de l'equip. L'eina incorporarà missatges privats, creació de rooms i mencions entre altres.
- Alertes: Finalment també es creu bàsic implementar un sistema d'alertes, que per aquesta primera versió es basarà en alertes per correu electrònic.

A part, d'aquestes funcionalitats, del producte resultant s'espera altres requisits de qualitat o no funcionals, com ara mantenibilitat, escalabilitat, rendiment, usabilitat, seguretat, etc. Per tal d'aconseguir-ho es té com a objectius l'aplicació de les millors bones pràctiques sabudes, com ara patrons, principis SOLID i altres, implementació de l'arquitectura més adient, bona organització del codi, concurrència, etc.

Un cop s'hagi aconseguit implementar aquestes funcionalitats amb la qualitat esperada es considerarà que el producte mínim lliurable s'ha assolit amb èxit, i es convertirà en entregable, juntament amb els artefactes resultats d'aquest per al projecte.

2.2. Altres funcionalitats

En aquest apartat descriurem una sèrie de funcionalitats que s'intentarà incorporar al projecte però només en cas que el temps ens ho permeti. Aquestes funcionalitats són considerades molt interessants i en algun cas importants però no essencials ni prioritàries dintre de la plataforma, a diferència de les incloses en el producte mínim.

- Gestió d'usuaris: S'ampliarà la gestió d'usuaris perquè accepti l'autenticació a través de proveïdors tercers, per exemple, via OAuth.
- Gamificació: S'incorporarà un seguit de recursos i estratègies de gamificació en la plataforma, com ara badges, reptes, ranking, i altres.
- Time tracking: Eina integrada que permetrà als usuaris portar un control de temps de les tasques que realitzen, tenir informes de temps invertit, etc.
- Xat: S'ampliarà el xat perquè passi de ser un xat bàsic a un xat amigable amb el desenvolupament de programari (formateig de codi, etc.) i que incorpori la compartició de fitxers.
- Fòrum: Eina que permetrà la comunicació dels equips en forma de fòrums, integrat amb la gestió del cicle de vida (tasques, etc).
- Wiki: Incorporació de wikis als projectes i equips, integrat amb la gestió del cicle de vida (tasques, etc).
- Aplicació Android: Aplicació nativa Android amb el màxim de funcionalitats possibles per a que els usuaris puguin tenir una millor experiència en la mobilitat.

2.3. Funcionalitats futures

En aquest cas es descriurà una sèrie de funcionalitats que es considera que la plataforma hauria d'incorporar si el projecte continués en el temps ja fora del projecte merament universitari, a mitjà o llarg termini. S'indiquen aquí, tot i no formar part de l'abast del projecte, perquè es vol mostrar la direcció cap a on es vol anar, i també per si en cas que la plataforma no es discontinués finalment tenir les línies futures ja marcades.

A grans trets i sense entrar en detall, la plataforma en un futur hauria d'incorporar les següents eines i/o funcionalitats:

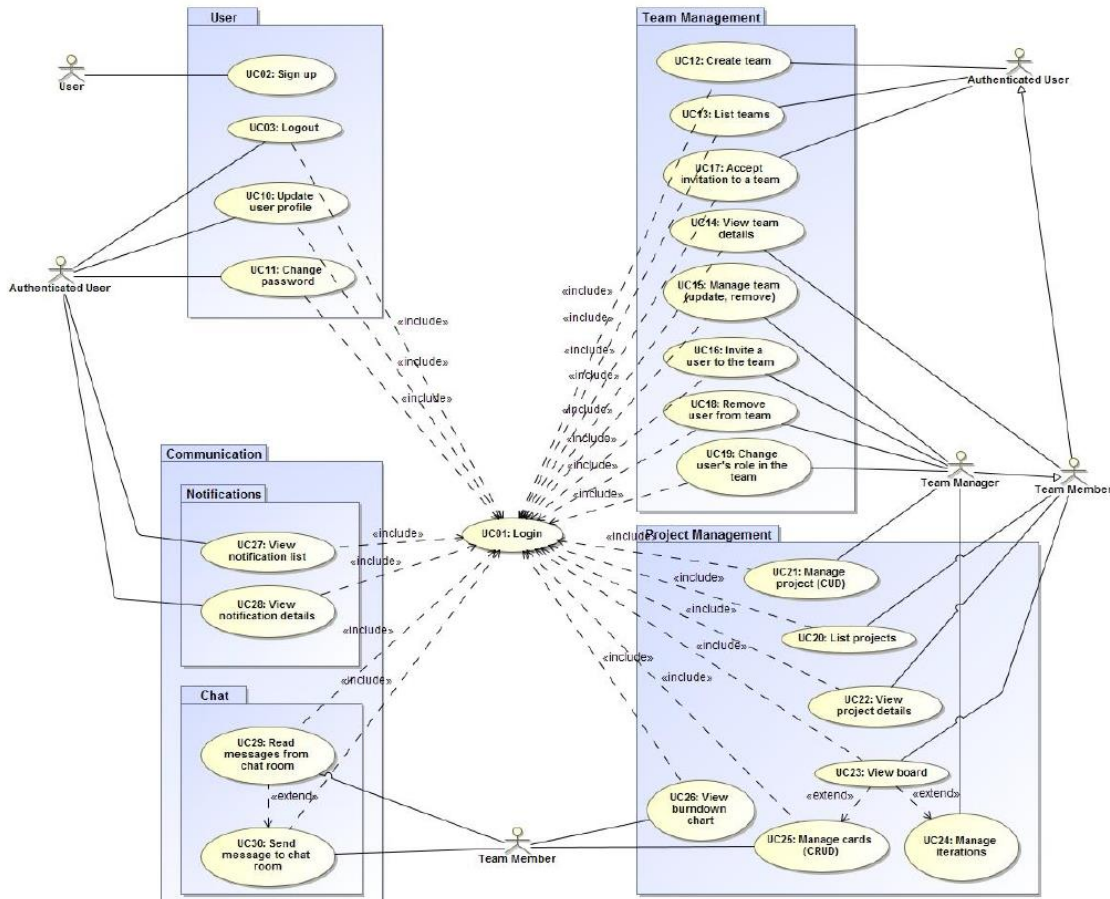
- SVC: incorporació de l'eina de control de versions, amb la gestió de repositoris amb Git.
- Integració continua: Plans, building, testing, etc.
- Continuous Deployment: Relacionat amb el punt anterior la possibilitat d'automatitzar les publicacions als diferents entorns segons les regles i configuracions especificades, etc.
- Integració amb eines de tercers.

3. Documentació de requisits

3.1. Stakeholders

Client: En aquest cas el client som nosaltres mateixos, els desenvolupadors del sistema.

3.2. Diagrama de casos d'ús



3.3. Actors

User: És tracta d'un usuari no autenticat en el sistema, és a dir, un usuari anònim.

Authenticated User: És tracta d'un usuari que s'ha autenticat en el sistema, i per tant ha d'estar registrat en el sistema i es coneix la seva identitat.

Team Member: És tracta d'usuaris autenticats que pertanyen a un equip.

Team Manager: És tracta de membres d'equip que tenen el rol de propietari de l'equip.

3.4. Fitxes de casos d'ús

UC01: Login

Descripció: Com a usuari no autenticat he de poder identificar-me en el sistema mitjançant les seves credencials per poder accedir a les funcionalitats que aquest ofereix per a usuaris autenticats.

Actor principal: User

Àmbit: Sistema de seguretat de Devilish

Nivell d'objectiu: tasca

Stakeholders:

Client: El client vol que l'aplicació estigui securitzada per autenticació bàsica (usuari i contrasenya).

Precondició: Cap

Garanties mínimes: S'ha donat feedback a l'usuari sobre el resultat de la identificació.

Garanties en cas d'èxit: L'usuari ara es troba identificat correctament en el sistema.

Escenari principal d'èxit:

- 1) L'usuari indica les seves credencials formades per una adreça d'email, que serà el seu nom d'usuari, i una contrasenya.
- 2) El sistema mostra la interfície personal de Devilish per a l'usuari ja autenticat.

Extensions:

- 1a) L'usuari no introdueix cap o algun dels camps de les credencials.
 - 1a1) El sistema avisa a l'usuari que ha d'introduir les credencials i torna a 1.
- 1b) L'usuari indica unes credencials incorrectes.
 - 1b1) El sistema informa a l'usuari que no s'ha pogut autenticar i torna a 1.

UC02: Sign up

Descripció: Com a usuari no autenticat he de poder registrar-me en el sistema per crear-me unes credencials que em permetin identificar-me en aquest.

Actor principal: User

Àmbit: Mòdul d'usuari de Devilish

Nivell d'objectiu: usuari

Stakeholders:

Client: El client vol que els usuaris es puguin registrar en el sistema.

Precondició: Cap

Garanties mínimes: S'ha donat feedback a l'usuari sobre el resultat del registre.

Garanties en cas d'èxit: L'usuari ara es troba registrat en el sistema amb les dades que ha indicat en el procés.

Escenari principal d'èxit:

- 1) L'usuari indica unes credencials escollides per ell i formades per una adreça d'email i una contrasenya (que serviran per autenticar-se).
- 2) El sistema dona feedback a l'usuari i dona l'opció d'autenticar-se (UC01: Login).

Extensions:

- 1a) L'usuari introdueix dades no vàlides o un email (nom d'usuari) ja existent.

1a1) El sistema avisa a l'usuari dels errors i torna a 1.

UC03: Logout

Descripció: Com a usuari autenticat he de poder tancar la sessió en el sistema per evitar deixar sessions obertes quan no hi soc que es puguin aprofitar per terceres persones.

Actor principal: Authenticated User

Àmbit: Mòdul d'usuari de Devilish

Nivell d'objectiu: usuari

Stakeholders:

Client: El client vol que l'usuari pugui sortir de la seva sessió.

Precondició: Cap

Garanties mínimes: Cap

Garanties en cas d'èxit: L'usuari ja no es troba identificat en el sistema i la seva sessió s'ha tancat.

Escenari principal d'èxit:

- 1) L'usuari indica que vol tancar la sessió en el sistema.
- 2) El sistema tanca la sessió de l'usuari.

UC04: Manage entity

Descripció: Cas d'ús reutilitzable per l'administració d'entitats.

Nivell d'objectiu: general

Escenari principal d'èxit:

- 1) L'usuari demana el llistat d'entitats utilitzant el cas d'ús UC05: List entities.
- 2) L'usuari torna al pas 1 o surt del cas d'ús.

Extensions:

- 3a) L'usuari indica que vol crear una nova entitat (UC06: Create entity).
 - 3a1) El sistema dona feedback a l'usuari i torna al pas 1.
- 3b) L'usuari indica que vol eliminar una entitat (UC09: Remove entity).
 - 3b1) El sistema dona feedback a l'usuari i orna al pas 1.
- 3c) L'usuari indica que vol veure el detall d'una entitat (UC07: View entity details).
 - 3c1) El sistema dona feedback a l'usuari i orna al pas 1.
- 3d) L'usuari indica que vol actualitzar les dades de l'entitat (UC08: Update entity).
 - 3d1) El sistema dona feedback a l'usuari i orna al pas 1.

UC05: List entities

Descripció: Cas d'ús reutilitzable per el llistat d'entitats.

Nivell d'objectiu: general

Escenari principal d'èxit:

- 1) L'usuari demana el llistat d'entitats podent aplicar o no uns criteris de cerca i un criteri d'ordenació o no.
- 2) El sistema mostra la llista d'entitats coincidents amb els criteris de cerca o totes en cas de no introduir-ne, i ordenades segons el criteri d'ordenació

indicat o el criteri d'ordenació per defecte si no se n'ha indicat cap. Aquesta llista pot ser una llista buida.

UC06: Create entity

Descripció: Cas d'ús reutilitzable per la creació d'entitats.

Nivell d'objectiu: general

Escenari principal d'èxit:

- 1) L'usuari indica que vol crear una nova entitat.
- 2) El sistema demana les dades de creació a l'usuari.
- 3) L'usuari introdueix les dades de creació.
- 4) El sistema verifica que les noves dades són correctes segons les validacions de creació.
- 5) El sistema crea la nova entitat.

Extensions:

- 5a) Les noves dades no compleixen les validacions de creació.
5a1) El sistema dona feedback a l'usuari i torna al punt 2.

UC07: View entity details

Descripció: Cas d'ús reutilitzable per la vista dels detalls d'entitats.

Nivell d'objectiu: general

Escenari principal d'èxit:

- 1) L'usuari indica que vol veure el detall d'una entitat.
- 2) El sistema mostra les dades de l'entitat.

UC08: Update entity

Descripció: Cas d'ús reutilitzable per l'actualització d'entitats.

Nivell d'objectiu: general

Precondició: L'entitat a actualitzar existeix en el sistema.

Escenari principal d'èxit:

- 1) L'usuari indica que vol actualitzar una entitat concreta.
- 2) El sistema demana les dades d'actualització a l'usuari.
- 3) L'usuari introdueix les dades d'actualització.
- 4) El sistema verifica que les noves dades són correctes segons les validacions d'actualització.
- 5) El sistema actualitza les dades de la nova entitat.

Extensions:

- 6a) Les noves dades no compleixen les validacions d'actualització.
4a1) El sistema dona feedback a l'usuari i torna al punt 2.

UC09: Remove entity

Descripció: Cas d'ús reutilitzable per la eliminació d'entitats.

Nivell d'objectiu: general

Escenari principal d'èxit:

- 1) L'usuari indica que vol eliminar una entitat.

- 2) El sistema comprova que l'entitat es pot eliminar, i en cas que no sigui possible surt del cas d'ús.
- 3) El sistema demana confirmació a l'usuari.
- 4) L'usuari confirma l'eliminació de l'entitat.
- 5) El sistema marca l'entitat com a eliminada.

Extensions:

- 4a) L'usuari cancel·la l'eliminació de l'entitat.
 - 4a1) Acaba el cas d'ús.

UC10: Update user profile

Descripció: Com a usuari autenticat he de poder actualitzar el meu perfil per personalitzar les meves dades.

Actor principal: Authenticated User

Àmbit: Mòdul d'usuari de Devilish

Nivell d'objectiu: usuari

Stakeholders:

Client: El client vol que l'usuari tingui un perfil on indicar diferents dades personals.

Precondició: Cap

Garanties mínimes: S'ha donat feedback a l'usuari sobre el resultat del procés.

Garanties en cas d'èxit: L'usuari ha actualitzat el seu perfil amb èxit.

Escenari principal d'èxit:

- 1) L'usuari indica que vol actualitzar el seu perfil
- 2) El sistema demana les dades de perfil actualitzables: nom, cognoms, email i idioma. Tots obligatoris.
- 3) L'usuari indica les dades de perfil.
- 4) El sistema verifica que les dades siguin vàlides. L'email és l'identificador d'usuari, pel que ha de tenir el format correcte i no pot coincidir amb el d'un altre usuari existent.
- 5) El sistema guarda el perfil amb les noves dades i dona feedback a l'usuari.

Extensions:

- 5a) Les dades no són vàlides.
 - 5a1) El sistema dona feedback a l'usuari i torna al punt 2.

UC11: Change password

Descripció: Com a usuari autenticat he de poder actualitzar la meua contrasenya per poder augmentar la seguretat del meu compte.

Actor principal: Authenticated User

Àmbit: Mòdul d'usuari de Devilish

Nivell d'objectiu: usuari

Stakeholders:

Client: El client vol que l'usuari tingui la possibilitat de canviar la seva contrasenya sempre que vulgui.

Precondició: Cap

Garanties mínimes: S'ha donat feedback a l'usuari sobre el resultat del procés.

Garanties en cas d'èxit: La contrasenya de l'usuari és la nova indicada per aquest en el procés.

Escenari principal d'èxit:

- 1) L'usuari indica que vol canviar la contrasenya.
- 2) El sistema li demana la contrasenya antiga més la nova.
- 3) El sistema verifica la contrasenya antiga.
- 4) El sistema verifica la validesa de la nova contrasenya.
- 5) El sistema canvia la contrasenya de l'usuari i li dona feedback.

Extensions:

- 4a) La contrasenya no és correcta.
 - 4a1) El sistema dona feedback a l'usuari i surt del cas d'ús.
- 5a) La nova contrasenya no és vàlida.
 - 5a1) El sistema dona feedback a l'usuari i torna al punt 2.

UC12: Create team

Descripció: Com a usuari autenticat he de poder crear equips per organitzar els membres participant dels meus projectes.

Actor principal: Authenticated User

Àmbit: Mòdul d'equips de Devilish

Nivell d'objectiu: usuari

Stakeholders:

Client: El client vol que l'usuari pugui crear equips d'usuaris.

Precondició: Cap

Garanties mínimes: S'ha donat feedback a l'usuari sobre el resultat del procés.

Garanties en cas d'èxit: S'ha creat un equip amb les dades introduïdes per l'usuari i amb aquest com a membre d'aquest amb el rol de Owner (propietari).

Escenari principal d'èxit:

- 1) L'usuari crea un equip utilitzant el cas d'ús UC06: Create entity, on:
 - a. L'entitat és l'equip.
 - b. Les dades de creació són el nom de l'equip.
 - c. Les validacions de creació són les següents:
 - i. El nom de l'equip no pot ser blanc i ha de ser únic en el sistema.
- 2) El sistema assigna l'usuari al nou equip i amb el rol de propietari (Owner).
- 3) El sistema dona feedback a l'usuari.

UC13: List teams

Descripció: Com a usuari autenticat he de poder veure un llistat amb els equips del que soc membre per tenir constància de tal fet i poder accedir a les seves funcionalitats pròpies.

Actor principal: Authenticated User

Àmbit: Mòdul d'equips de Devilish

Nivell d'objectiu: usuari

Stakeholders:

Client: El client vol que l'usuari pugui veure la llista d'equips dels que forma part.

Precondició: Cap

Garanties mínimes: S'ha donat feedback a l'usuari sobre el resultat del procés.

Garanties en cas d'èxit: Es mostra la llista d'equips dels quals l'usuari en forma part.

Escenari principal d'èxit:

- 1) L'usuari llista els equips del quals forma part utilitzant el cas d'ús UC05: List entities, on:
 - a. L'entitat és el equip.
 - b. Els criteris de cerca poden ser el nom de l'equip i el rol que té en aquests. Per defecte no hi ha criteri.
 - c. Els criteris d'ordenació poden ser el nom de l'equip i el rol que té en aquests. El criteri per defecte és el nom de l'equip.

UC14: View team details

Descripció: Com a membre de l'equip vull poder veure els detalls de l'equip.

Actor principal: Team Member

Àmbit: Mòdul d'equips de Devilish

Nivell d'objectiu: usuari

Stakeholders:

Client: El client vol que els membres d'un equip puguin veure les dades d'aquest.

Precondició: Cap

Garanties mínimes: S'ha donat feedback a l'usuari sobre el resultat del procés.

Garanties en cas d'èxit: Es mostren les dades de l'equip.

Escenari principal d'èxit:

- 1) L'usuari veu els detalls de l'equip utilitzant el cas d'ús UC07: View entity details, on:
 - a. L'entitat és l'equip.
 - b. Les dades de l'entitat mostrades són el nom de l'equip, el llistat d'usuaris que en formen part (email, rol) i el llistat de projectes (nom).

UC15: Manage team (update, remove)

Descripció: Com a administrador de l'equip he de poder actualitzar les seves dades o eliminar-los si les propietats d'aquests canvien o ja no té sentit que es trobi en el sistema.

Actor principal: Team Manager

Àmbit: Mòdul d'equips de Devilish

Nivell d'objectiu: usuari

Stakeholders:

Client: El client vol que els administradors dels equips puguin canviar les seves dades o eliminar-los.

Precondició: L'equip que es vol administrar existeix en el sistema.

Garanties mínimes: S'ha donat feedback a l'usuari sobre el resultat del procés.

Garanties en cas d'èxit: L'equip s'ha administrat correctament, és a dir, s'ha eliminat correctament del sistema o s'han actualitzat les dades indicades per l'usuari.

Escenari principal d'èxit:

- 1) L'administrador de l'equip actualitza les dades de l'equip utilitzant el cas d'ús UC08: Update entity, on:

- a. L'entitat és l'equip.
- b. Les dades d'actualització són el nom de l'equip.
- c. Les validacions d'actualització són les següents:
 - i. El nom de l'equip no pot ser blanc i ha de ser únic en el sistema.

Extensions:

- 1a) L'administrador de l'equip elimina l'equip utilitzant el cas d'ús UC09:
Remove entity, on:
 - a. L'entitat és l'equip.
 - b. L'entitat es pot eliminar sempre.

UC16: Invite a user to the team

Descripció: Com a administrador de l'equip he de poder invitar a altres usuaris a l'equip perquè aquests passin a formar part de l'equip si l'accepten.

Actor principal: Team Manager

Àmbit: Mòdul d'equips de Devilish

Nivell d'objectiu: usuari

Stakeholders:

Client: El client vol que l'administrador de l'equip pugui afegir usuaris a aquest mitjançant invitacions.

Precondició: Cap

Garanties mínimes: S'ha donat feedback a l'usuari sobre el resultat del procés.

Garanties en cas d'èxit: S'ha enviat una invitació a l'usuari que l'administrador vol afegir a l'equip.

Escenari principal d'èxit:

- 1) L'administrador de l'equip indica que vol afegir un usuari a l'equip i subministra l'email que l'identifica.
- 2) El sistema comprova que l'usuari existeixi en el sistema.
- 3) El sistema envia a l'usuari una notificació d'invitació a l'equip.

Extensions:

- 3a) El email no identifica a cap usuari del sistema.
3a1) El sistema dona feedback a l'usuari i torna al pas 1.

UC17: Accept invitation to a team

Descripció: Com a usuari autenticat que ha rebut una invitació a un equip he de poder acceptar la invitació per poder passar a formar part d'aquest equip.

Actor principal: Authenticated User

Àmbit: Mòdul d'equips de Devilish

Nivell d'objectiu: usuari

Stakeholders:

Client: El client vol que els usuaris que rebin una invitació a un equip puguin afegir-se a l'equip acceptant aquesta invitació.

Precondició: L'usuari autenticat té una notificació d'invitació a un equip.

Garanties mínimes: S'ha donat feedback a l'usuari sobre el resultat del procés.

Garanties en cas d'èxit: L'usuari autenticat ara forma part dels usuaris de l'equip.

Escenari principal d'èxit:

- 1) L'usuari obre la notificació d'invitació a l'equip i l'accepta.
- 2) El sistema comprova que la invitació sigui vàlida.
- 3) El sistema afegeix l'usuari a l'equip.

Extensions:

3a) La invitació no és vàlida, per exemple, per caducitat o perquè l'equip ja no existeix.

3a1) El sistema dona feedback a l'usuari i surt del cas d'ús.

UC18: Remove user from team

Descripció: Com a administrador de l'equip he de poder eliminar usuaris de l'equip perquè no puguin accedir a aquest.

Actor principal: Team Manager

Àmbit: Mòdul d'equips de Devilish

Nivell d'objectiu: usuari

Stakeholders:

Client: El client vol que els administradors de l'equip puguin eliminar usuaris d'aquest.

Precondició: Cap.

Garanties mínimes: S'ha donat feedback a l'usuari sobre el resultat del procés.

Garanties en cas d'èxit: L'usuari ja no forma part dels usuaris de l'equip.

Escenari principal d'èxit:

- 1) L'administrador de l'equip llista els usuaris de l'equip utilitzant el cas d'ús UC14: View team details, e indica l'usuari que vol eliminar.
- 2) El sistema elimina l'usuari del grup mitjançant el cas d'ús UC09: Remove entity on:
 - a. L'entitat és la relació entre el usuari i l'equip.
 - b. L'entitat és pot eliminar sempre i quan el usuari no sigui el propietari.

UC19: Change user's role in the team

Descripció: Com a administrador de l'equip he de poder canviar el rol dels membres d'un equip per poder reorganitzar els rols en aquest.

Actor principal: Team Manager

Àmbit: Mòdul d'equips de Devilish

Nivell d'objectiu: usuari

Stakeholders:

Client: El client vol que els membres de l'equip puguin canviar de rol.

Precondició: Cap

Garanties mínimes: S'ha donat feedback a l'usuari sobre el resultat del procés.

Garanties en cas d'èxit: El membre de l'equip passa a tenir el rol assignat per l'administrador.

Escenari principal d'èxit:

- 1) L'administrador de l'equip llista els usuaris de l'equip utilitzant el cas d'ús UC14: View team details, e indica el nou rol que tindrà aquest en l'equip.
- 2) El sistema comprova que el rol es pugui assignar.
- 3) El sistema assigna el rol escollit al membre de l'equip.

Extensions:

- 3a) El rol no es pot assignar a aquest membre de l'equip.
3a1) El sistema dona feedback a l'usuari i surt del cas d'ús.

UC20: List projects

Descripció: Com a membre de l'equip he de poder veure el llistat de projectes d'aquest.

Actor principal: Team Member

Àmbit: Mòdul de projectes de Devilish

Nivell d'objectiu: usuari

Stakeholders:

Client: El client vol que els membres de l'equip puguin veure la llista de projectes que té l'equip.

Precondició: Cap

Garanties mínimes: S'ha donat feedback a l'usuari sobre el resultat del procés.

Garanties en cas d'èxit: Es mostra la llista de projectes de l'equip.

Escenari principal d'èxit:

- 1) L'usuari llista els projectes de l'equip utilitzant el cas d'ús UC05: List entities, on:
 - a. L'entitat és el projecte.
 - b. Els criteris de cerca són el identificador de projecte i adicionalment el nom del projecte.
 - c. Els criteris d'ordenació poden ser el nom del projecte.

UC21: Manage projects (create, update, remove)

Descripció: Com a administrador de l'equip he de poder administrar els projectes de l'equip.

Actor principal: Team Manager

Àmbit: Mòdul de projectes de Devilish

Nivell d'objectiu: usuari

Stakeholders:

Client: El client vol que els administradors dels equips puguin crear, actualitzar i eliminar projectes.

Precondició: Cap

Garanties mínimes: S'ha donat feedback a l'usuari sobre el resultat del procés.

Garanties en cas d'èxit: El projecte s'ha administrat correctament.

Escenari principal d'èxit:

- 1) L'administrador de l'equip actualitza les dades del projecte utilitzant el cas d'ús UC08: Update entity, on:
 - a. L'entitat és el projecte.
 - b. Les dades d'actualització són el nom del projecte.
 - c. Les validacions d'actualització són les següents:
 - i. El nom del projecte no pot ser blanc i ha de ser únic en el sistema per a el equip concret.

Extensions:

- 1a) L'administrador de l'equip crea el projecte utilitzant el cas d'ús UC06: Create entity, on:
 - a. L'entitat és el projecte.

- b. Les dades de creació són el nom del projecte i el tipus de cicle de vida (Scrum, Kanban).
 - c. Les validacions de creació són les següents:
 - i. El nom del projecte no pot ser blanc i ha de ser únic en el sistema per a el equip concret.
- 1b) L'administrador de l'equip elimina el projecte utilitzant el cas d'ús UC09: Remove entity, on:
- a. L'entitat és el projecte.
 - b. L'entitat es pot eliminar sempre.

UC22: View project details

Descripció: Com a membre de l'equip vull poder veure els detalls del projecte.

Actor principal: Team Member

Àmbit: Mòdul de projectes de Devilish

Nivell d'objectiu: usuari

Stakeholders:

Client: El client vol que els membres d'un equip puguin veure les dades d'un projecte de l'equip.

Precondició: Cap

Garanties mínimes: S'ha donat feedback a l'usuari sobre el resultat del procés.

Garanties en cas d'èxit: Es mostren les dades del projecte.

Escenari principal d'èxit:

- 1) L'usuari veu els detalls del projecte utilitzant el cas d'ús UC07: View entity details, on:
 - a. L'entitat és projecte.
 - b. Les dades de l'entitat mostrades són el nom del projecte i el taulell del projecte (UC23: View board).

UC23: View board

Descripció: Com a membre de l'equip vull poder veure el taulell d'un projecte.

Actor principal: Team Member

Àmbit: Mòdul de projectes de Devilish

Nivell d'objectiu: usuari

Stakeholders:

Client: El client vol que els membres d'un equip puguin veure el taulell del projecte.

Precondició: Cap

Garanties mínimes: S'ha donat feedback a l'usuari sobre el resultat del procés.

Garanties en cas d'èxit: Es mostra el taulell del projecte.

Escenari principal d'èxit:

- 1) L'usuari veu el taulell del projecte utilitzant el cas d'ús UC22: View project details. Les dades que mostrarà el taulell són les columnes d'aquest, inclòs el backlog si és el cas, i les targetes existents en aquest.

UC24: Manage iterations

Descripció: Com a administrador de l'equip he de poder administrar iteracions per als projectes per poder organitzar el treball.

Actor principal: Team Manager

Àmbit: Mòdul de projectes de Devilish

Nivell d'objectiu: usuari

Stakeholders:

Client: El client vol que els administradors puguin crear iteracions i administrar-les.

Precondició: Cap

Garanties mínimes: S'ha donat feedback a l'usuari sobre el resultat del procés.

Garanties en cas d'èxit: Les iteracions s'han administrat correctament.

Escenari principal d'èxit:

- 1) L'administrador de l'equip actualitza les dades de la iteració utilitzant el cas d'ús UC08: Update entity, on:
 - a. L'entitat és la iteració.
 - b. Les dades d'actualització són les dates d'inici i final.
 - c. Les validacions d'actualització són les següents:
 - i. Les dates han de ser vàlides i la data d'inici no pot ser igual o superior a la final.

Extensions:

- 1a) L'administrador de l'equip crea la iteració utilitzant el cas d'ús UC06: Create entity, on:
 - a. L'entitat és la iteració.
 - b. Les dades de creació són les dates d'inici i final.
 - c. Les validacions de creació són les següents:
 - i. Les dates han de ser vàlides i la data d'inici no pot ser igual o superior a la final.
- 1b) L'administrador de l'equip elimina una iteració utilitzant el cas d'ús UC09: Remove entity, on:
 - a. L'entitat és la iteració.
 - b. L'entitat es pot eliminar sempre.
- 1c) L'administrador de l'equip mira els detalls d'una iteració utilitzant el cas d'ús UC07: View entity details, on:
 - a. L'entitat és la iteració.
 - b. Les dades de l'entitat són les dates d'inici i final, el seu identificador i el seu backlog.
- 1d) L'administrador de l'equip llista les iteracions del projecte utilitzant el cas d'ús UC05: List entities, on:
 - a. L'entitat és la iteració.
 - b. Els critèris de cerca són el identificador de projecte.
 - c. Els critèris d'ordenació poden ser la data d'inici de la iteració o la data final.

UC25: Manage cards

Descripció: Com a membre de l'equip he de poder administrar les targetes dels taulells als que tinc accés per poder evolucionar el meu progrés de treball.

Actor principal: Team Manager

Àmbit: Mòdul de projectes de Devilish

Nivell d'objectiu: usuari

Stakeholders:

Client: El client vol que els membres de l'equip puguin administrar les targetes dels taulells als que tenen accés.

Precondició: Cap

Garanties mínimes: S'ha donat feedback a l'usuari sobre el resultat del procés.

Garanties en cas d'èxit: Les targetes s'han administrat correctament.

Escenari principal d'èxit:

- 1) El membre de l'equip actualitza les dades de la targeta utilitzant el cas d'ús UC08: Update entity, on:
 - a. L'entitat és la targeta.
 - b. Les dades d'actualització són el títol, la descripció, la prioritat, els punts d'estimació i la columna en el taulell.
 - c. Les validacions d'actualització són les següents:
 - i. El títol i la columna no pot ser blanc. El backlog és com una columna.

Extensions:

- 1a) El membre de l'equip crea la targeta utilitzant el cas d'ús UC06: Create entity, on:
 - a. L'entitat és la targeta.
 - b. Les dades de creació són el títol (obligatori), la descripció, la prioritat, els punts d'estimació i la columna en el taulell (opcionals). La columna per defecte és el backlog en cas que no s'especificqui cap.
 - c. Les validacions de creació són les següents:
 - i. El títol no pot ser blanc.
- 1b) El membre de l'equip elimina una targeta utilitzant el cas d'ús UC09: Remove entity, on:
 - a. L'entitat és la targeta.
 - b. L'entitat es pot eliminar sempre.
- 1d) El membre de l'equip mira els detalls d'una targeta utilitzant el cas d'ús UC07: View entity details, on:
 - a. L'entitat és la targeta.
 - b. Les dades de l'entitat són l'identificador, el títol, la descripció, la prioritat, els punts d'estimació i la columna en el taulell.

UC26: View Burndown Chart

Descripció: Com a membre de l'equip he de poder veure el Burndown Chart d'un projecte.

Actor principal: Team Member

Àmbit: Mòdul de projectes de Devilish

Nivell d'objectiu: usuari

Stakeholders:

Client: El client vol que els membres de l'equip puguin veure el Burndown Chart dels projectes.

Precondició: Cap

Garanties mínimes: S'ha donat feedback a l'usuari sobre el resultat del procés.

Garanties en cas d'èxit: El sistema mostra el Burndown Chart del projecte seleccionat.

Escenari principal d'èxit:

- 1) El membre de l'equip accedeix a un projecte (UC22: View project details) o al llistat de projectes (UC20: List projects).
- 2) El membre de l'equip indica que vol veure el Burndown Chart d'un projecte.
- 3) El sistema li mostra el Burndown Chart.

UC27: View notifications list

Descripció: Com a usuari autenticat vull poder rebre notificacions.

Actor principal: Authenticated User

Àmbit: Mòdul de comunicació de Devilish

Nivell d'objectiu: usuari

Stakeholders:

Client: El client vol els usuaris autenticats puguin rebre notificacions i les vegin llistades en els seu dashboard.

Precondició: Cap

Garanties mínimes: S'ha donat feedback a l'usuari sobre el resultat del procés.

Garanties en cas d'èxit: Es mostra la llista de notificacions de l'usuari.

Escenari principal d'èxit:

- 1) L'usuari llista les seves notificacions utilitzant el cas d'ús UC05: List entities, on:
 - a. L'entitat és la notificació.
 - b. Els criteris de cerca són el identificador de l'usuari.
 - c. Els criteris d'ordenació poden ser la data de creació.

UC28: View notification details

Descripció: Com a usuari autenticat vull poder veure els detalls d'una notificació.

Actor principal: Authenticated User

Àmbit: Mòdul de comunicació de Devilish

Nivell d'objectiu: usuari

Stakeholders:

Client: El client vol els usuaris autenticats puguin veure els detalls d'una notificació

Precondició: Cap

Garanties mínimes: S'ha donat feedback a l'usuari sobre el resultat del procés.

Garanties en cas d'èxit: El sistema mostra els detalls de la notificació a l'usuari.

Escenari principal d'èxit:

- 1) L'usuari mostra els detalls d'una notificació utilitzant el cas d'ús UC07: View entity details, on:
 - a. L'entitat és la notificació.
 - b. Les dades de l'entitat són data de creació, títol i text.

UC29: Read messages from chat room

Descripció: Com a membre de l'equip vull poder llegir els missatges publicats al xat de l'equip.

Actor principal: Team Member

Àmbit: Mòdul de comunicació de Devilish

Nivell d'objectiu: usuari

Stakeholders:

Client: El client vol els membres de l'equip es puguin comunicar via xat.

Precondició: El xat és d'un equip al que el membre de l'equip pertany.

Garanties mínimes: S'ha donat feedback a l'usuari sobre el resultat del procés.

Garanties en cas d'èxit: Es mostra la llista de missatges del xat.

Escenari principal d'èxit:

- 1) El membre de l'equip accedeix als missatges del xat utilitzant el cas d'ús UC05: List entities, on:
 - a. L'entitat és el missatge de xat.
 - b. Els criteris de cerca són el identificador de la room.
 - c. Els criteris d'ordenació és la data de creació del missatge.

UC30: Send message to chat room

Descripció: Com a membre de l'equip vull poder enviar missatges al xat de l'equip.

Actor principal: Team Member

Àmbit: Mòdul de comunicació de Devilish

Nivell d'objectiu: usuari

Stakeholders:

Client: El client vol els membres de l'equip es puguin comunicar via xat.

Precondició: El xat és d'un equip al que el membre de l'equip pertany.

Garanties mínimes: S'ha donat feedback a l'usuari sobre el resultat del procés.

Garanties en cas d'èxit: El missatge enviat per el membre de l'equip forma part de la llista de missatges del xat.

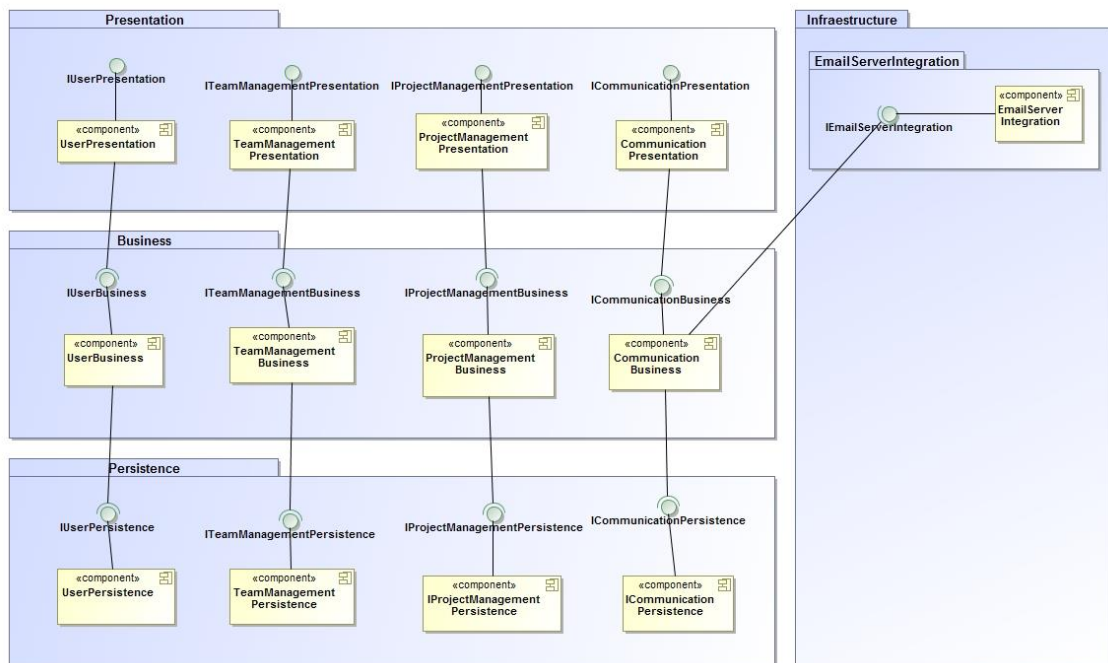
Escenari principal d'èxit:

- 1) El membre de l'equip accedeix als missatges del xat utilitzant el cas d'ús UC29: Read messages from chat room.
- 2) El membre de l'equip escriu i envia un missatge a la room.
- 3) El sistema l'afegeix als missatges de la room i el mostra als usuaris connectats.

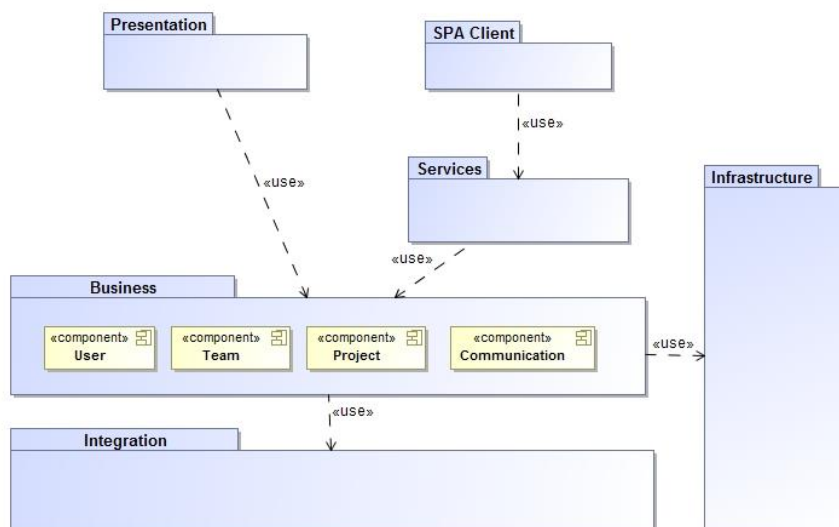
5. Arquitectura

En un principi s'havia proposat seguir l'estructura de capes pròpies de DDD, però degut a la falta de temps i d'experiència en JavaEE, s'ha cregut convenient començar amb una arquitectura clàssica de 3 capes més una capa transversal d'infraestructura. En posteriors refinament s'intentarà arribar a la arquitectura proposada.

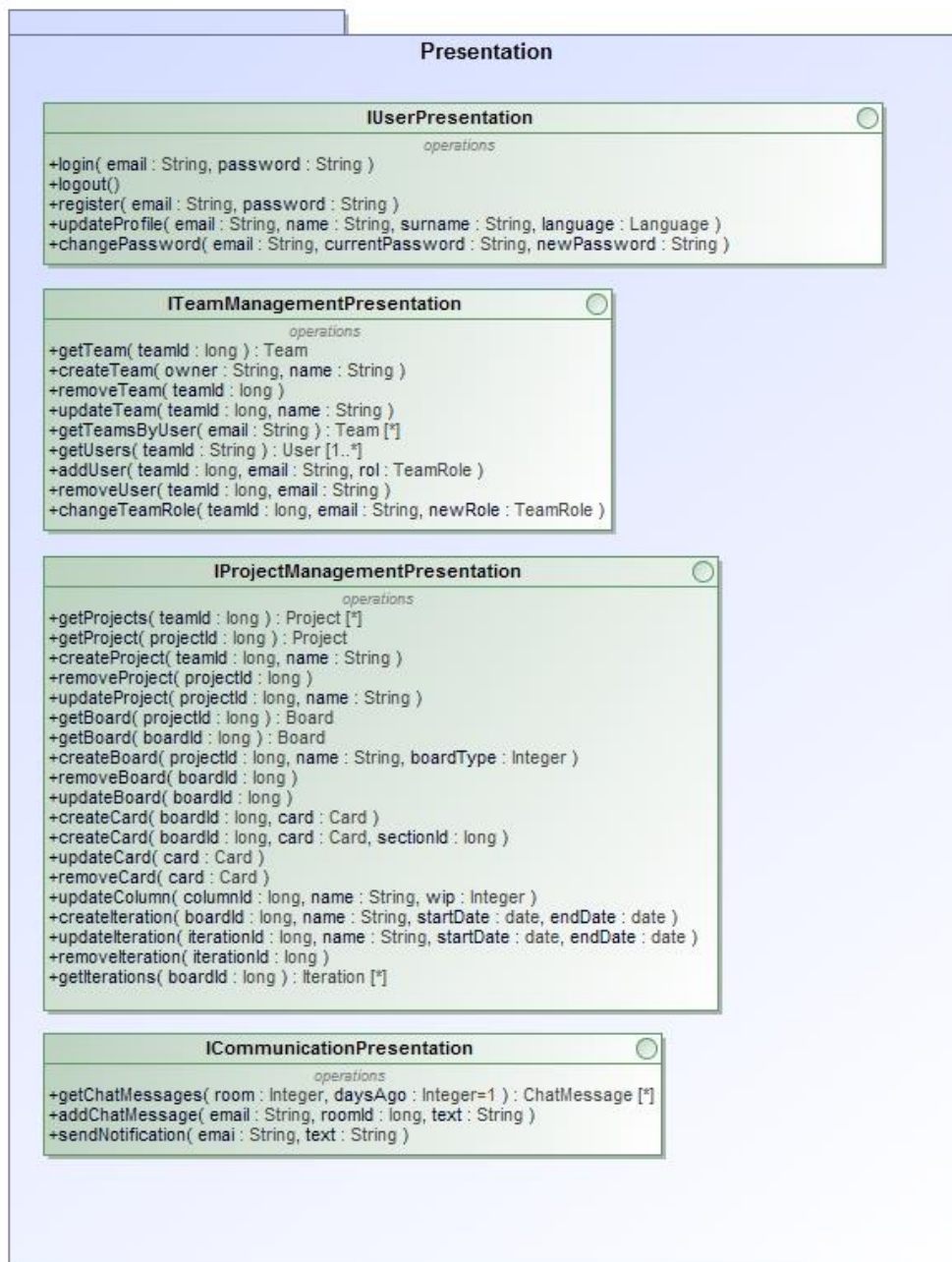
El diagrama següent mostra la arquitectura a alt nivell sense cap refinament.



En posteriors refinaments s'ha afegit una capa de serveis, quedant la arquitectura com segueix:



Les interfícies de Presentació tindran els següents contractes:



Per a les capes de Negoci i Persistència els contractes seran els mateixos ja que el servei ofert es difondrà de dalt a baix entre capes. En aquest cas es segueix un patró facade per al disseny del punt d'entrada de les diferents capes, això en aporta exposar només els serveis necessaris i amagar internament el que no sigui necessari exposar.

Aquests contractes han variat una mica un cop creada la implementació, i variaran per afegir més funcionalitat en les següents fases.

Per a la capa de presentació s'ha fet servir el patró MVC, ja que ens permet deslligar les vistes de la lògica d'una forma clara, ajudant a la mantenibilitat del codi.

Per a la capa d'infraestructura tenim la integració amb el servidor de emails, la qual presenta la següent interfície:



Com es veu en el diagrama s'han creat quatre mòduls: usuari, equip, projecte i comunicació. On cada un d'ells inclou la funcionalitat que el seu nom indica i que podem veure en el diagrama de casos d'ús que tenim en apartats anteriors.

Aquesta arquitectura en fases posteriors ha d'evolucionar a una arquitectura amb microserveis, amb comunicació per missatgeria, i amb un API Gateway com a punt d'entrada per a les diferents aplicacions client. Evidentment, això està fora de l'scope del TFG.

6. Tecnologies i eines

Les tecnologies i eines presentades en els següents apartats són tan les proposades inicialment per al projecte com les finalment utilitzades, algunes per motius de desconeixement d'alternatives potser millors i/o per estar familiaritzat amb elles, i altres per decisió meditada. En el transcurs del projecte, s'han fet diferents prospeccions i també s'han produït canvis que han variat les decisions, ja sigui per falta de temps o per creure que el canvi era favorable.

6.1. Llenguatges

Java 8 és el escollit com a llenguatge de programació per implementar el back-end de la nostra plataforma. Per al front-end es volia crear una SPA i utilitzar només HTML5, CSS3 i Javascript, però per falta de temps s'ha creat una capa de presentació JSF, deixant per al futur la creació de la SPA. En properes iteracions també es vol utilitzar SASS o LESS, encara per decidir quin dels dos llenguatges, per a millorar la creació dels CSS.

JSON és el format utilitzat per al transport de dades entre el futur front-end SPA i el back-end a través de la capa de serveis REST, ja implementada parcialment.

6.2. Plataforma de desenvolupament

6.2.1. Back-end

Tal com marca l'àrea escollida del projecte la plataforma de desenvolupament que farem servir és JavaEE. Aquesta és una plataforma

de components per a aplicacions distribuïdes empresarials que ens proporciona un conjunt de llibreries i serveis que ens ajudaran a aconseguir arquitectures heterogènies i multicapa, i requisits no funcionals com ara alta disponibilitat, seguretat, escalabilitat, mantenibilitat, etc.

Amb aquesta plataforma implementarem el back-end de l'aplicació i la seva API web encarregada d'exposar els serveis públics d'aquest. Per això, utilitzarem els seus components de negoci (EJB), els serveis REST per la API, i el ORM JPA-Hibernate per la persistència. També s'investigarà en el futur tots els seus components per a decidir si algun d'ells s'incorpora al nostre stack tecnològic, potser substituint els descrits en apartats posteriors.

A part de la capa de serveis REST, la capa de negoci exposa els serveis de l'aplicació a través d'interfícies remotes i locals RMI, on accedeix la capa de presentació.

Com a servidor d'aplicacions s'ha utilitzat JBoss (WildFly), degut a les seves característiques e integració amb Eclipse, i per estar familiaritzat pel treball en assignatures anteriors.

6.2.2. *Front-end*

Per al front-end web, com s'ha comentat s'havia decidit prescindir de JSF, en favor d'una SPA. Però les circumstàncies ens han fet replantejar la decisió, i s'ha creat una capa de presentació en JSF que accedeix a la capa de negoci via EJBs / RMI.

Per la futura SPA ens recolzarem en el framework ReactJS o en algun altre com ara AngularJS si el primer es queda curt, juntament amb altres com RequireJS, jQuery, Modernizr, etc.

El client web es tractarà d'un client pesat que implementarà el patró arquitectònic MVC o MVVM depenent del framework escollit finalment.

6.2.3. *Apps*

En un futur es crearà una aplicació Android nativa, amb Java i AXML com a llenguatges, i utilitzant el SDK d'Android, tot i que actualment estan havent canvis que poden variar les desicions.

6.3. *Bases de dades*

S'ha fet una prospecció sobre el paradigma de base de dades a utilitzar, tenint en compte els requisits de l'aplicació i la infraestructura de la que disposarem.

6.3.1. *Relacional*

S'ha decidit utilitzar una base de dades relacional, més concretament PostgreSQL, ja que era la base de dades proposada en l'entorn de treball, i ja existia experiència prèvia.

6.3.2. NoSQL

En un futur no es descarta utilitzar una base de dades NoSQL, probablement MongoDB, per a algun dels components de la aplicació, ja que el més probable és que algunes de les estructures de dades encaixi millor amb el paradigma de documents en lloc del relacional.

6.4. Eines, frameworks i llibreries

6.4.1. IDEs

Eclipse ha sigut el nostre IDE principal per al desenvolupament de la plataforma. S'ha escollit per estar ja familiaritzat amb ell, però sobretot per les seves funcionalitats i la seva modularitat, la qual aporta una gran capacitat d'integració amb altres eines i tecnologies que s'han fet servir en el projecte, com ara Git, JBoss, etc. Tot i aquesta decisió, inicialment es va provar IntelliJ, però degut a falta de familiaritat i de temps es va descartar.

També s'ha utilitzat Sublime Text per a la implementació del front-end SPA, i es va fer un anàlisi de WebStorm, tot i que al final es va descartar el treball fet.

Per a l'aplicació Android, s'utilitzarà Android Studio.

6.4.2. Control de versions

S'ha utilitzat Git com a SVC. Es tracta d'un sistema de control de versions descentralitzat molt popular avui dia que ens aporta velocitat, flexibilitat e integració amb multitud d'eines entre moltes altres avantatges. Per treballar en local ens hem baixat l'última versió d'aquest a través de la web de Git. A més a més, s'ha utilitzat la plataforma GitHub per treballar en remot i així poder compartir, tindrè mobilitat i tenir els repositoris "ben guardats". S'ha valorat la opció d'instal·lar GitLab en un servidor però s'ha descartat per evitar sobreesforços de manteniment.

6.4.3. Gestió del cicle de vida

Per a la gestió del cicle de vida del desenvolupament de la plataforma es volia utilitzar Trello. Aquesta és una eina online i gratuïta que ens permet crear taulells completament personalitzats, i que, tot i que no ens serà de molta utilitat en el projecte, disposa d'una aplicació per mòbil. Es va escollir aquest eina per la seva gratuïtat i sobretot per la seva simplicitat i flexibilitat. Finalment no s'ha utilitzat per falta de temps.

Es va valorar també fer servir Jira o Visual Studio Team Services per tenir una plataforma que integri tant la gestió del cicle de vida com altres tipus d'eines com ara el SVC, la CI, el testing, etc, però es va descartar en favor de la simplicitat i per les mateixes raons que són l'origen d'aquest projecte. A més, aquestes dues eines ha sigut àmpliament analitzades per a extreure conclusions i requisits per al nostre projecte.

6.4.4. Gestió documental

Es té la intenció de fer servir la wiki que la plataforma GitHub ofereix per a documentar el projecte en un futur. No es descarta afegir alguna altra eina com ara Sharepoint o Google Docs com eina principal que es complementi o substitueixi completament la wiki de GitHub si aquesta no ens satisfà totes les necessitats en aquest tema.

6.4.5. Modelatge

Magic Draw ha sigut l'eina escollida per al modelatge degut a ser de les eines més completes i professionals en aquest tema, i gracies a que disposem de bastanta experiència i una llicència temporal.

6.4.6. Automatització

Es té la intenció d'utilitzar en un futur Jenkins per a automatitzar processos i implementar la Integració Continua i possiblement el Continuous Deployment.

6.4.7. Testing

Com a eines de testing s'utilitzarà JUnit per a les proves unitàries en Java, i QUnit per Javascript, tot i aquest últim framework no està del tot decidit. Probablement es farà mocking amb Mockito o JMock, ha investigar. Potser també s'utilitza Jasmine en el front-end, i per a les proves d'acceptació s'utilitzarà Selenium. Les proves de carrega dels serveis es faran amb jMeter. Postman es farà servir per fer els tests sobre la PAI REST de la capa de serveis. S'investigarà la inclusió d'algun altre framework o eina. Actualment, a part de Postman no s'ha pogut incorpora cap de les eines planificades, però s'incorporaran en un futur.

6.4.8. Logging

En aquest apartat s'utilitzarà quasi sense cap dubte Log4j, per ja estar familiaritzat amb ell.

6.4.9. AOP

Es molt probable que s'afegeixi un framework per a la programació orientada a aspectes, per poder extreure el codi transversal (logging, instrumentació, etc.) del codi funcional. De moment només es coneix AspectJ, però s'investigarà per veure quina és la millor decisió.

6.4.10. Contenedor d'injecció de dependències

S'utilitzarà injecció de dependències ja que es creu bàsic per a assolir la qualitat del codi desitjada, i s'implementarà per mitjà d'un contenedor, encara per decidir, ja que només conec Spring, però al tractar-se d'un framework complet, i per tant més pesat s'ha d'investigar. En fases posteriors a l'entrega del TFG s'ha de rsoldre.

6.4.11. Mètriques

S'ha d'investigar la utilització d'eines com ara jHawk o altres.

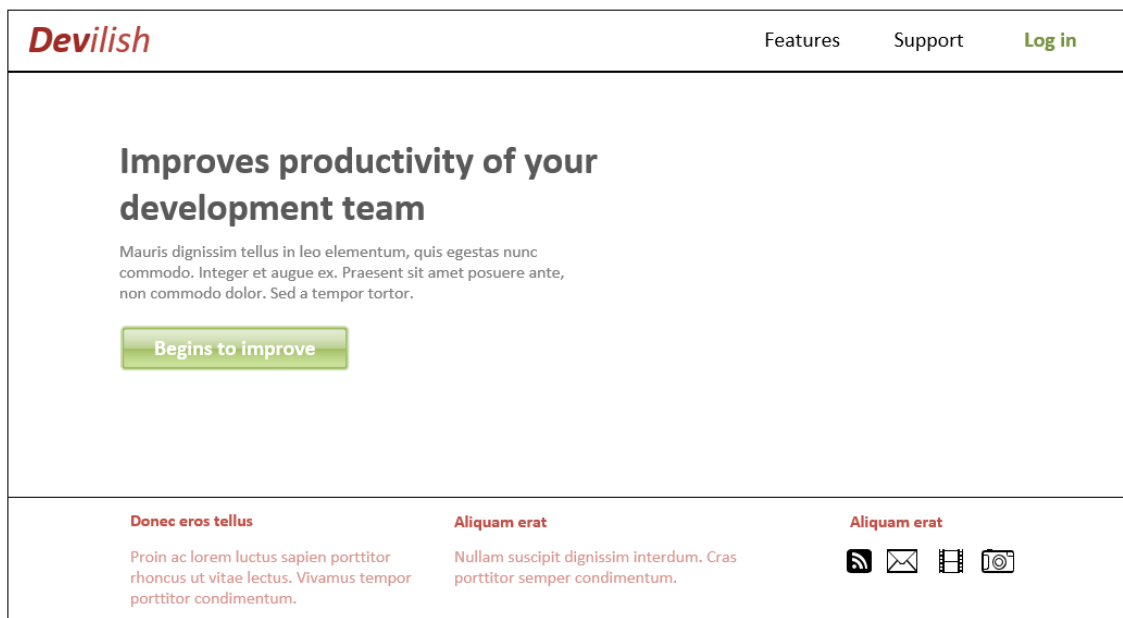
6.4.12. Prototipatge web

S'investigarà la utilització d'alguna eina per al prototipatge web com ara Axure.

7. Model de pantalles

Tot seguit es mostren els models de pantalla més significatius que es van desenvolupar com a prototips inicials per a la implementació del sistema.

Pàgina d'inici



Pàgina d'autenticació





Devilish [Features](#) [Support](#) [Log in](#)

Create your account

Mauris iaculis erat est, aliquam gravida magna rhoncus id. Vestibulum nec odio et justo molestie commodo.

Donec eros tellus
Proin ac lorem luctus sapien porttitor rhoncus ut vitae lectus. Vivamus tempor porttitor condimentum.










Aliquam erat
Nullam suscipit dignissim interdum. Cras porttitor semper condimentum.

Aliquam erat
   

Taulell del projecte





Devilish [Features](#) [Support](#) [Oscar Rubio](#)

My Team > My Project

| | To Do | Doing | Done |
|--|--|---|---|
|   |   |  |     |

Donec eros tellus
Proin ac lorem luctus sapien porttitor rhoncus ut vitae lectus. Vivamus tempor porttitor condimentum.

Aliquam erat
Nullam suscipit dignissim interdum. Cras porttitor semper condimentum.

Aliquam erat
   

8. Conclusions

Els objectius del projecte s'han assolit parcialment, sens dubte la implementació del sistema ha sigut el gran perjudicat per la poca disponibilitat de la que s'ha disposat. En resum:

- Sí s'ha posat en pràctica els coneixements adquirits durant la carrera.
- S'ha aprofundit en les tecnologies Java parcialment: s'ha fet bastant anàlisis però no s'ha pogut posar en pràctica gran part de l'analitzat.
- S'ha posat la primera pedra de Devilish, però dista molt la quantitat de funcionalitat implementada que es va planificar del que realment s'ha aconseguit.

Tot i això, s'ha fet un bon anàlisis de diferents frameworks i tecnologies Java que ens pot servir per a les següents fases.

Un altre aspecte que no s'ha assolit és el seguiment de la planificació. La metodologia utilitzada ha complert, però el temps del que s'ha disposat ha sigut molt inferior del planificat, fet que ha influït molt negativament en el projecte.

A partir d'ara, comença una segona fase on s'incorporarà tot el que en aquesta primera aproximació no ha arribat, i ha d'evolucionar a la idea real de la plataforma Devilish, apuntant cap a una arquitectura de microserveis amb APIs REST i clients SPA i Mobile.

9. Glossari

- **DDD:** Domain-Driven Design, sèrie de pràctiques de desenvolupament de programari que es centren en una orientació al domini.
- **Java EE:** Entorn tecnològic de Java per a sistemes empresarials, i distribuïts, que incorpora una sèrie de frameworks, llibreries, etc., destinades a tals desenvolupaments.
- **JSF:** Java Server Faces, tecnologia per a capa de presentació inclosa en Java EE.
- **REST:** Filosofia i pràctiques de disseny d'APIs web que utilitzen orientació a recurs i verbs HTTP per a les peticions.
- **SPA:** Single Page Application, és tracta d'un client web pesat que incorpora la majoria de la lògica de la aplicació, i només connecta el mínim necessari al servidor, normalment per fer peticions de dades a una API REST. Una altra característica és que es tracta d'una única pàgina web que es construeix en temps d'execució, normalment mitjançant Javascript.

10. Bibliografia

No s'ha utilitzat cap referència bibliogràfica en aquesta memòria.

11. Annexos

No existeixen annexos.