



Desenvolupament d'una aplicació web d'oferta i cerca de feina

Memòria de Projecte Final de Grau

Grau Multimèdia, itinerari d'Enginyeria Web

Autor: Anna Ferry Mestres

Consultor: Ignasi Lorente Puchades

Professor: Carlos Casado Martinez

Data d'entrega: 13/06/2016



Aquesta obra està subjecta a una llicència de [Reconeixement-NoComercial-SenseObraDerivada 4.0 Espanya de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/4.0/es/)

A la **Jara**, per la paciència que ha tingut i la que li falta per tenir

Abstract

Aquest projecte consisteix en el disseny i desenvolupament de **faenio**, una aplicació web per a posar en contacte persones que ofereixen habilitats amb altres que les poden necessitar.

N'hi ha prou amb crear-se un perfil i **faenio** s'encarrega de mostrar tots aquells professionals que ofereixen els seus serveis al nostre voltant. Es pot filtrar per categoria, per tal d'afinar la cerca i trobar a aquell professional que ens solucionarà el problema. Un cop el tenim, hi podem xatejar per acabar de pactar els detalls i després del servei podem emetre una valoració.

A l'aplicació hi ha dos tipus d'usuari:

- Proveïdors: Aquells professionals que ofereixen els seus serveis a través de la plataforma
- Clients: Aquells usuaris que necessiten els serveis d'un professional

A nivell tècnic, l'aplicació està desenvolupada emprant la metodologia MEAN (MongoDB – Express.js – Angular.js – Node.js), un conjunt de llibreries basades en JavaScript que cobreix tant el *frontend*, el *backend* i la base de dades.

faenio segueix el plantejament *Mobile First* i tot i que s'adapta tant a dispositius d'escriptori com dispositius mòbils, està plantejat per acabar convertint-se en una aplicació mòbil.

Paraules clau: *feina, proximitat, aplicació, desenvolupament, oportunitat, autònoms, MEAN*

Abstract (english version)

This project is based on the design and development of **faenio**, a Web application to put together people with skills with other people who may need them.

We just have to create a profile and **faenio** is in charge of showing all those professionals who offer their services around us. We can filter by category to refine our search and find a professional who will solve the problem. Once we have it, we can finally chat to reach an agreement and after the service is made we can make an assessment on the provider.

There are two types of users in the application:

- Providers: Those professionals who offer their services through the platform
- Customers: Those users who need the services from a professional

Technically, the application is developed using the MEAN methodology (MongoDB - Express.js - Angular.js - Node.js), based on a set of JavaScript libraries that cover the frontend, backend and database.

faenio follows the *Mobile First* approach and although it fits both desktop and mobile devices, it is actually designed to end up becoming a mobile application.

Keywords: *job, closeness, application, development, chance, freelance, MEAN*

1. Introducció.....	9
2. Definició del projecte.....	10
3. Objectius.....	11
3.1 Objectius principals.....	11
3.2 Objectius secundaris.....	11
4. Continguts.....	12
4.1 Estructurals.....	12
4.2 Funcionals.....	15
5. Metodologia.....	16
6. Arquitectura de l'aplicació.....	17
6.1 Client.....	17
6.2 Servidor.....	17
6.3 Bases de dades.....	17
7. Plataforma de desenvolupament.....	18
7.1 Programari.....	18
7.2 Maquinari.....	19
8. Planificació.....	20
8.1 Dates clau.....	20
8.2 Diagrama de Gantt.....	20
9. Procés de treball.....	23
9.1 PAC1.....	23
9.2 PAC2.....	24
9.3 PAC3.....	24
9.4 Lliurament final.....	25
10. APIs utilitzades.....	28
10.1 Google Maps.....	28
10.2 Facebook i Twitter.....	28
11. Base de dades.....	30
11.1 Model Entitat Relació.....	30
11.2 Creació de la base de dades.....	31
11.3 Generació del contingut de la base de dades.....	31
12. Casos d'us.....	33
12.1 Diagrama UML de casos d'us.....	33
12.2 Descripció dels casos d'us.....	33
13. Prototips.....	38

13.1 Lo-Fi	38
13.2 Hi-Fi.....	43
14. Perfils d'usuari	48
14.1 Usuaris proveïdors de serveis	48
14.2 Usuaris consumidors de serveis.....	48
15. Històries d'usuari	49
16. Usabilitat/UX.....	51
17. Seguretat.....	55
18. Instruccions d'instal·lació.....	56
19. Bugs.....	58
20. Projectió a futur.....	59
21. Estudi de viabilitat.....	60
21.1. Pressupost.....	60
21.2 Anàlisi de mercat	61
21.3 Micro-pla d'empresa.....	64
21.4 Conclusions de viabilitat	65
22. Conclusions	67
Annex 1. Lliurables del projecte.....	68
Annex 2. Codi font (extractes)	69
Aplicació client: encaminador del perfil	69
API.....	70
Enviament de missatges en temps real	71
Aplicació servidor: Cerca de publicacions.....	73
Annex 3. Llibreries externes.....	75
Annex 4. Captures de pantalla	80
Annex 5. Guia d'usuari	87
Executar l'aplicació	87
Iniciar sessió.....	87
Navegació per l'aplicació	88
Annex 6. Guia d'estil	89
Logotip.....	89
Paleta tipogràfica i mida de fonts	89
Paleta de colors	89
Retícula	90
Elements d'interacció	90
Annex 7. Glossari	92
Annex 8. Bibliografia	93

Figures i taules

A continuació es mostra el llistat d'imatges, taules, gràfics, diagrames, etc., numerades, amb títols i les pàgines on apareixen.

Índex de figures

Figura 1. Desenvolupament en cascada	16
Figura 2. Diagrama de Gantt	22
Figura 3. API de Facebook	28
Figura 4. Fragment de codi de passport.js on es passa la informació del perfil de Facebook al nou usuari	29
Figura 5. Diagrama d'entitat relació.....	30
Figura 6. Diagrama UML de casos d'us.....	33
Figura 7. Wireframe pàgina inici escriptori	38
Figura 8. Wireframe pàgina detall escriptori.....	39
Figura 9. Wireframe pàgina llistat escriptori.....	39
Figura 10. Wireframe pàgina perfil escriptori	40
Figura 11. Detall del menú d'usuari obert.....	40
Figura 12. Wireframe pàgina inici mòbil	41
Figura 13. Wireframe pàgina detall mòbil.....	41
Figura 14. Wireframe pàgina resultats mòbil.....	42
Figura 15. Wireframe pàgina perfil mòbil	42
Figura 16. Mockup pàgina inici escriptori	43
Figura 17. Mockup pàgina detall escriptori.....	44
Figura 18. Mockup pàgina llistat escriptori	44
Figura 19. Mockup pàgina perfil escriptori	45
Figura 20. Detall del menú d'usuari obert.....	45
Figura 21. Mockup pàgina inici mòbil.....	46
Figura 22. Mockup pàgina detall mòbil.....	46
Figura 23. Mockup pàgina llistat mòbil	47
Figura 24. Mockup pàgina perfil mòbil.....	47
Figura 25. Afegir Heroku com a Master i publicar el repositori	56
Figura 26. Confirmació de desplegament finalitzat.....	57
Figura 27. Error a l'executar l'aplicació al servidor.....	57
Figura 28. Disseny de les diferents pantalles de l'aplicació.....	80
Figura 29. Historial de commits a Bitbucket.....	80
Figura 30. Formulari d'inici de sessió	81
Figura 31. Formulari de registre.....	81
Figura 32. Formulari de registre amb correu electrònic.....	82
Figura 33. Valoració de serveis amb comentari i puntuació numèrica	82
Figura 34. Pàgina d'inici: filtre per categories	83
Figura 35. Nova publicació de servei.....	83

Figura 36. Editar publicació de servei	84
Figura 37. "Els meus missatges" (converses amb usuaris)	84
Figura 38. Nova conversa amb un altre usuari.....	85
Figura 39. Conversa amb en Jordi sobre un servei.....	85
Figura 40. Editar perfil d'usuari.....	86
Figura 41. Perfil d'usuari	86
Figura 42. Logotip sobre fons fosc	89
Figura 43. Logotip sobre fons clar	89
Figura 44. Reticula de 12 columnes	90

Índex de taules

Taula 1. Dates clau	20
Taula 2. Diagrama de Gantt	21
Taula 3. Bugs	58
Taula 4. Recursos humans.....	61
Taula 5. Recursos materials	61

1. Introducció

En els últims anys i arrel de la crisi que ens ha caigut sobre les espatlles s'ha vist un canvi en el model productiu del país amb un alt creixement de l'autoocupació, en alguns casos per convicció, en altres per pura supervivència, en resposta a la molta demanda i poca oferta de llocs de treball com a assalariat.

Com a treballadora autònoma que sóc, en aquest treball m'he volgut posar en la pell del "client", dissenyant i desenvolupant l'aplicació que m'agradaria tenir per a oferir els meus serveis i els de tots aquells qui, al igual que jo, es *guanyen les garrofes* com poden sense seguir el model tradicional del funcionariat.

Així doncs no és una aplicació en la que es llisten ofertes laborals i els cercadors de feina hi envien el seu currículum, sinó que és més aviat el contrari: els professionals llisten les seves candidatures i són els clients els que els aniran a cercar quan tinguin una necessitat, a l'estil d'una aplicació d'anuncis classificats.

Un professional crea un perfil a l'aplicació i dona d'alta la candidatura dels seus serveis, informant de la remuneració que espera rebre i si s'escau mostrant exemples (fotografies o enllaços) que mostrin el que ha fet prèviament. A partir d'aquí només li toca esperar a que algú tingui una necessitat que ell pugui cobrir i entrin en contacte.

Per altra banda, aquells usuaris que tinguin una necessitat (una fuita a la cuina, la pàgina web de la botiga, un ordinador que ha deixat de funcionar, classes de repàs,...) quan accedeixen a **faenio** veuen tots els professionals que tenen a prop, de manera que la transacció pot ser molt més immediata.

La meva intenció no és la d'incentivar el treball en negre, encara que això és difícil de controlar i quedarà en mans de qui ho utilitzi: Les eines per si mateixes no són bones ni dolentes, tot i que puguin ser utilitzades per al bé i per al mal.

Al final, si el client troba la solució al seu problema i el professional rep una remuneració per la seva feina, això és el que més importa.

2. Definició del projecte

El present TFG s'emmarca dins l'àrea d'Enginyeria Web del Grau de Multimèdia de la UOC i tracta el procés de desenvolupament d'una aplicació web relacionada amb la cerca de feina aplicant els coneixements adquirits durant el Grau des de la planificació inicial, el disseny de la interfície, la maquetació, la programació i la seva implementació final en un servidor web.

El plantejament d'aquest treball de final de grau és desenvolupar des de zero una aplicació web focalitzada en posar en contacte a persones que ofereixen les seves habilitats amb altres que les puguin necessitar.

L'usuari només s'ha de crear un perfil i ja pot oferir els seus serveis o cercar els que necessiti. I per a agilitzar la creació del perfil en un sol clic es pot registrar mitjançant les APIs de Facebook o de Twitter, una característica molt emprada en els últims temps.

El punt fort es trobarà en la localització dels usuaris segons la distància, de manera que es puguin establir vincles amb persones properes: potser el pròxim client, o la solució al problema, es troba just a l'edifici del costat.

Alhora d'escollir un candidat per la feina, l'usuari es pot orientar per un sistema de vots i comentaris deixats per altres usuaris que ja han tingut una experiència laboral prèvia amb el candidat. Per polir detalls es compta amb un sistema de xat, el segon punt fort de l'aplicació, que permet que els usuaris puguin comunicar-se en temps real.

La plataforma està desenvolupada utilitzant íntegrament Javascript tant per la banda del client com la del servidor utilitzant el paquet MEAN i està basada en el plantejament *Mobile First*¹, motiu pel qual s'adapta perfectament a dispositius mòbils. De fet l'objectiu final és que, en una fase posterior a l'entrega del treball de final de grau, s'acabi convertint en una aplicació mòbil.

¹ El concepte *mobile first* consisteix en dissenyar les interfícies pensant en els terminals mòbils i posteriorment adaptar-los a pantalles més grans

3. Objectius

L'objectiu general del projecte és el de crear **faenio**, una aplicació web del tipus plataforma col·laborativa orientada a la recerca de feina en un estat completament funcional i estable.

A continuació es llisten els objectius principals i secundaris del treball de final de grau fixats en la primera fase de definició i planificació.

3.1 Objectius principals

- Desenvolupar una aplicació web que permeti als usuaris publicar les seves candidatures oferint el seus serveis com a professionals per a feines puntuals
- Produir un producte viable per a l'ús públic que doni resposta a un problema real
- Dissenyar un producte atractiu i usable
- Mostrar l'assoliment de l'aprenentatge que s'ha dut a terme al llarg dels estudis del Grau Multimèdia

3.2 Objectius secundaris

- Implementar la geolocalització per tal que els usuaris puguin cercar els resultats basant-se en la seva ubicació i d'aquesta manera poder trobar una solució més immediata
- Oferir la possibilitat de registrar-se fent servir xarxes socials com Facebook o Twitter
- Utilitzar noves tecnologies relacionades amb el desenvolupament web no tractades durant el Grau i aprendre'n el seu funcionament
- Crear una aplicació referent que permeti als usuaris trobar una solució ràpida i propera quan els sorgeixi una necessitat que requereixi la intervenció d'un professional
- Desenvolupar una aplicació que s'adapti perfectament a un dispositiu mòbil per a la seva posterior compilació a aplicació Android

4. Continguts

En aquest apartat es defineixen els continguts de l'aplicació tant a nivell estructural (continguts físics amb els que es trobarà l'usuari) com a nivell funcional (què es pot fer amb l'aplicació).

4.1 Estructurals

A continuació es defineixen els continguts dels diferents apartats de l'aplicació, posant especial èmfasi en les pàgines principals (inici, llistat, perfil d'usuari i detall del servei).

Pàgina d'inici

- Bloc introductor i amb una frase de presentació i dos botons: "Registra't", que du al formulari de registre, i "com funciona", que dirigeix a una altra pàgina amb una definició més exhaustiva
- Cercador de publicacions amb opció de filtre addicional per localització
- Resum visual de la funcionalitat de l'aplicació
- Llistat amb els darrers anuncis publicats a l'aplicació i un botó "Veure tots els anuncis" per accedir al llistat complet d'anuncis
- Llistat de totes les categories donades d'alta a l'aplicació per a què l'usuari pugui filtrar la seva cerca segons els seus interessos concrets
- Informació numèrica sobre l'estat de l'aplicació: nombre de clients que la utilitzen, nombre de professionals que ofereixen els seus serveis i nombre d'anuncis publicats en total

Pàgina de llistat

- *Breadcrumbs* (molles de pa) dels nivells superiors als que pertany la publicació segons la cerca que s'hagi realitzat. Pot mostrar localització, categoria i paraula de cerca.
- Llistat de les publicacions resultants d'una cerca o d'una categoria
- Cada element de publicació mostra:
 - o Nom de la categoria a la qual pertany
 - o Imatge representativa de la categoria (en futures versions es mostrarà una imatge proporcionada per l'usuari, integrada a la publicació)
 - o Titular de la publicació continguda en un enllaç que dirigeix a la pàgina de detall de la mateixa
 - o Preu del servei
 - o Fotografia de l'usuari autor de la publicació (si en té) o una icona genèrica. Aquesta estarà continguda en un enllaç que dirigeix a la pàgina de perfil de l'autor

- Puntuació mitjana de l'autor representada amb un sistema d'estrelles (puntuació de l'1 al 5) i recompte del total de valoracions obtingudes
- En les versions d'escriptori es mostra un mapa a la dreta amb la ubicació dels resultats

Pàgina de detall

- *Breadcrumbs* (molles de pa) dels nivells superiors als que pertany la publicació, tant pel que fa a la categoria de l'anunci com a la localització
- Titular de la publicació
- Categoria a la qual pertany
- Descripció de la publicació
- Detalls d'interès de la publicació:
 - Preu del servei o del cost/hora
 - Termini de finalització del servei
 - Formes de pagament admeses²
- Botó per a enviar un missatge a l'autor de la publicació
- Mapa amb la localització aproximada del servei o de l'autor (segons el tipus d'anunci)
- Possibilitat de compartir la publicació via Facebook, Twitter, Google+ i LinkedIn
- Informació sobre l'autor de la publicació:
 - Nom i descripció curta
 - Fotografia (si en té) o icona genèrica
 - Descripció amb una breu biografia
 - Localització
 - Data de registre a l'aplicació
 - Valoracions emeses per altres usuaris arrel de serveis anteriors realitzats

Pàgina de perfil

- Fotografia de l'usuari (si en té) o icona genèrica
- Nom i descripció curta
- Descripció amb una breu biografia
- Possibilitat d'editar el perfil
- Informació addicional: localització, data de registre i data de la darrera connexió
- Informació sobre l'activitat de l'usuari: Nombre de serveis adjudicats i de serveis rebuts
- Possibilitat de compartir el perfil amb tercers (via Facebook, Twitter, Google+ i LinkedIn)
- Llistat (si en té) d'ofertes publicades i possibilitat d'editar-les o eliminar-les

² Funcionalitat de futura implementació en una propera versió de l'aplicació

- Llistat (si en té) de valoracions rebudes d'altres usuaris
- Llistat (si en té) de contactes rebuts d'altres usuaris a partir dels anuncis publicats amb un botó per a poder assignar el servei al client
- Llistat (si en té) de serveis rebuts amb un botó per a poder valorar i comentar l'experiència

Altres pàgines

S'han definit amb detall les pàgines principals de l'aplicació. A continuació s'enumeren altres pàgines que també són importants, però que se'n poden resumir els continguts a grans trets:

- **Sobre nosaltres:** Què es **faenio**, identificació fiscal i dades de contacte
- **Política de privacitat:** Informació legal requerida en qualsevol lloc web on es tracti amb dades sensibles dels usuaris
- **Ajuda:** Descripció del funcionament de l'aplicació en detall
- **Edició de perfil:** Formulari amb les dades de l'usuari que poden ser editades
- **Creació/edició d'una publicació:** Formulari amb la informació relacionada amb la publicació d'un servei a la plataforma
- **Els meus missatges:** Llistat de les diferents converses obertes amb altres usuaris. Al fer clic sobre qualsevol dels elements enllaça a la pàgina de xat, on es veu l'historial de missatges
- **Xat:** Cadascuna de les converses mantingudes amb altres usuaris, on es mostra l'historial de missatges ordenat cronològicament (zona central). A la part inferior s'hi troba el formulari d'enviament de missatges en temps real i a la barra lateral es mostra la informació bàsica de l'interlocutor amb qui s'està mantenint la conversa

Elements comuns

Capçalera

- Si l'usuari no ha iniciat sessió, es mostra un enllaç de registre i un d'inici de sessió
- Si l'usuari ha iniciat sessió es mostra el seu nom i un accés al seu menú d'usuari
- A totes les pàgines tret de la pàgina d'inici es mostra el cercador de publicacions a la capçalera (a la pàgina d'inici es mostra en una altra ubicació)

Peu de pàgina

- Enllaços a pàgines internes: "Sobre nosaltres", "Política de privacitat" i "Ajuda"
- Enllaços per a compartir la pàgina en xarxes socials (Facebook i Twitter)

Finestra modal de registre: Formulari amb els camps principals per a registrar-se a la plataforma (un cop realitzat el registre es poden completar la resta d'informacions). També es permet la opció de registre mitjançant Facebook i Twitter

Finestra modal d'inici de sessió: Formulari d'inici de sessió (usuari i paraula de pas) i possibilitat de realitzar l'autenticació mitjançant xarxes socials (Facebook i Twitter)

4.2 Funcionals

El que es descriu en aquesta secció són les funcionalitats de l'aplicació, és a dir, tot allò que l'usuari pot fer amb l'eina. I el que pot fer és:

- Registrar i iniciar sessió a l'aplicació, tant a través d'un e-mail i contrasenya com mitjançant Twitter i Facebook (mètode que crea un perfil d'usuari agafant la informació bàsica i imatge de perfil proporcionades per les plataformes)
- Crear publicacions de serveis, assignant una categoria i una localització
 - o Possibilitat que la localització s'indiqui manualment o l'aplicació trobi les coordenades de l'usuari en aquell moment³
 - o Possibilitat que les publicacions tinguin un preu per hora o un preu per servei complet
- Editar la informació de perfil de l'usuari
- Editar i eliminar publicacions
- Cercar publicacions per localització, per categoria i per paraules clau
- Visualitzar dels resultats obtinguts de la cerca en un mapa
- Enviar missatges entre usuaris en temps real
- Veure l'historial de missatges enviats i rebuts a través del menú "Els meus missatges"
- Veure l'historial de serveis rebuts
- Veure l'historial de serveis realitzats
- Publicar comentaris sobre serveis amb una valoració numèrica (de l'1 al 5) que es mostra en forma de estrelles
- Compartir una pàgina de perfil d'un usuari a Facebook, Twitter, Google+ i LinkedIn
- Compartir una pàgina de detall d'un servei a Facebook, Twitter, Google+ i LinkedIn

³ Coordenades aproximades basades en les dades proporcionades pel navegador

5. Metodologia

Per a la realització del projecte s'utilitza una metodologia clàssica o en cascada basada en el diagrama de Gantt⁴. Tot i que en el desenvolupament de programari orientat a clients acostuma a ser més productiva una metodologia de desenvolupament àgil⁵ (*Kanban, Scrum, Lean, etc.*), aquest cas es presenta com una excepció, ja que els requisits estan definits des d'un inici i previsiblement no canviaran durant el cicle de vida del desenvolupament, permetent una estimació de terminis i costos més acurada. És per aquest motiu que s'ha optat per un model en cascada.

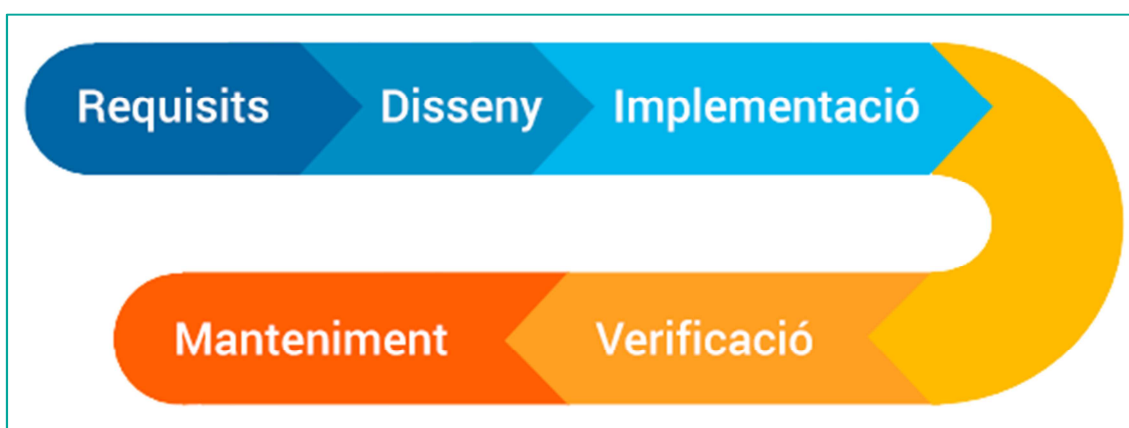


Figura 1. Desenvolupament en cascada

Els processos de desenvolupament en cascada acostumen a ser més segurs per estar orientats a la planificació, permetent l'assoliment amb més seguretat de les fites principals del projecte, equivalents als lliuraments de l'avaluació continua.

En les entregues de l'avaluació s'espera rebre un feedback per part del consultor de l'assignatura que permeti millorar la qualitat del producte final i a cada entrega es durà a terme una revisió de la planificació establerta per detectar i corregir possibles desajustos.

En cadascuna de les entregues es revisarà també el contingut de la present memòria i s'hi afegiran els nous apartats que corresponguin al lliurament en qüestió.

⁴ El diagrama de Gantt del projecte al que es fa menció es pot trobar a l'apartat 9.2 de la present memòria, dins de la planificació del projecte

⁵ Metodologia àgil: https://ca.wikipedia.org/wiki/Metodologia_àgil

6. Arquitectura de l'aplicació

El desenvolupament de l'aplicació està centrat en una arquitectura basada en Javascript en totes les seves capes utilitzant el que s'ha denominat *MEAN stack*, un conjunt de solucions informàtiques de programari lliure i que està format per MongoDB, ExpressJS, AngularJS i NodeJS.

6.1 Client

Del costat del client s'utilitza AngularJS, una llibreria basada en l'arquitectura MVC (Model-Vista-Controlador). Aquesta capa de l'aplicació s'encarrega de gestionar la interacció del client amb l'aplicació disposant de la lògica necessària per a mostrar i gestionar la interfície així com a executar la comunicació amb el servidor.

En aquesta capa es treballa també amb Bootstrap que consta de multitud d'eines visuals i d'interacció que acceleren el desenvolupament de la interfície gràfica, així com també amb un seguit de mòduls i llibreries de tercers que faciliten la feina del desenvolupament (veure Annex 3. Llibreries externes per al detall complet).

6.2 Servidor

La plataforma NodeJS permet utilitzar el llenguatge de programació Javascript en la banda de servidor i permet realitzar tasques com l'encaminament o la creació de pàgines dinàmiques.

A la banda de servidor s'hi troba també ExpressJS, una llibreria de Javascript que gestiona les sol·licituds i les respostes que el client (l'aplicació creada en AngularJS) s'encarrega de generar. És a dir, du a terme les tasques d'encaminament.

6.3 Bases de dades

La capa de persistència de l'aplicació disposa d'una base de dades no relacional anomenada MongoDB. El concepte d'aquest tipus d'arquitectura de base de dades es desar la informació en una estructura de col·leccions de documents de tipus JSON enlloc de desar les dades en taules com les bases de dades relacionals (com ara SQLServer, MySql, PostgreSQL, etc.).

MongoDB permet tractar amb grans volums de dades i té una gran facilitat per a l'escalabilitat. Una aplicació com **faenio** pot arribar a generar grans quantitats de dades. A més està altament enfocat a la indexabilitat de les dades i compta amb índexs geoespacionals, quelcom molt important en una aplicació com **faenio** que un us important d'aquest tipus de dades.

7. Plataforma de desenvolupament

En aquest apartat es detallen les eines que s'han utilitzat en els processos de disseny i desenvolupament de **faenio**, tant a nivell de programari com de maquinari.

7.1 Programari

Adobe Illustrator CS6

Per al disseny del logotip i per a l'elaboració dels prototips, tant de baixa com alta resolució, es treballa també amb Adobe Illustrator. Tot i que hi ha altres eines més enfocades a fer *wireframes* amb facilitat (com Axure o Moqups), s'ha optat per seguir tot el procés amb el mateix programari de manera que, en l'etapa de prototipatge en alta definició es pot aprofitar part del material ja generat, sense necessitat de començar de zero.

URL: <http://www.adobe.com/es/products/illustrator.html>

NodeJS

Apart d'utilitzar-lo com a servidor de l'aplicació, NodeJS i el seu gestor de paquets npm (Node Package Manager) permet instal·lar aplicacions que resulten indispensables per a treballar amb el projecte, com Mongo DB, ExpressJS i Stylus. S'instal·la la versió 4.4.0.

URL: <https://nodejs.org/en/>

MongoDB

Per a treballar amb un servidor local, cal instal·lar MongoDB a la màquina de treball. L'instal·lador està disponible per Windows, Linux, OSX i Solaris. En aquest cas s'instal·la la versió 3.2.4 per a Windows 64 bits. URL: <https://www.mongodb.org/downloads>

ConEmu

Es tracta d'una millora de la consola de Windows que aporta funcionalitats que fan l'experiència amb la línia de comandos més agradable similar a les consoles de Linux. Permet obrir diverses finestres. URL: <https://sourceforge.net/projects/conemu/>

Bitbucket

Aquest gestor de control de canvis basat en la tecnologia Git i permet sincronitzar la feina feta amb un servidor on es registren els canvis que s'han realitzat a cada arxiu, permetent tornar enrere cap a una versió anterior en qualsevol punt del procés de treball. Bitbucket permet tenir repositoris privats gratuïts, i per això ha estat escollit per sobre del seu competidor més conegut, GitHub. URL: <https://bitbucket.org/>

SourceTree

Client de Git i Mercurial per a administrar de manera visual tots els repositoris. Tot i que es pot treballar amb git només utilitzant la consola de comandes, SourceTree resulta molt útil per a tenir tots els repositoris organitzats i per a veure'n l'estat general en un sol cop d'ull.

URL: <https://www.sourcetreeapp.com/>

Sublime Text 3 + Emmet

Sublime Text és un editor de text que permet treballar amb múltiples fitxers a la vegada i que disposa d'un codi de colors que millora la llegibilitat.

S'hi poden instal·lar diversos plugins mitjançant el gestor de paquets Package Manager, i un dels que resulten més útils alhora de maquetar és Emmet, que facilita l'escriptura auto-completant el codi que s'utilitza més sovint.

URL Sublime Text: <http://www.sublimetext.com/> | URL Emmet: <http://emmet.io/>

Stylus

Aquest preprocessador de CSS permet escriure codi d'una manera molt més ràpida que el sistema CSS tradicional. Permet utilitzar variables, importar altres fitxers stylus i compilar-ho tot en un únic CSS minificat. URL: <http://stylus-lang.com/>

Trello

Es tracta d'una aplicació que permet organitzar projectes de manera visual. Té versió tant d'escriptori com *app* per a mòbil, i en el transcurs del projecte s'han utilitzats exhaustivament ambdues versions. És una eina realment útil que permet administrar la llarga llista d'activitats del projecte així com a indicar alertes de dates de finalització de cadascuna, podent saber així si s'està seguint la planificació establerta o si hi ha desviaments. URL: <https://trello.com>

7.2 Maquinari

El projecte s'ha treballat utilitzant de forma alternada un ordinador de sobretaula Acer amb un processador Intel Core i3 a 2.93GHz i 6GB de RAM sobre un Windows 7 Professional i un ordinador portàtil Lenovo amb un processador Intel Core i5 a 1.7GHz i 8GB de RAM sobre un Windows 7 Professional.

El fet de treballar amb un sistema de control de versions (Bitbucket + SourceTree) fa que treballar en màquines diferents no sigui gens incòmode, ja que la sincronització de dades és pràcticament immediata.

8. Planificació

El projecte es desenvoluparà des del Febrer fins al Juny de 2016, coincidint amb l'inici i el final del semestre, amb una dedicació estimada de 14 hores setmanals (unes 280 hores totals) amb previsió de dedicar dues hores diàries.

8.1 Dates clau

La planificació del projecte ve condicionada per les dates d'entrega de les diferents PAC així com el lliurament final del projecte que s'han de respectar i complir sense marge de flexibilitat.

Aquestes dates són les següents:

Activitat	Data
Entrega PAC1	08/03/2016
Entrega PAC2	06/04/2016
Entrega PAC3	08/05/2016
Lliurament final	13/06/2016

Taula 1. Dates clau

8.2 Diagrama de Gantt

A continuació es presenta el diagrama de Gantt amb les activitats previstes per aquest projecte basat en jornades de dues hores:

Tasques	Duració (dies)	Inici	Final
PAC1		24/02/2016	08/03/2016
Definició del projecte	10	24/02/2016	04/03/2016
Pla de treball	1	05/03/2016	05/03/2016
Selecció i compra del domini	1	06/03/2016	06/03/2016
Redacció de la memòria	2	07/03/2016	08/03/2016
PAC2		09/03/2016	06/04/2016
Pressupost	1	09/03/2016	09/03/2016
Anàlisi de mercat	1	10/03/2016	10/03/2016

Viabilitat	1	11/03/2016	11/03/2016
Diagrames UML i casos d'us	4	12/03/2016	15/03/2016
Model E/R	2	16/03/2016	17/03/2016
Perfils d'usuari	1	18/03/2016	18/03/2016
Usabilitat/UX	1	19/03/2016	19/03/2016
Disseny de Wireframes de baixa definició	4	20/03/2016	23/03/2016
Disseny de Wireframes d'alta definició	7	24/03/2016	30/03/2016
Actualització de la memòria	7	31/03/2016	06/04/2016
PAC3		07/04/2016	08/05/2016
Instal·lació servidor local de proves	1	07/04/2016	07/04/2016
Maquetació web (HTML5, CSS3)	6	08/04/2016	13/04/2016
Programació frontend	6	14/04/2016	19/04/2016
Implementació de la base de dades	1	20/04/2016	20/04/2016
Programació backend	13	21/04/2016	03/05/2016
Actualització de la memòria	5	04/05/2016	08/05/2016
FINAL		09/05/2016	13/06/2016
Finalització maquetació i programació	10	09/05/2016	18/05/2016
Implementació al servidor final	1	19/05/2016	19/05/2016
Tests i correccions	6	20/05/2016	25/05/2016
Actualització de la memòria	7	26/05/2016	01/06/2016
Vídeo de presentació	7	02/06/2016	08/06/2016
Realització de la presentació virtual	4	09/06/2016	12/06/2016
Lliurament final	1	13/06/2016	13/06/2016

Taula 2. Diagrama de Gantt

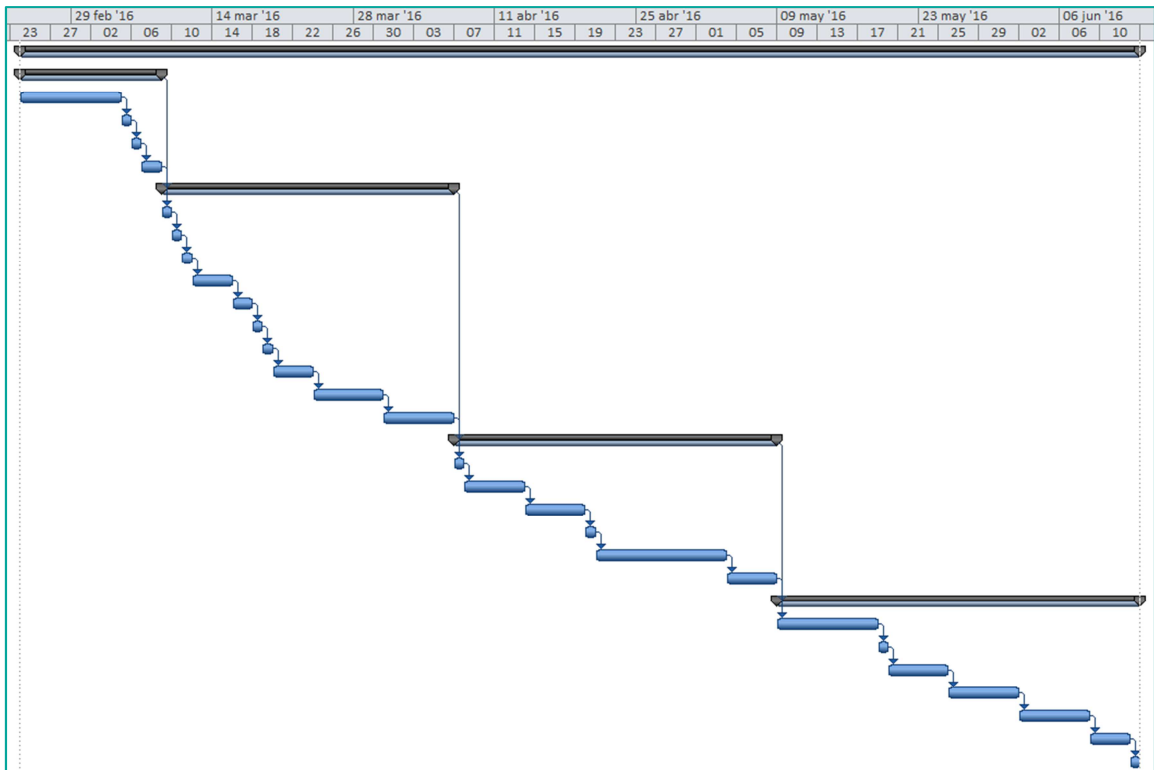


Figura 2. Diagrama de Gantt

9. Procés de treball

A continuació es detalla el procés que s'ha seguit per al desenvolupament de **faenio** dividit entre les diferents entregues i els problemes que s'han presentat i que no estaven contemplats a la planificació inicial.

9.1 PAC1

La primera entrega del projecte tracta, sobretot, de la definició del projecte: les seves característiques principals, els objectius que es volen assolir i, sobretot, la definició de les activitats i la planificació d'aquestes en el temps disponible: un trimestre.

Aproximadament el 80% del temps previst per aquesta entrega s'ha destinat en pensar en la idea del projecte, ja que al principi van aparèixer infinitat de dubtes que ocasionaven un bloqueig mental sever a l'autora. Temps que realment es podia haver aprofitat per avançar en alguna de les moltes tasques a dur a terme, ja que de totes les opcions de possibles projectes a desenvolupar, s'ha acabat realitzant la primera de totes.

Per al nom de l'aplicació s'ha fet una pluja d'idees, de entre les quals s'ha escollit "**faenio**" perquè compleix els requisits que s'havien definit:

- Disponibilitat del domini .COM
- Un nom breu, de 7 lletres o menys
- Quelcom relacionat amb la feina

Després de seleccionar el nom, se n'ha registrat el domini genèric amb extensió ".com" al proveïdor Hostalia⁶. Es valora la opció de registrar en un futur l'extensió ".cat", ja que el contingut de l'aplicació està redactat en català.

Un cop es té acotada la idea principal, se'n defineixen les activitats a realitzar i es divideixen en el temps. Des del primer moment es va veure que no seria una planificació senzilla i que el compliment tampoc ho seria. Per una banda, la planificació no compta amb cap dia de descans: segons el diagrama de Gantt es treballa cada dia durant 3 mesos incloent caps de setmana i festius. Es fa expressament perquè és la única manera d'abastar totes les activitats

⁶ Hostalia: <https://www.hostalia.com/>

definides, i de totes maneres certes fites complexes com “programar el *frontend* i el *backend*” tenen destinades la meitat (o menys) d'hores que haurien de tenir en condicions normals.

9.2 PAC2

La segona entrega es divideix en tres parts. Per una banda hi ha la preparació del projecte, que abasta el pressupost, l'estudi de mercat, l'anàlisi de viabilitat, etc. Per altra banda hi ha la fase de disseny, que alhora també està dividida en dues parts: el disseny de l'enginyeria del projecte (definició del model de dades, anàlisi de casos d'us, diagrames UML,...) i per altra banda el disseny gràfic i dels prototips.

Com s'esperava, la planificació ha estat dura de complir. S'ha pogut seguir bé en quant a terminis de les fites, però en moltes jornades s'han hagut de dedicar hores extra. L'aplicació Trello⁷ resulta de gran ajuda per a calendaritzar les activitats.

Es defineixen les classes a utilitzar i es realitza un diagrama UML de model entitat relació, però després de fer els prototips de baixa resolució es troben aspectes a millorar i s'afegeixen al model de dades.

La fase de disseny gràfic comença amb el disseny del logotip. A partir d'aquí ja es tenen els colors per a poder treballar els prototips d'alta resolució. I un cop es tenen els prototips també es pot definir la guia d'estil i apartats de la memòria com el d'usabilitat.

9.3 PAC3

En aquesta ocasió el 99% del treball s'ha centrat en el desenvolupament de l'aplicació. S'han maquetat les diferents pàgines, s'hi ha aplicat estils, s'han dotat de comportament i s'ha desenvolupat el *backend*. En aquest moment un usuari es pot registrar, iniciar sessió i publicar un nou anunci de servei.

A aquestes alçades del projecte la planificació encara ha estat més dura de complir que en la fita anterior. Ja tenia coneixements de Javascript, però no d'Angular, MongoDB, Express i MongoDB, i el fet de realitzar un projecte utilitzant tecnologies desconegudes i aprendre-les en

⁷ **Trello** és una aplicació que ajuda a organitzar projectes de manera visual. Permet administrar la llarga llista d'activitats del projecte així com a indicar alertes de dates de finalització de cadascuna, podent saber així si s'està seguint la planificació establerta o si hi ha desviaments. <https://trello.com>

paral·lel durant el desenvolupament ha estat un repte complicat de superar i per tant el temps dedicat ha estat molt superior al planificat, mentre que la funcionalitat que es volia arribar a tenir en el moment d'aquesta entrega no s'ha completat al 100%.

Apart de necessitar més temps de l'esperat per al desenvolupament de l'aplicació també m'he trobat amb que al haver de treballar amb aplicacions desconegudes per a mi he hagut d'avançar fent moltes proves-error (especialment al principi) i tinc la sensació que el codi no es tot lo òptim que hauria de ser i que algunes funcionalitats es podrien millorar.

La intenció es fer un *refactoring* abans del lliurament final i deixar-ho tot tant eficient com sigui possible, però en vista de tota la feina que hi ha pendent per a fer em veig obligada a seguir amb la filosofia "**better done than perfect**". És una filosofia que no acostumo a dur a terme perquè sóc bastant perfeccionista, però quan hi ha una data d'entrega inamovible és la opció a seguir.

En aquesta fita, la redacció de la memòria s'ha deixat una mica en segon pla per a centrar pràcticament tots els esforços en el desenvolupament de l'aplicació, per tant les modificacions que s'han fet a la memòria no han estat excessivament nombroses.

Juntament amb la memòria i una primera versió de l'aplicació, es lliura un document en el que s'indica com utilitzar la versió alfa de **faenio**, especificant fins a on s'ha arribat en el desenvolupament, així com les funcionalitats que encara estan pendents de desenvolupar.

9.4 Lliurament final

Aquesta etapa ha estat la més dura i en la que la planificació s'ha hagut de modificar sobre la marxa en diverses ocasions, tant per funcionalitats que han costat més de desenvolupar del que s'havia previst, com problemes nous que han anat sorgint a mida que s'avançava, com interrupcions externes que també m'han robat temps.

En el lliurament de la PAC3 van quedar moltes funcionalitats importants pendents de desenvolupar i he arribat a tenir dubtes de si podria arribar a obtenir un producte final digne de ser presentat davant un tribunal, però les moltes hores de dedicació han donat els seus fruits i els objectius principals i secundaris s'han assolit de manera satisfactòria.

Així doncs, en relació al progrés de l'aplicació, les funcionalitats desenvolupades en el transcurs d'aquesta darrera etapa han estat les següents:

- Resoldre *bugs* pendents de l'entrega anterior
- Cerca de publicacions per localització
- Cerca de publicacions per categoria (i localització)
- Generar i mostrar *breadcrumbs* reals
- Enviament de missatges entre usuaris en temps real
- Marcar serveis com a realitzats (assignar un servei a un usuari client)
- Publicació de comentaris sobre serveis i mostra d'informació real dels comentaris als perfils i a les publicacions
- Possibilitat d'editar la informació de perfil de l'usuari
- Possibilitat d'editar i eliminar publicacions
- Inici de sessió i registre d'usuari mitjançant Facebook i Twitter, agafant la informació bàsica i imatge de perfil proporcionades per les plataformes
- Afegir validacions en el registre d'un nou usuari
- Possibilitat de compartir una pàgina de perfil d'un usuari o una pàgina de detall d'un servei a Facebook, Twitter, Google+ i LinkedIn
- Acabar d'adaptar l'aplicació a dispositius mòbils així com els diferents navegadors
- Mostrar correctament els serveis al mapa
- Comptador de visites de les pàgines de detall dels serveis
- Desar (i mostrar) puntuació mitjana d'un usuari i puntuació mitjana d'un servei
- Desar (i mostrar) data de la darrera connexió de l'usuari
- Afegir validacions de *frontend* en el formulari de publicació i en el d'edició de perfil
- Afegir llistat de categories a la pàgina d'inici
- Mostrar el mapa amb els resultats de cerca centrat en el punt geogràficament central entre tots els resultats obtinguts

Malauradament, tot i haver complert amb els objectius principals i secundaris del treball, el gran abast del projecte i el temps disposat han provocat que algunes de les funcionalitats secundàries previstes en un inici s'hagin relegat a una futura revisió de l'aplicació, que serien les que es llisten a continuació:

- Sistema per a recordar la contrasenya de l'usuari
- Possibilitat que els usuaris puguin afegir la seva imatge de perfil
- Resoldre el 100% dels *bugs* detectats⁸

⁸ Veure apartat 19. *Bugs* de la present memòria

- Incloure els mètodes de pagament als serveis
- Compartir perfil/publicació per e-mail
- Activar/desactivar serveis

Un altre aspecte important a destacar es que degut a problemes logístics⁹ finalment l'aplicació s'ha centrat només en la ciutat de Barcelona

En relació a l'evolució del present document de memòria del projecte, abans de realitzar el lliurament final se n'ha repassat i editat el contingut en la seva totalitat tant per a completar contingut inacabat, com per a corregir errates i també per a que s'ajusti a la realitat del producte final, ja que hi ha certs punts que han variat respecte a la definició inicial.

Durant l'etapa de desenvolupament s'han pres certes decisions que han afectat a la planificació inicial i que implicaven modificacions sobre els dissenys mostrats en els prototips, i per tant, tots aquests canvis s'havien de documentar. Per una banda s'han descartat les funcionalitats que s'ha comentat més amunt, i per altra s'ha afegit el següent:

- Afegir el llistat de categories a la pàgina d'inici
- Mostrar les estrelles al perfil de l'usuari i a les pàgines de detall dels serveis indicant de forma gràfica la seva valoració mitjana
- A la pàgina de perfil d'usuari mostrar el llistat de contactes rebuts d'una publicació per tal de poder assignar el servei al client
- A la pàgina de perfil d'usuari mostrar el llistat de serveis que li han estat assignats per a que aquest en pugui emetre'n una valoració
- Al cercador de la pàgina d'inici, afegir selector de localització per a filtrar geogràficament
- Comptador de visites en les pàgines de publicació
- Afegir la possibilitat de compartir publicacions i perfils també a Google+ i LinkedIn
- Mostrar informació de la darrera connexió de l'usuari, dins la seva pàgina de perfil

També ha estat necessari documentar a la memòria la necessitat d'afegir el model "conversa" entre usuaris (ChatRoom), per tant s'ha refet el diagrama UML del model de dades (que també s'ha actualitzat per afegir nous atributs derivats de les noves funcionalitats) i se n'ha redactat la funcionalitat.

⁹ Queda pendent de resoldre la possibilitat d'afegir localitzacions noves a la base de dades en forma de polígons de coordenades cada cop que es publiqui un servei si aquestes no existeixen. Les que s'han creat ha estat de forma manual, però encara no s'ha trobat la manera de fer-ho automàticament.

10. APIs utilitzades

Apart de les APIs de tercers que tot seguit es comentaran, cal assenyalar que l'aplicació s'ha desenvolupat de tal manera que ella mateixa sigui una API RESTful, que estarà encarregada de gestionar les dades introduïdes pels usuaris. Així per exemple, amb la ruta */post* seguit d'un identificador, mitjançant el mètode GET mostrarà el servei demanat, mentre que mitjançant PUT l'actualitzarà o mitjançant DELETE eliminarà el registre.

D'aquesta manera es facilita l'escalabilitat del projecte i se li dona una lògica interna més ben definida, separant la lògica del client de la del servidor obtenint així un més baix acoblament.

10.1 Google Maps

S'utilitza la **Google Maps API** en la seva versió 3, que permet utilitzar la completa tecnologia cartogràfica creada per Google i que facilita una sèrie de funcionalitats per al treball amb mapes:

- Geolocalització d'adreces directa (introduint una adreça) o inversa (trobar el nom d'un lloc introduint la latitud i la longitud)
- Presentació de mapes amb diverses capes de visualització diferents
- Càlcul de distància entre dos punts¹⁰

10.2 Facebook i Twitter

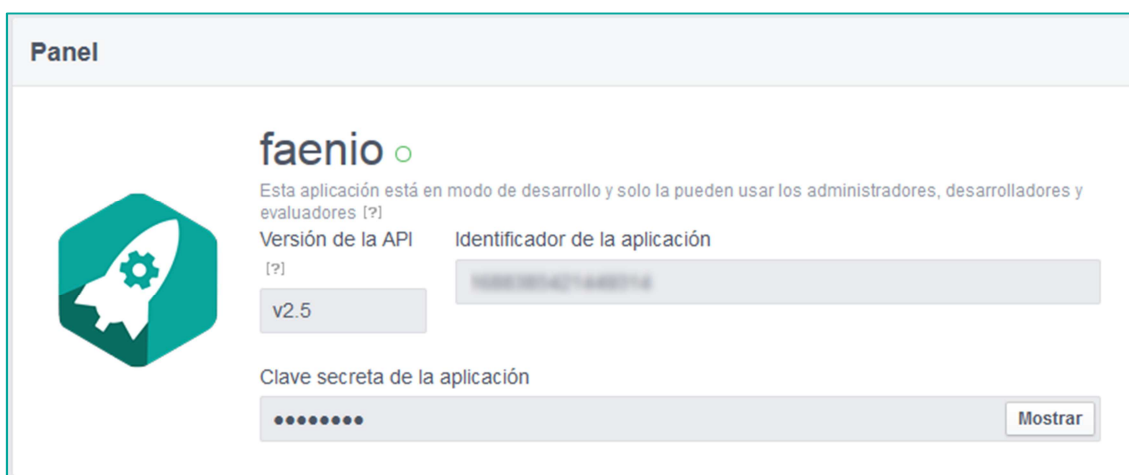


Figura 3. API de Facebook

¹⁰ Aquesta característica finalment només s'utilitzarà en una propera versió de l'aplicació en la que es millorarà l'algoritme de cerca. Actualment els resultats estan localitzats dintre d'un polígon de coordenades

Per al registre i l'autenticació d'usuaris a la plataforma s'ha utilitzat la llibreria passport que ahora fa us de l'API de Facebook i Twitter, que mitjançant el protocol obert oAuth, donen la possibilitat d'identificar un usuari.

```
if (err) {
  return done(err);
}
if (!user) {
  user = new User({
    firstName: profile.name.givenName,
    lastName: profile.name.familyName,
    displayName: profile.displayName,
    email: profile.emails[0].value,
    imageUrl: 'https://graph.facebook.com/' + profile.id + '/picture?height=200&width=200',
    role: 'user',
    username: profile.username,
    provider: 'facebook',
    facebook: profile._json
  });
  user.save(function(err) {
    if (err) done(err);
    return done(err, user);
  });
} else {
  return done(err, user);
}
})
```

Figura 4. Fragment de codi de passport.js on es passa la informació del perfil de Facebook al nou usuari

Apart de realitzar l'autenticació també es dona accés a la informació pública compartida del seu perfil en aquestes xarxes socials. Així de Facebook se n'obté el nom i cognoms de l'usuari, l'email i la imatge de perfil, mentre que de Twitter se n'obté el nom d'usuari, el nom, la descripció i la imatge de perfil, informacions que es desen en el nou usuari que es crea pràcticament de forma automàtica.

11. Base de dades

11.1 Model Entitat Relació

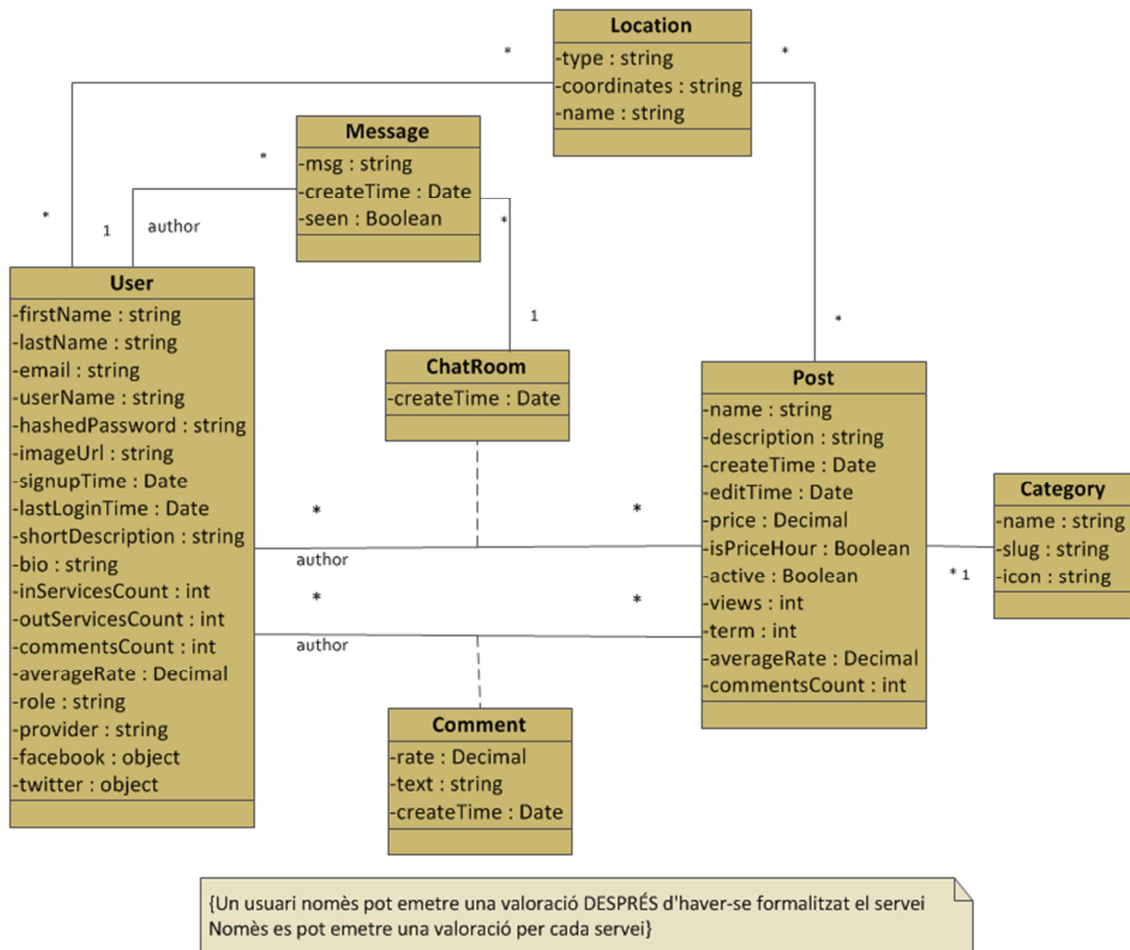


Figura 5. Diagrama d'entitat relació

User i **Post** són dos models primaris sobre els que es basen la majoria d'accions de l'aplicació, ja que tot està relacionat amb ells.

User es tant el professional que oferta els seus serveis com el client que els demanda i **Post** és la oferta d'un servei.

Els usuaris s'envien missatges entre ells a través de converses, que són les sales de xat (**ChatRoom**), i aquestes converses van directament relacionats amb **Posts**: per cada servei del que un usuari vulgui més informació es crea una nova conversa en la que **User** és l'usuari emissor i l'autor del servei és l'usuari receptor. Els missatges (**Message**) estan vinculats a sales de xat i a **User**, que és l'autor del missatge.

Els comentaris (**Comment**) estan relacionats tant amb **Usuari** com **Publicació** (són una relació associativa) i la restricció que un usuari nomès pugui emetre una valoració després d'haver-se formalitzat el servei es controla des de la capa de negoci.

11.2 Creació de la base de dades

A MongoDB no es crea una base de dades per comandes de l'estil *createDB* o *createDataBase*, sinó que es crea en el mateix instant que s'insereix informació en aquesta base de dades, és a dir, quan s'afegeix un element dins d'una col·lecció¹¹.

Per a crear per exemple un usuari de la col·lecció *usuaris* es podria fer de la manera següent:

```
// independentment de si la base de dades "faenio" existeix o
// no se li encomana que l'utilitzi
use faenio

// s'afegeix un nou usuari, que pot no tenir l'estructura
// marcada en el model de dades
db.users.save({nom: 'Anna', cognoms: 'Ferry'});
```

Després de fer això ja hi ha creada tant la base de dades (**faenio**) com la col·lecció de dades (**User**), que prèviament no existien.

11.3 Generació del contingut de la base de dades

A l'apartat anterior s'ha vist com afegir contingut a la base de dades de manera manual, que és en realitat com s'ha afegit la major part del contingut de l'aplicació: el generen els propis usuaris que es registren i interactuen amb l'eina. No obstant, també hi ha altres opcions per a afegir noves dades, que es veuran a continuació.

Generació automàtica (*seed.js*)

A l'executar l'aplicació durant la fase de desenvolupament, es realitzava una crida a un fitxer *seed.js* que s'encarregava de publicar una sèrie d'informació bàsica a la base de dades de manera que sempre hi ha informació sobre la que treballar i fer les proves còmodament.

Aquesta injecció de dades consistia en dos usuaris de prova (model *User*), dos serveis (model *Post*), diverses categories (model *Category*), un servei assignat a un dels usuaris, un servei rebut per l'altre usuari, i una valoració de servei (model *Comment*) desada tant al servei com a l'usuari.

¹¹ En un sistema de base de dades no relacionals (NoSQL) es tracta amb col·leccions (grup de documents) en lloc del sistema de taules utilitzat en les bases de dades relacionals

En la versió de producció aquesta funcionalitat s'ha desactivat, ja que cada cop que s'inicia l'aplicació s'esborren els altres usuaris i serveis que s'hagin publicat de prova i lògicament la idea és que es mantinguin tots a partir d'ara.

Importar dades des d'un fitxer extern

Per a treballar amb les localitzacions s'ha importat les dades dels districtes de Barcelona¹² a partir d'un fitxer en format JSON que conté objectes basats en polígons de coordenades.

Per a realitzar la importació, cal escriure la següent comanda a la terminal, indicant el nom de la base de dades de destí (**faenio-dev**), la ruta física del fitxer dins de l'equip (**C:\Users\Nuka\Desktop\barcelona.json**) i el nom de la col·lecció generada (**locations**):

```
$ mongoimport -d faenio-dev  
C:\Users\Nuka\Desktop\barcelona.json -c locations
```

Exemple de la importació en local

Com a nota addicional, posteriorment per tal de tenir la possibilitat de realitzar cerques per localització, s'han creat índexs a la base de dades tant en el model a cercar (Post) com el de les localitzacions (Location). Per a fer-ho cal afegir les següents dues comandes a la consola:

```
db.locations.createIndex({ geometry: "2dsphere" })  
db.posts.createIndex({ location: "2dsphere" })
```

¹² Dades obtingudes de HoodsProject <https://github.com/96Levels/European-Neighborhoods-Json-Coords>

12. Casos d'us

12.1 Diagrama UML de casos d'us

El següent diagrama mostra gràficament la interacció de l'usuari amb el sistema:

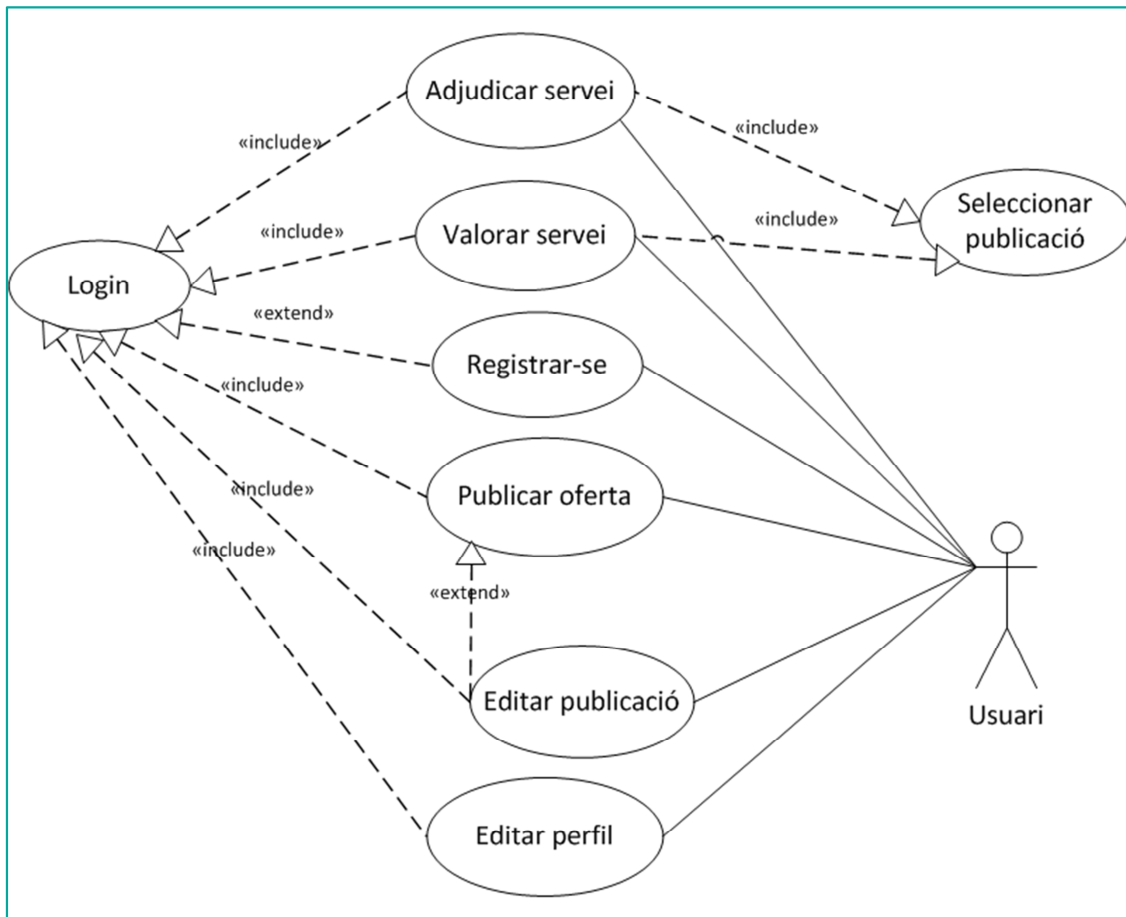


Figura 6. Diagrama UML de casos d'us

Qualsevol usuari de l'aplicació pot publicar una oferta de servei o pot demandar els serveis d'un altre usuari. A l'aplicació l'usuari pot ser alhora client o proveïdor, i per tant s'ha utilitzat un únic actor per a representar-lo. Els casos d'us es detallen en el següent apartat.

12.2 Descripció dels casos d'us

Detall dels casos d'us mostrats en el diagrama anterior amb la seqüència d'interaccions entre l'usuari i l'aplicació.

Per a aclarir la terminologia emprada als casos d'us, es defineixen dos tipus d'usuari:

- Oferent: Aquell usuari que ofereix els seus serveis (proveïdor)
- Demandant: Aquell usuari que busca una ajuda especialitzada (client)

Cas d'us: Publicar oferta

Resum de la funcionalitat: Permet a l'usuari publicar un anunci en el que ofereixi els seus serveis a altres usuaris de l'aplicació

Precondicions: L'usuari ha iniciat sessió a l'aplicació

Escenari principal d'èxit:

1. L'usuari prem el botó "Publicar oferta"
2. El sistema mostra una pantalla amb el formulari "Dades de registre de publicació"
L'usuari omple les dades i prem el botó de confirmació
3. El sistema valida les dades introduïdes i les desa a la base de dades

Escenaris alternatius:

3a. Les dades introduïdes no són vàlides per algun dels següents motius:

- El format d'alguna de les dades introduïdes no és l'esperat
- Falta proporcionar alguna dada obligatòria
 - 3a1. El sistema mostra el formulari "Dades de registre de publicació" amb un missatge d'error adequat
 - 3a2. Anar al punt 2

Cas d'us: Adjudicar servei

Resum de la funcionalitat: L'usuari demandant contracta un servei d'un oferent

Precondicions: L'usuari ha iniciat sessió a l'aplicació

Actor principal: Usuari demandant

Actor de suport: Usuari oferent

Escenari principal d'èxit:

1. L'ofertant publica una oferta de serveis (cas d'us publicar oferta)
2. El demandant selecciona un servei (cas d'us inclòs seleccionar publicació)
3. Oferent i demandant arriben a un acord i es realitza el servei
4. El demandant decideix que vol escriure una opinió sobre el servei rebut i la publica al sistema puntuant a l'ofertant, si ho vol (cas d'us valorar servei)

Escenaris alternatius:

2a. El demandant vol més informació abans de realitzar l'adjudicació del servei i envia un missatge a l'ofertant

2a1. La resposta es satisfactòria i continua el cas d'us en el pas 3

2a2. La resposta no es satisfactòria (no s'arriba a un acord) i finalitza el cas d'us

Cas d'us: Seleccionar publicació

Resum de la funcionalitat: Permet a l'usuari visualitzar les publicacions i seleccionar-ne una

Escenari principal d'èxit:

1. El sistema mostra el cercador
2. El demandant realitza la cerca d'especialistes que s'ajustin a la seva necessitat
3. El sistema mostra un llistat de publicacions
4. L'usuari repeteix els passos 2 i 3 fins que decideixi seleccionar una publicació
5. El demandant troba una oferta que podria ajustar-se al que busca i hi accedeix per veure'n el contingut complert
6. El sistema mostra la pàgina de detall de la publicació

Escenaris alternatius:

2a. El demandant no troba cap oferta que s'ajusti a la seva cerca

- 2a1. El demandant crea una alerta al sistema per tal de rebre una notificació en el cas que un oferent publiqui una oferta que coincideixi amb els seus interessos

Cas d'us: Valorar servei

Resum de la funcionalitat: Un cop realitzat el servei, l'usuari demandant pot deixar una opinió publicada així com també puntuar a l'oferent

Precondicions: L'usuari ha iniciat sessió a l'aplicació

Escenari principal d'èxit:

1. L'usuari demandant prem la opció "El meu perfil" del menú d'usuari
2. El sistema mostra l'historial de serveis realitzats com a oferent i com a demandant
3. L'usuari selecciona el servei que vol valorar
4. L'usuari prem el botó "Valorar" i emet una puntuació de l'1 al 5. Opcionalment pot deixar també un comentari
5. L'usuari confirma la seva valoració i queda enregistrada al sistema

Escenaris alternatius:

2a. L'usuari no ha realitzat ni rebut cap servei

- 2a1. El sistema mostra un llistat amb zero resultats
- 2a2. Finalitza el cas d'us

3a. L'usuari ja ha valorat el servei prèviament

- 3a1. El sistema no mostra el botó "Valorar" perquè només es pot emetre una valoració per a cada servei rebut
- 3a2. Finalitza el cas d'us

Cas d'us: Login**Resum de la funcionalitat:** Permet a l'usuari identificar-se al sistema**Escenari principal d'èxit:**

1. El sistema mostra la pantalla de "Login"
2. L'usuari introdueix les dades d'identificació (totes són obligatòries) o prem un botó per a identificar-se mitjançant les xarxes socials habilitades
3. El sistema comprova que les dades d'identificació són correctes i enregistra la identificació de l'usuari a la seva sessió

Escenaris alternatius:

2a. L'usuari selecciona "Cancel·lar"

2a1. El cas d'us finalitza

3a. Les dades proporcionades per l'usuari no són correctes o hi ha algun problema durant la identificació amb la xarxa social seleccionada

3a1. El sistema mostra la pantalla "Login" amb un missatge d'error adequat

3a2. Anar al punt 2

Cas d'us: Registrar-se**Resum de la funcionalitat:** Permet a l'usuari enregistrar-se al sistema per a poder fer servir altres casos d'us**Casos d'us relacionats:** Estén el cas d'us "Login" en el pas 2**Escenari principal d'èxit:**

1. El sistema mostra a l'usuari el formulari "Dades de registre d'usuari"
2. L'usuari introdueix les dades sol·licitades (totes obligatòries) o prem un botó per a identificar-se mitjançant les xarxes socials habilitades
3. El sistema comprova que les dades són correctes i enregistra l'usuari

Escenaris alternatius:

2a. L'usuari selecciona "Cancel·lar"

2a1. El cas d'us finalitza

3a. Hi ha algun problema durant la identificació amb la xarxa social seleccionada o les dades proporcionades per l'usuari no són correctes per algun d'aquests errors:

- El mot de pas i la seva confirmació no coincideixen
- El nom d'usuari és d'un usuari existent

3a1. El sistema mostra el formulari "Dades de registre d'usuari" amb un missatge

d'error adequat

3a2. Anar al punt 2

Cas d'us: Editar publicació

Resum de la funcionalitat: Permet a l'usuari modificar un servei publicat per ell mateix

Precondicions: L'usuari ha iniciat sessió a l'aplicació

Escenari principal d'èxit:

1. L'usuari accedeix al seu perfil
2. El sistema mostra la seva informació de perfil i un llistat de les seves publicacions
3. L'usuari en selecciona una i prem el botó "Editar"
4. Estén el cas d'us Publicar Oferta en el pas 2

Escenaris alternatius:

2a. L'usuari no té cap servei publicat

2a1. Finalitza el cas d'us

Cas d'us: Editar perfil

Resum de la funcionalitat: Permet a l'usuari modificar la informació del seu perfil que es mostrarà públicament a la resta d'usuaris

Precondicions: L'usuari ha iniciat sessió a l'aplicació

Escenari principal d'èxit:

1. L'usuari accedeix al seu perfil
2. El sistema mostra la seva informació de perfil
3. L'usuari prem el botó "Editar perfil"
4. El sistema mostra el formulari "Dades de perfil"
5. L'usuari omple les dades i prem el botó "Desar" per a emmagatzemar els canvis
6. El sistema enregistra les modificacions

Escenaris alternatius:

5a. Les dades introduïdes no són vàlides per algun dels següents motius:

- El format d'alguna de les dades introduïdes no és l'esperat
- Falta proporcionar alguna dada obligatòria

5a1. El sistema mostra el formulari "Dades de perfil" amb un missatge d'error adequat

5a2. Anar al punt 4

13. Prototips

13.1 Lo-Fi

A continuació es mostren els *wireframes* de les pantalles principals del projecte en versió escriptori i en versió mòbil. Únicament s'han dissenyat les que s'han considerat més importants, és a dir la pàgina d'inici, el llistat de resultats, la pàgina de detall d'una publicació i la pàgina de perfil de l'usuari. A partir d'aquestes i seguint la mateixa línia es desenvoluparan les següents (edició de perfil, edició d'anunci, *login*, sobre nosaltres...) ja directament en l'etapa de maquetació.

Versió per a escriptori

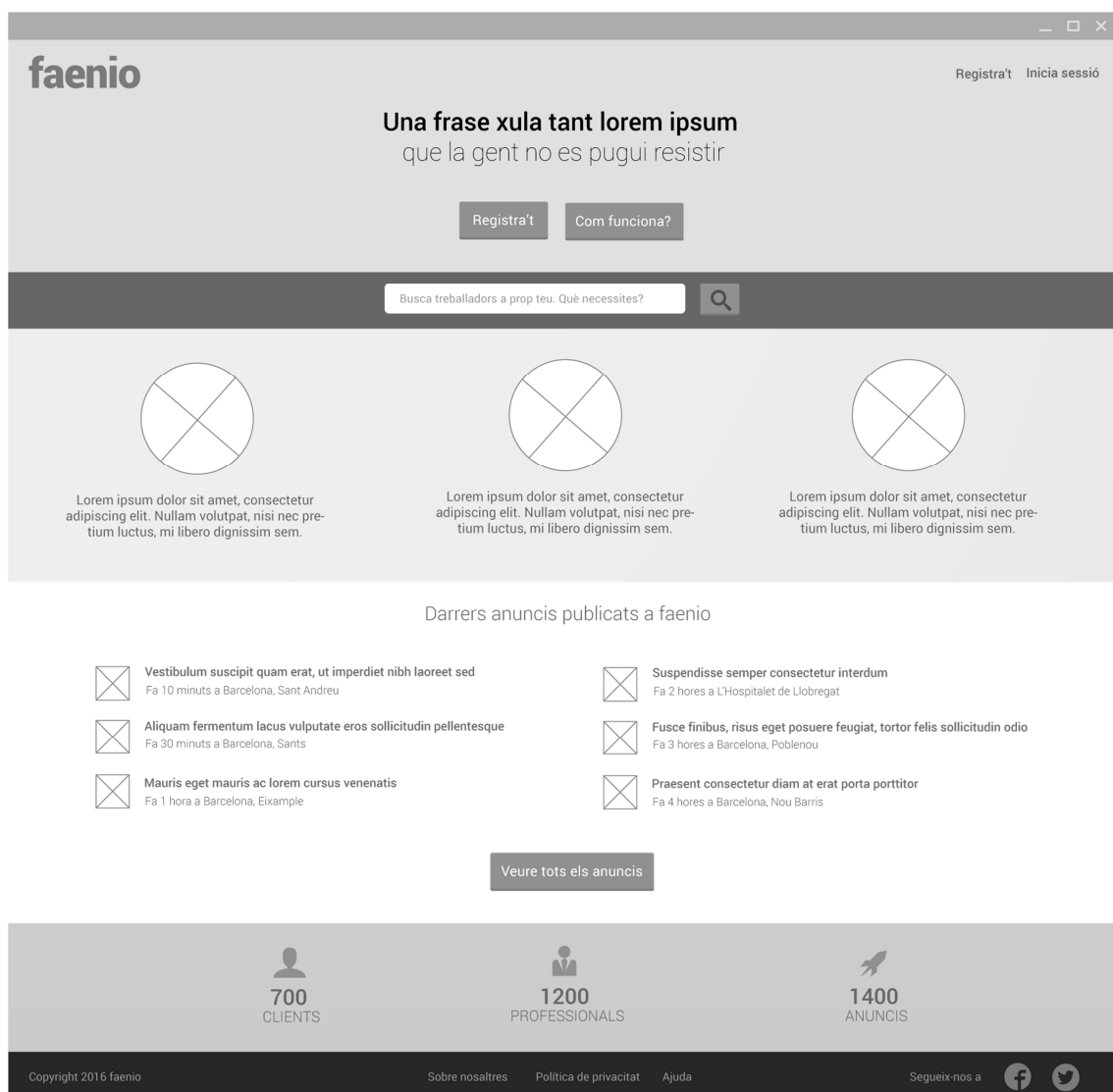


Figura 7. Wireframe pàgina inici escriptori

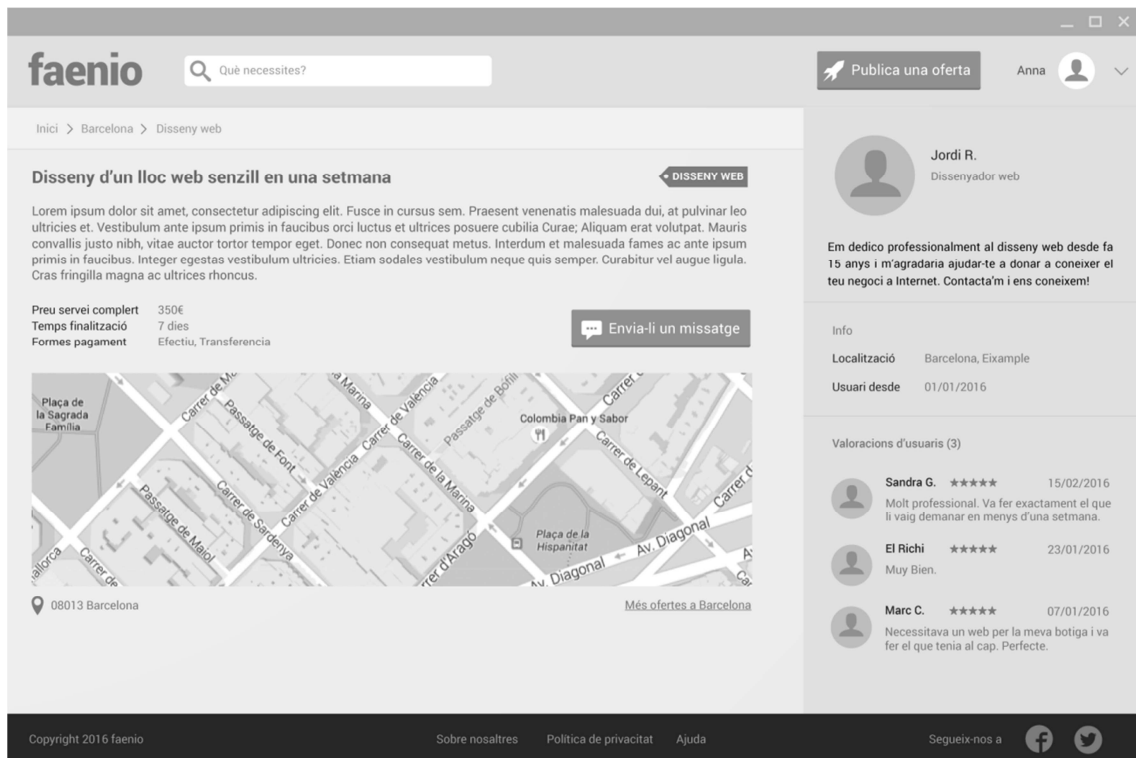


Figura 8. Wireframe pàgina detall escriptori

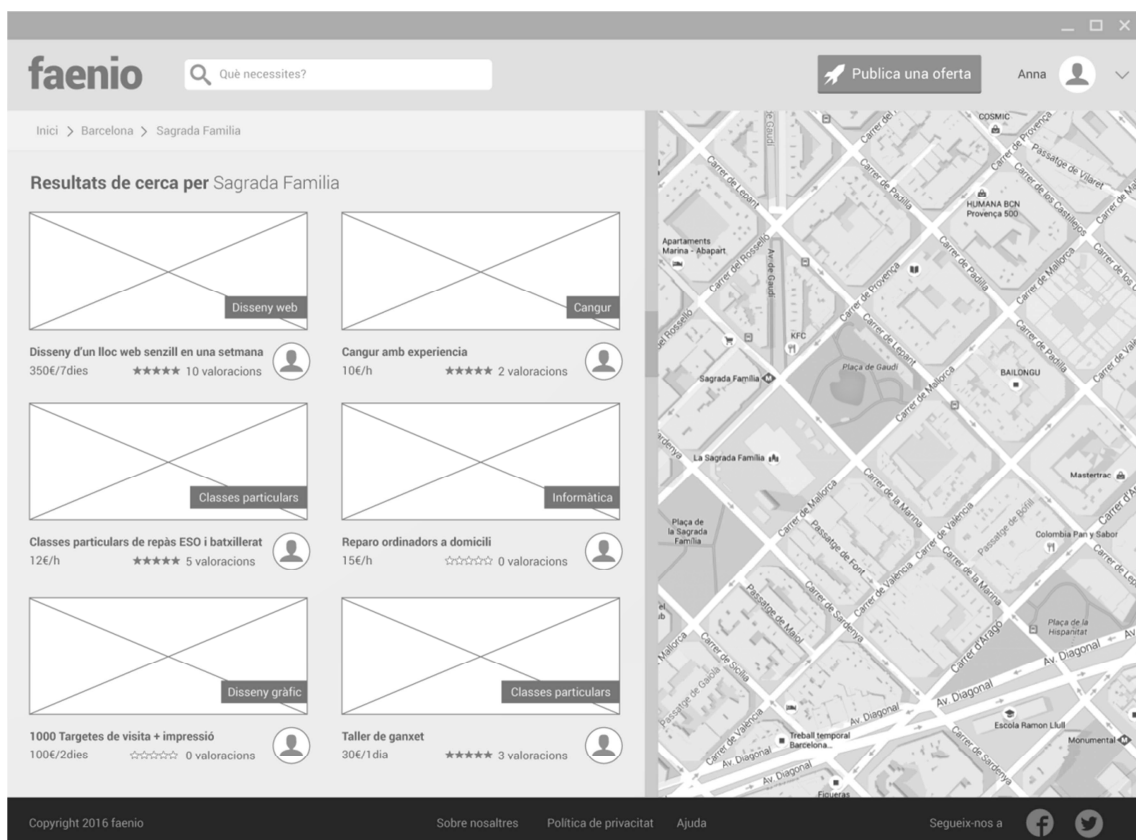


Figura 9. Wireframe pàgina llistat escriptori

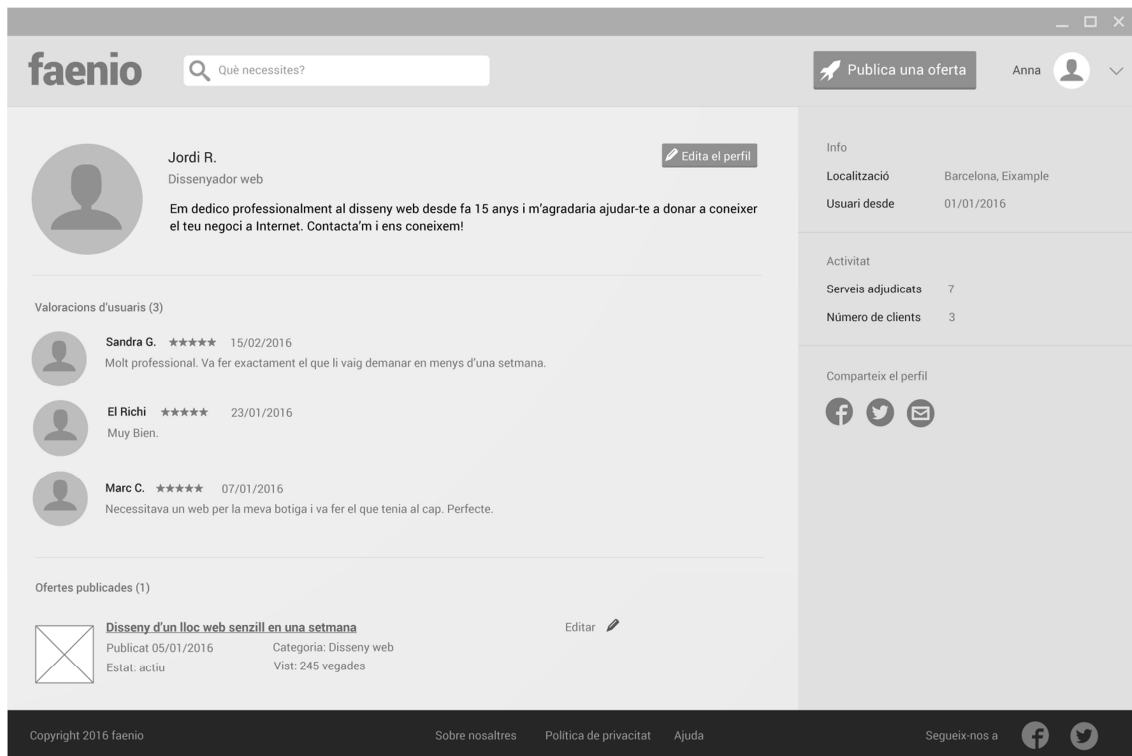


Figura 10. Wireframe pàgina perfil escriptori



Figura 11. Detall del menú d'usuari obert

Versió per a mòbil

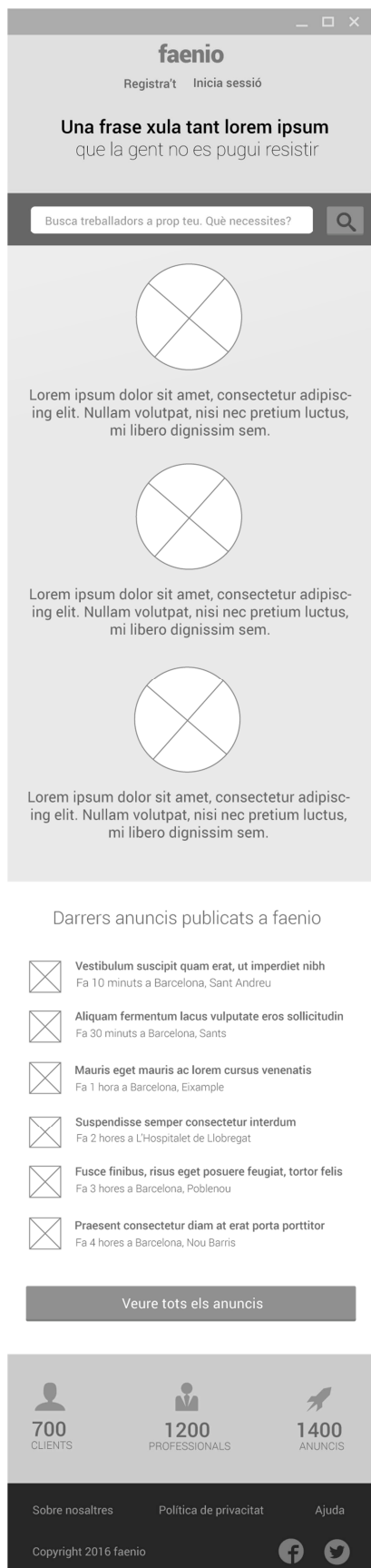


Figura 12. Wireframe pàgina inici mòbil

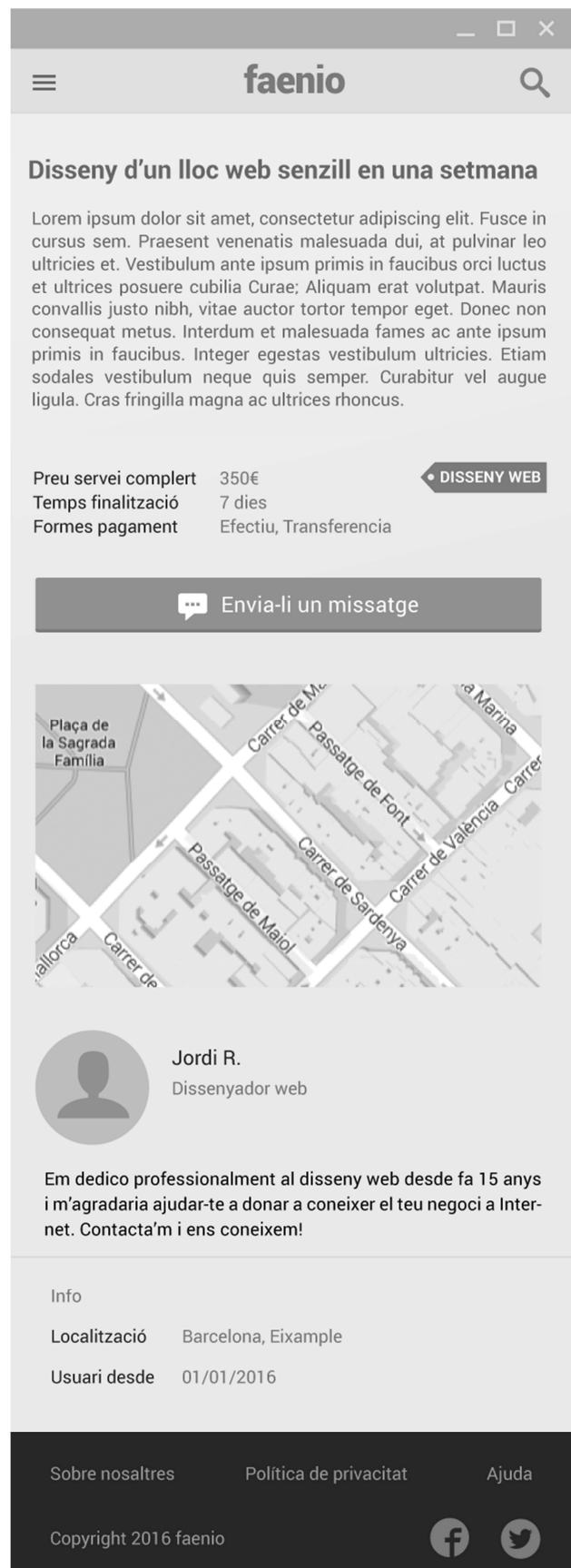


Figura 13. Wireframe pàgina detall mòbil

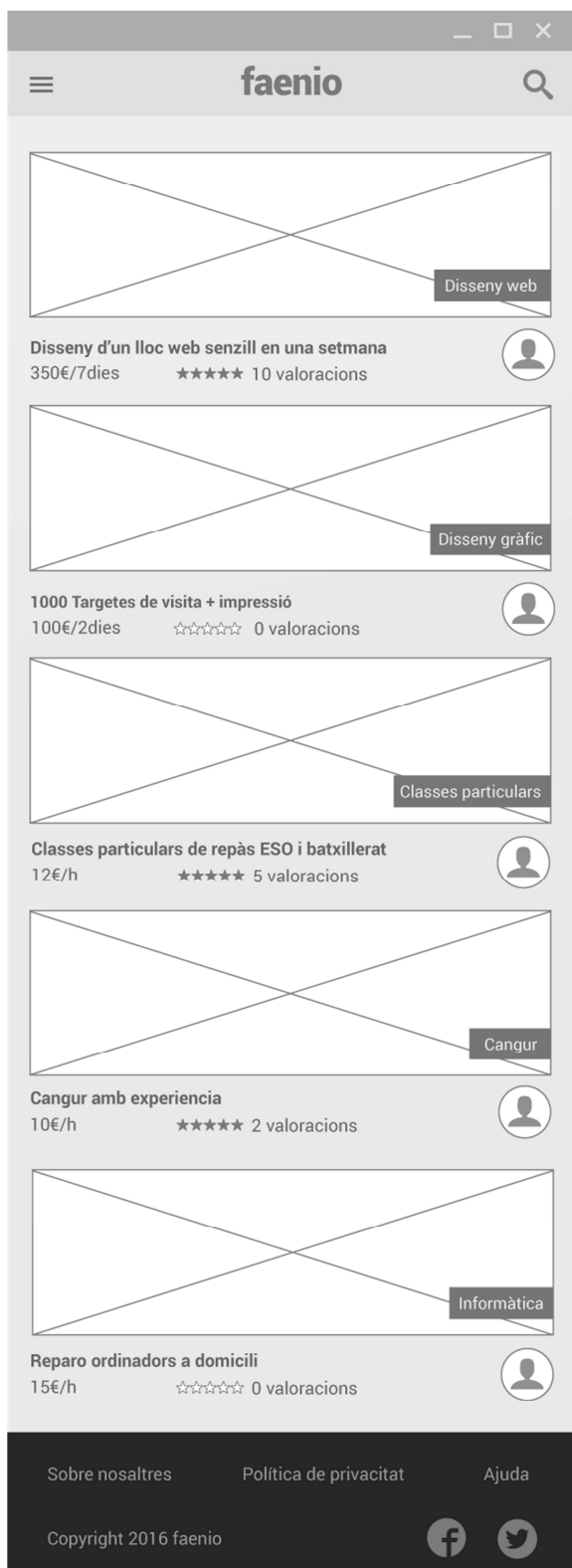


Figura 14. Wireframe pàgina resultats mòbil



Figura 15. Wireframe pàgina perfil mòbil

13.2 Hi-Fi

Els *wireframes* anteriors s'han evolucionat fins a obtenir prototips d'alta definició de les pantalles principals del projecte en versió escriptori i en versió mòbil.

En les pàgines de llistat i d'anunci s'estudiarà la possibilitat d'afegir una imatge personalitzada proporcionada per l'usuari que publiqui la oferta, però en tot cas això serà en una versió de l'aplicació posterior al lliurament final. En la versió actual es mostra una il·lustració iconogràfica relacionada amb la categoria a la qual pertany la publicació.

Quan es desenvolupi la opció d'afegir una imatge, aquesta es mostrarà en la pàgina de detall i en el llistat, mentre que els anuncis sense imatge seguiran mostrant per defecte aquesta il·lustració iconogràfica.

Versió per a escriptori

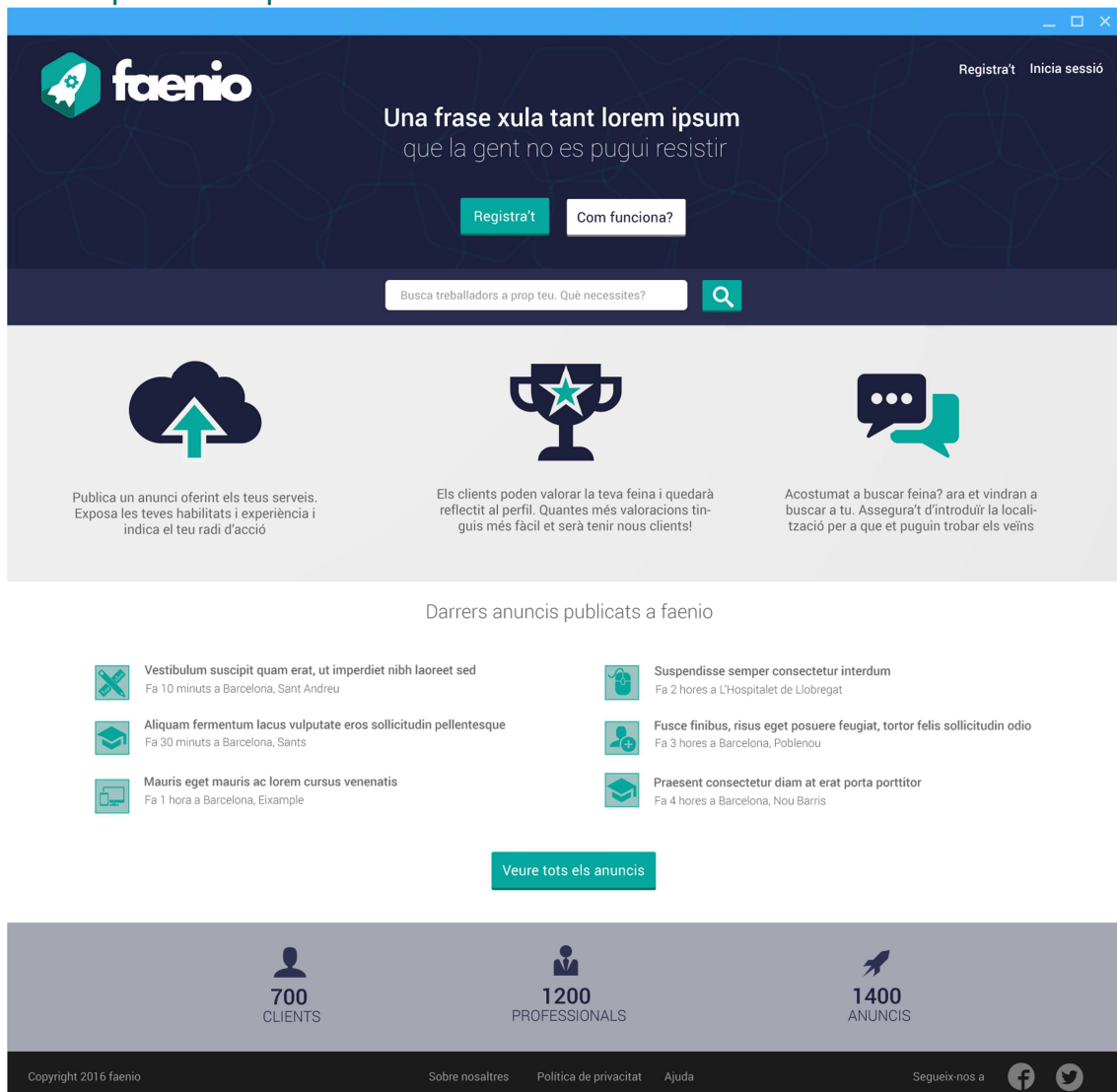


Figura 16. Mockup pàgina inici escriptori

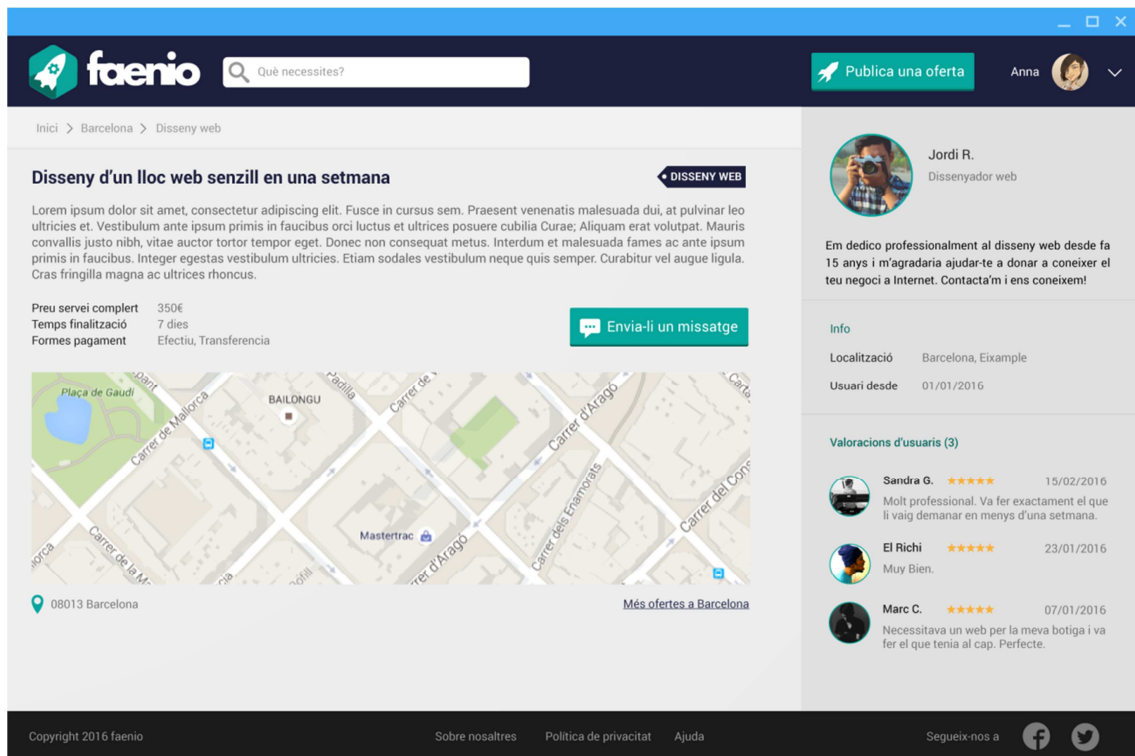


Figura 17. Mockup pàgina detall escriptori

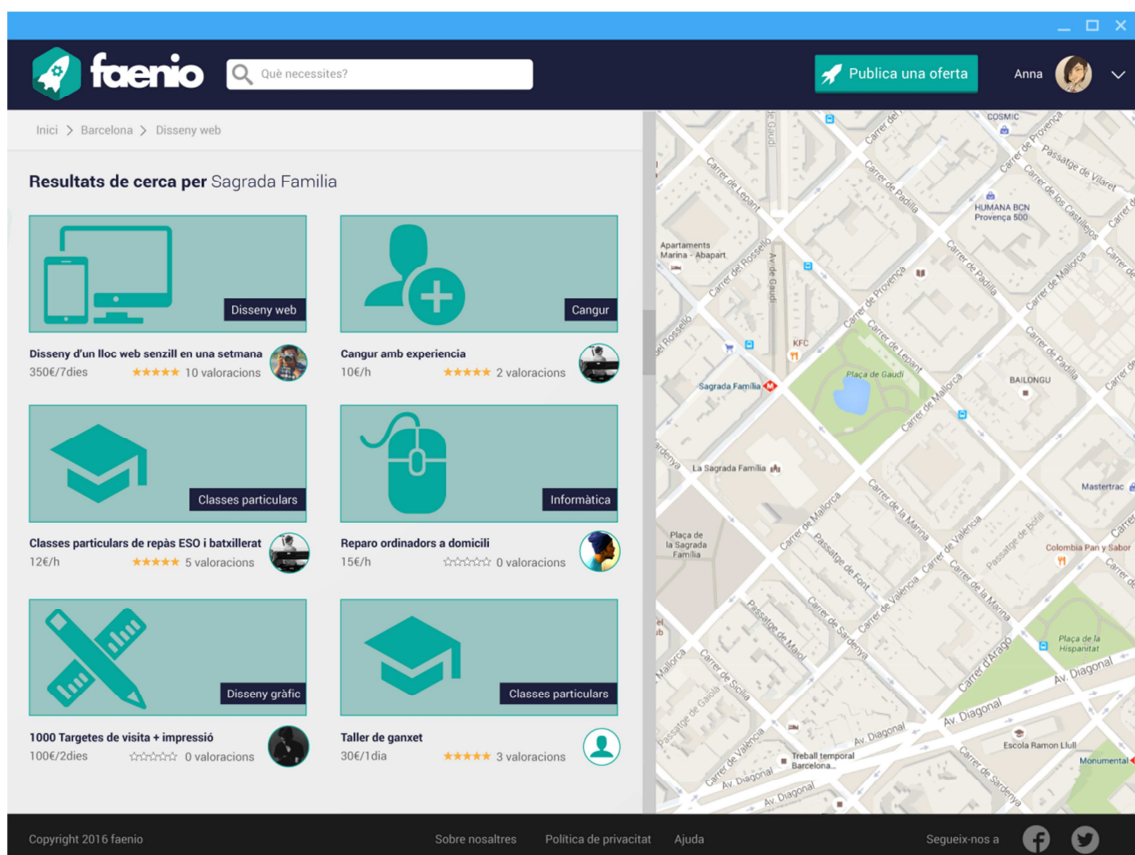


Figura 18. Mockup pàgina llistat escriptori

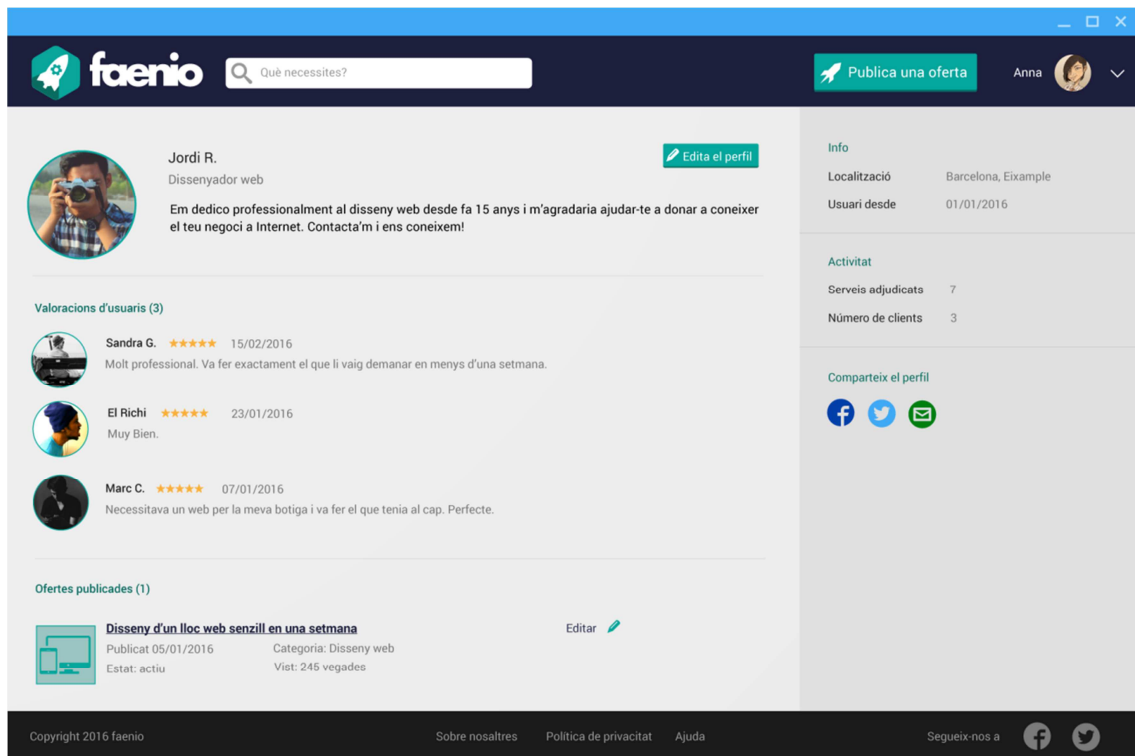


Figura 19. Mockup pàgina perfil escriptori

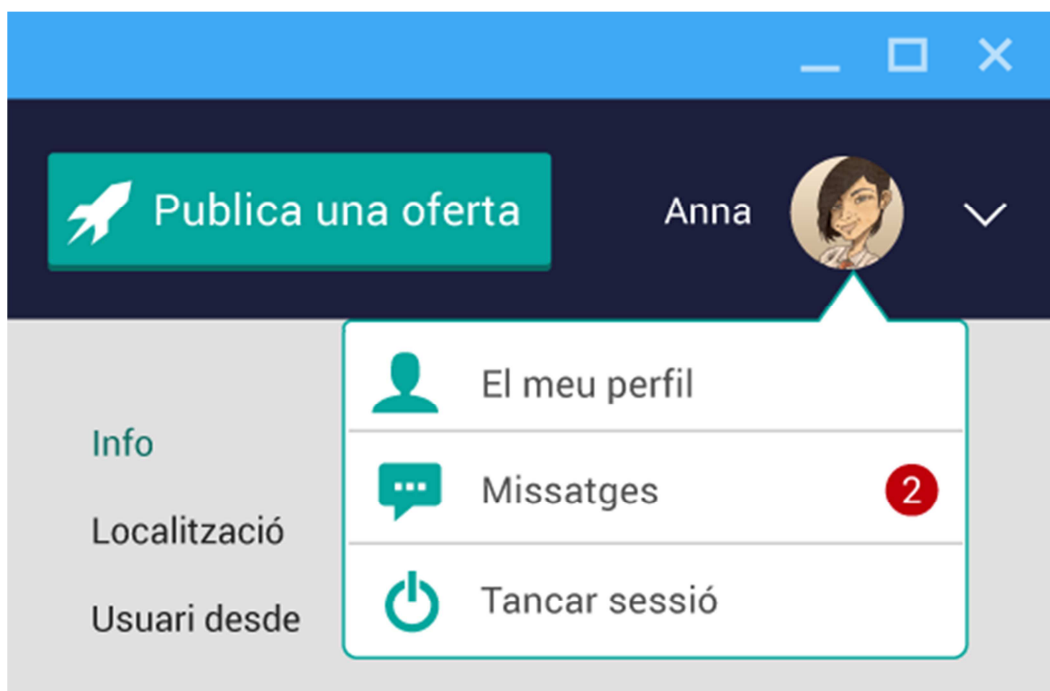


Figura 20. Detall del menú d'usuari obert

Versió per a mòbil



Figura 21. Mockup pàgina inici mòbil

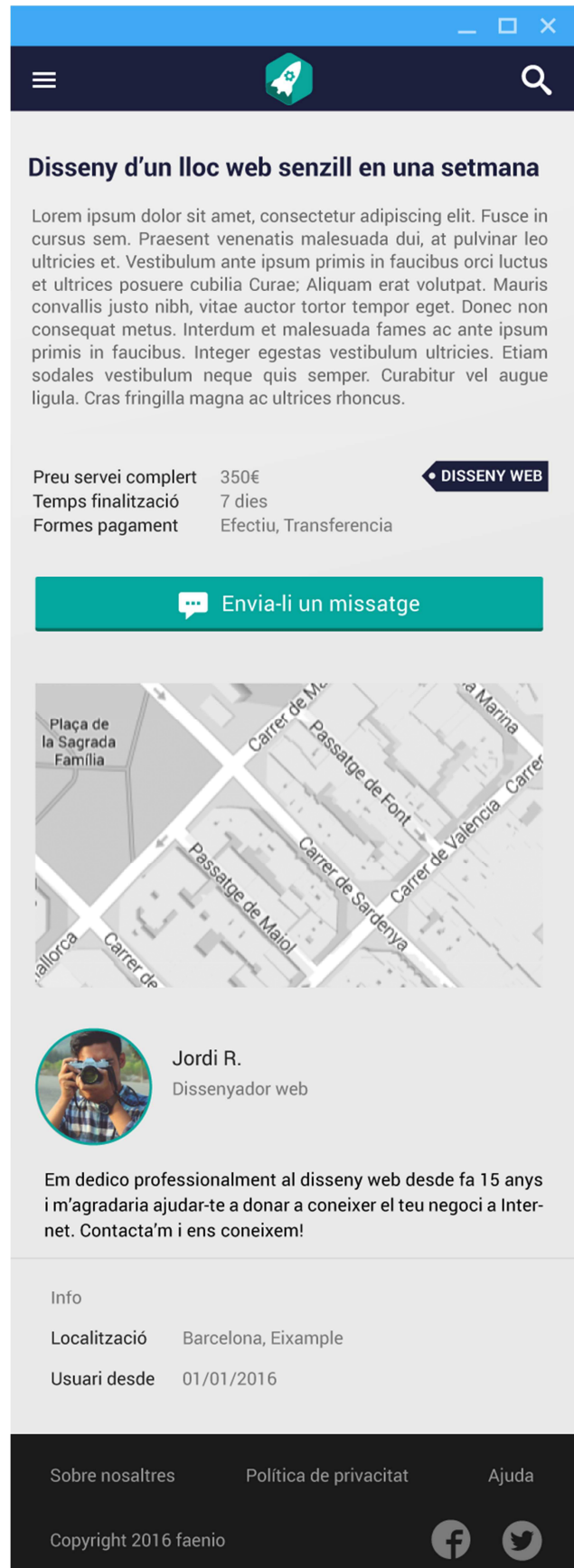


Figura 22. Mockup pàgina detall mòbil



Figura 23. Mockup pàgina llistat mòbil

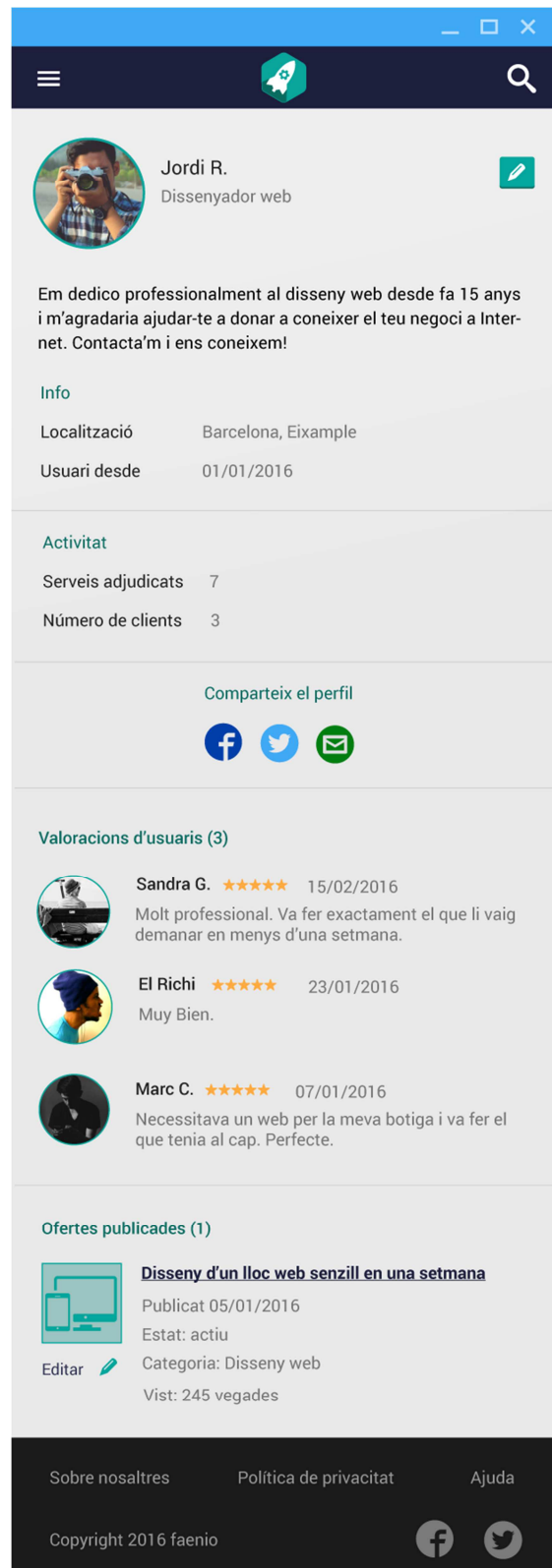


Figura 24. Mockup pàgina perfil mòbil

14. Perfils d'usuari

A la plataforma hi ha dos tipus d'usuaris: els que volen oferir els seus serveis i els que necessiten el servei de tercers. No obstant, a la pràctica l'aplicació no fa distinció dels dos rols i un usuari pot actuar indistintament com a proveïdor de serveis com a consumidor.

14.1 Usuaris proveïdors de serveis

L'usuari proveïdor de serveis és aquell que té una habilitat que vol explotar comercialment dins la plataforma. Aquest usuari es servirà de l'aplicació per a obtindrà clients i, consegüentment, augmentar els seus ingressos.

La interacció d'aquests usuaris amb la plataforma serà principalment la publicació d'anuncis oferint els seus serveis, i contestant els missatges dels usuaris que estiguin interessats en tenir més informació abans de realitzar l'adjudicació del servei. És a dir, la seva funció és la de generar el contingut de l'aplicació i per tant fonamenta la base de l'aplicació.

Aquests usuaris procuraran que tant la seva informació de perfil com la informació dels seus anuncis sigui prou detallada com per a generar confiança en els potencials clients.

Les característiques d'un proveïdor de serveis són les d'un usuari adaptat a les noves tecnologies que vol oferir les seves habilitats per a guanyar uns diners extra. Poden tenir diferents procedències, com persones a l'atur, persones que volen guanyar uns diners apart del sou, jubilats amb energia i temps lliure, joves que no han trobat la primera feina, etc.

14.2 Usuaris consumidors de serveis

La plataforma es completa amb aquest segon perfil d'usuari, que serà consumidor de la informació i alhora client.

La interacció d'aquests usuaris amb la plataforma és la de cercar informació, adjudicar-se serveis i realitzar valoracions d'usuaris proveïdor de manera que serveixi de guia per a altres usuaris consumidors. També contactaran amb proveïdors de serveis per a tal d'obtenir més informació abans d'adjudicar un servei.

Les característiques d'un consumidor són les d'un usuari adaptat a les noves tecnologies amb la necessitat d'encarregar una tasca o servei a una tercera persona i disposat a buscar una persona geogràficament propera que realitzi la tasca o servei a un preu que consideri adequat al seu pressupost.

15. Històries d'usuari

A continuació es mostra un recull d'històries d'usuari de l'aplicació, amb diversos criteris d'acceptació a sota de cadascuna d'elles.

Com a usuari consumidor vull tenir la possibilitat de contactar amb el proveïdor del servei abans de formalitzar una transacció

- Verificar que només es pot enviar un missatge si l'usuari està identificat al sistema
- Verificar que el receptor realment rep el missatge escrit per l'emissor
- Verificar que apareix un avís de missatge nou cada cop que n'arriba un

Com a usuari consumidor vull poder consultar les opinions d'altres usuaris sobre un servei determinat per tal d'escollir el perfil final

- Verificar que es mostren les valoracions només de l'usuari que s'ha seleccionat
- Verificar que les valoracions es publiquen correctament

Com a estudiant vull accedir a l'aplicació mitjançant Twitter o Facebook per no haver de recordar sempre l'usuari i contrasenya

- Verificar el funcionament de la llibreria que permet accedir al sistema mitjançant aquestes xarxes socials
- En cas d'actualitzacions de la llibreria verificar que segueix funcionant

Com a estudiant vull filtrar per categoria per trobar classes de reforç

- Verificar que el sistema de cerca busca correctament per categoria
- Verificar que les categories existents encaixin dins les necessitats dels usuaris que utilitzaran l'aplicació
- Verificar que els resultats de cerca s'ordena per localització geogràfica (de més proper a més lluny)

Com a aturat vull publicar anuncis per guanyar uns diners extres

- Verificar que quan es faci clic a "Acceptar o Desar" es registri la sol·licitud al sistema i es faci visible per a la resta d'usuaris
- Verificar que quan es faci clic a "Acceptar o Desar" el formulari quedi bloquejat per a la seva edició
- Verificar que hi ha una validació dels camps i que tots els obligatoris s'han introduït

Com a aturat vull saber com funciona l'eina abans de començar a utilitzar-la

- Verificar que existeixen les pàgines d'ajuda amb la informació de l'aplicació
- Verificar que existeix una via de contacte per a que els usuaris puguin demanar més informació
- Verificar que els continguts de la documentació d'ajuda responen a tot el que un usuari mitjà pot esperar, per tal d'evitar respondre contínuament consultes

Com a autònom vull indicar el meu preu hora i termini de finalització perquè el "client" sàpiga el que haurà de pagar pels meus serveis

- Verificar que el camp s'ha omplert i que s'ha validat com a nombre real
- Verificar que la informació es mostra correctament a la fitxa de la publicació

Com a autònom vull poder seleccionar diversos tipus de pagament perquè com més facilitats li doni al comprador, més interès tindrà en que treballi per ell

- Verificar que es poden registrar diversos mètodes de pagament
- Verificar que els mètodes de pagament es mostren correctament a la fitxa

Com a jubilat vull trobar a persones a prop que em puguin acompanyar al metge sense haver d'escriure gaire perquè em sigui més fàcil

- Verificar l'accessibilitat de l'eina

Com a jubilat vull poder accedir a l'eina sense haver de donar-me d'alta a cap Facebook ni Twitter perquè és massa modern i complicat

- Verificar que la llibreria que permet accedir al sistema funciona correctament mitjançant usuari i contrasenya
- Verificar la funcionalitat de "recordar contrasenya" o "restablir contrasenya"

16. Usabilitat/UX

La creació de la interfície gràfica i el disseny s'ha realitzat tenint sempre present l'usuari que utilitzarà l'aplicació. El Disseny Centrat en l'Usuari (DCU) és una filosofia de disseny que situa a l'usuari en el centre del procés de creació per aconseguir un producte final amb la funcionalitat adequada pels perfils d'usuari que s'han definit, adaptant la tecnologia utilitzada a les seves expectatives i creant interfícies que facilitin l'assoliment dels seus objectius.

El procés que s'ha seguit és el que defineix la norma ISO 13407¹³ i que es pot desgranar en quatre fases:

- Especificar el context d'us: Identificar a les persones a les que es dirigeix el producte, per a què l'utilitzaran i en quines condicions
- Especificar els requisits: Identificar els objectius de l'usuari del producte
- Produir solucions de disseny
- Avaluació: Validació de les solucions de disseny o detecció de problemes d'usabilitat mitjançant tests amb usuaris

Per a avaluar la usabilitat s'han utilitzat el conjunt d'heurístics¹⁴ proposats per *Jakob Nielsen*, que són els següents:

Visibilitat de l'estat del sistema

El lloc web o aplicació ha de mantenir sempre informat l'usuari del que està ocorrent i proporcionar-li una resposta en un temps raonable.

En les crides asíncrones (és a dir, cada cop que s'executi un procés intern arrel d'una acció de l'usuari) sempre es mostra una capa i una icona de "Carregant" per a que l'usuari sàpiga que realment la pàgina està processant l'acció que ha demanat. I en cas de que es produeixi un error, apareixerà un missatge per pantalla.

¹³ **ISO 13407:1999** Human-centered design processes for interactive systems:
http://www.iso.org/iso/catalogue_detail.htm?csnumber=21197

¹⁴ L'**avaluació heurística** és una variant de la inspecció d'usabilitat on els especialistes en usabilitat jutgen si cada element de la interfície d'usuari segueix els principis d'usabilitat establerts. Aquest mètode forma part del que es coneix com "enginyeria de la usabilitat rebaixada"

Adequació entre el sistema i el món real

El lloc web o aplicació ha d'utilitzar el llenguatge de l'usuari, amb expressions i paraules que li resultin familiars. La informació ha d'aparèixer en un ordre lògic i natural.

S'utilitza el llenguatge de l'usuari, sense nomenclatures tècniques per a evitar confusions. Els textos explicatius, els missatges d'error i els missatges d'alerta i, en definitiva, qualsevol comunicació des de l'aplicació a l'usuari es realitza en un llenguatge entenedor i alhora proper (en segona persona del singular), per a fer una interacció i un diàleg més càlids.

Llibertat i control per part de l'usuari

En cas d'elegir alguna opció del lloc web o aplicació per error, l'usuari ha de disposar d'una "sortida d'emergència" clarament delimitada per a abandonar l'estat no volgut en què es troba sense haver de mantenir un diàleg llarg amb el lloc o aplicació.

Difícilment l'usuari podrà equivocar-se, però en el cas que això ocorri només cal que canviï a qualsevol altra pàgina de l'aplicació.

A la capçalera de la pàgina sempre s'hi mostra el logotip amb la opció de clicar-lo i tornar a la pàgina principal. D'aquesta manera l'usuari no es pot perdre mai.

Consistència i estàndards

Els usuaris no cal que sàpiguen que diferents paraules, situacions o accions signifiquen el mateix. És convenient seguir convencions.

S'utilitza un llenguatge consistent al llarg de l'aplicació per tal de no enviar un missatge confús. Així per exemple si en una pàgina es demana "iniciar sessió", des de qualsevol lloc on es faci referència a aquesta opció s'utilitzarà la mateixa nomenclatura, sense variacions com "login" o "accedir", etc.

Per altra banda, els elements que suposin una interacció (enllaços, botons, elements de formulari) estan dissenyats de manera que l'usuari sàpiga que pot interactuar amb ells.

Prevenició d'errors

És important prevenir l'existència d'errors mitjançant un disseny adequat. Tot i així, els missatges d'error han d'incloure una confirmació abans d'executar les accions de correcció.

Abans d'eliminar un element es demana una confirmació per pantalla, per tal d'assegurar que no s'ha cridat la opció de manera accidental i que l'usuari és conscient de l'acció que realitzarà. El mateix ocorrerà quan un usuari decideixi desactivar un anunci, indicant-li que a partir d'aquell moment altres usuaris no podran visualitzar aquella publicació (tot i que de moment la possibilitat de desactivar anuncis no està desenvolupada).

Reconeixement abans que record

Fer visibles objectes, accions i opcions perquè l'usuari calgui que recordi informació entre diferents seccions o parts del lloc web o aplicació. Les instruccions d'ús han de ser visibles o fàcilment localitzables.

Es tracta d'una aplicació amb un nombre reduït de pàgines i per tant s'ha plantejat una navegació directa, de manera que des del menú principal, amb un sol nivell de navegació, es pugui accedir a la resta de seccions. En tot moment usuari té les seccions a l'abast.

Flexibilitat i eficiència en l'ús

Els acceleradors o dreceres de teclat poden fer més ràpida la interacció per a usuaris experts, de tal manera que el lloc web o aplicació sigui útil tant per a usuaris novells com avançats.

L'aplicació està maquetada tenint en compte la guia d'accessibilitat promoguda per l'organització W3C¹⁵, en la que una de les seves pautes indica que les funcionalitat de la pàgina han de ser accessibles des del teclat. Per això en els enllaços del menú s'hi ha configurat una drecera mitjançant l'atribut "accesskey". Exemple:

```
<a href="/sobre-nosaltres/" accesskey="s" title=" Sobre nosaltres">Sobre nosaltres</a>
```

Disseny estètic i minimalista

Les pàgines no han de contenir informació irrellevant o innecessària. Cada informació extra competeix amb la informació rellevant i en disminueix la visibilitat.

L'arquitectura de la informació s'ha basat en el paradigma *Mobile First*, fet que obliga a prescindir de continguts innecessaris per a centrar-se únicament en el més important. Per altra

¹⁵ W3C <https://www.w3.org/>

banda s'ha dissenyat l'aplicació fent una barreja de *flat design*¹⁶ i *material design*¹⁷, ambdós tendències basades en una conceptualització neta i minimalista.

Ajuda els usuaris a reconèixer i diagnosticar els errors i recuperar-se'n

Els missatges d'error s'han d'expressar en un llenguatge comú i senzill, que indiqui amb precisió el problema i suggereixi les possibles alternatives o solucions.

Els processos que s'executen al sistema retornen errors amb "llenguatge humà" per tal que l'usuari no es trobi desemparat davant d'errors del tipus "Error #23974-09".

S'han afegit validacions en els formularis d'introducció de dades (inici de sessió, registre d'usuari, edició de perfil i creació/edició de serveis) que mostren missatges d'error de manera amigable i entenedora.

Ajuda i documentació

Encara que és millor que el lloc web o aplicació pugui ser usat sense documentació, pot ser necessari proveir cert tipus d'ajuda. En aquest cas, l'ajuda ha de ser fàcil de localitzar, ha d'especificar els passos necessaris i no ha de ser gaire extensa.

Tot i que l'aplicació esta pensada per a poder utilitzar-se de manera senzilla, existeix una pàgina d'ajuda amb la documentació necessària per a què l'usuari no es senti desemparat. S'hi pot accedir des d'un botó molt visible a una zona destacada de la pàgina d'inici i al peu de totes les pàgines. De moment el contingut de la pàgina és fictici (obtingut d'un generador de "Lorem ipsum"), però de cara a l'aplicació es donarà per fet que és contingut d'ajuda real.

¹⁶ Flat design: <http://www.silocreativo.com/2014/01/flat-design-menos-es-mas-diseno-web/>

¹⁷ Material design: https://es.wikipedia.org/wiki/Material_design

17. Seguretat

Autenticació d'usuaris

Sempre que es tracta amb informació personal dels usuaris cal assegurar-se que s'utilitzen de la manera més segura possible. Per aquesta raó s'ha fet us del mòdul *Passport*, un mòdul de tercers que encripta les claus d'usuari i gestiona l'accés a l'aplicació creant un *token* de seguretat que es guarda en sessió i que s'elimina quan l'usuari es desconnecta o passa massa temps inactiu.

Passport és també qui fa possible que els usuaris puguin registrar-se i iniciar sessió mitjançant Facebook i Twitter, creant un nou usuari que emmagatzema les dades proporcionades per les API d'aquestes aplicacions.

Codi Restrictiu

El llenguatge Javascript ha estat considerat en ocasions com a poc consistent i poc segur a l'hora de programar i per aquesta raó es va proposar que s'haurien d'introduir restriccions que forcessin a que el codi hagués d'acomplir més requeriments per a garantir la seva coherència. Per exemple, és habitual considerar com a mala pràctica el fet de no declarar variables abans de ser utilitzades perquè pot induir a errors.

La directiva *"use strict"*, utilitzada en l'aplicació **faenio**, indica el mode en que el navegador ha d'executar el codi Javascript i com aquest admet o no admet certes convencions. *Strict mode*, com s'anomena aquest mode del navegador, tracta de fer més segura i fiable la programació amb Javascript i elimina alguna de les "parts dolentes" del llenguatge.

18. Instruccions d'instal·lació

Per a desplegar l'aplicació al servidor remot de Heroku cal tenir localment instal·lat tant git com Heroku Toolbelt¹⁸, una interfície de línia de comandes que permet interactuar amb l'allotjament web.

Un cop s'ha instal·lat el *toolbelt* s'ha d'introduir les credencials d'usuari:

```
Nuka@FORSYTHIA7 D:\Webs\faenio
$ heroku login
Enter your Heroku credentials.
Email:
Password (typing will be hidden):
```

Si no hi hagués cap repositori creat se n'hauria de fer un de nou, però com que "faenio" ja existeix només cal afegir Heroku com a remot. Per això cal estar al directori de l'aplicació i escriure la comanda:

```
$ heroku git:remote -a faenio
```

I seguidament ja es pot fer un push a Heroku amb la línia següent:

```
$git push heroku master
```

```
Nuka@FORSYTHIA7 D:\
$ cd webs/faenio

Nuka@FORSYTHIA7 D:\Webs\faenio
$ heroku git:remote -a faenio
Heroku CLI submits usage information back to Heroku. If you would like to disable this, set `skip_
heroku/config.json
set git remote heroku to https://git.heroku.com/faenio.git

Nuka@FORSYTHIA7 D:\Webs\faenio
$ git push heroku master
Counting objects: 1264, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (1216/1216), done.
Writing objects: 100% (1264/1264), 2.46 MiB | 71.00 KiB/s, done.
Total 1264 (delta 403), reused 0 (delta 0)
```

Figura 25. Afegir Heroku com a Master i publicar el repositori

¹⁸ Eina Heroku Toolbelt: <https://toolbelt.heroku.com/>

Si tot ha anat bé, els arxius s'hauran publicat al servidor i es mostrarà el missatge “deploy... done” a la línia de comandes com es mostra a la figura 26 confirmant que el procés ha finalitzat.

```
remote: ----> Discovering process types
remote:   Procfile declares types   -> (none)
remote:   Default types for buildpack -> web
remote:
remote: ----> Compressing...
remote:   Done: 17.8M
remote: ----> Launching...
remote:   Released v3
remote:   https://faenio.herokuapp.com/ deployed to Heroku
remote:
remote: Verifying deploy... done.
To https://git.heroku.com/faenio.git
 * [new branch]      master -> master

Nuka@FORSYTHIA7 D:\Webs\faenio
$ |
```

Figura 26. Confirmació de desplegament finalitzat

En aquest moment ja es pot accedir a *faenio.herokuapp.com* per a veure si realment el procés ha funcionat. Pot ser que això no sigui així i el que es mostri sigui la següent finestra d'error:

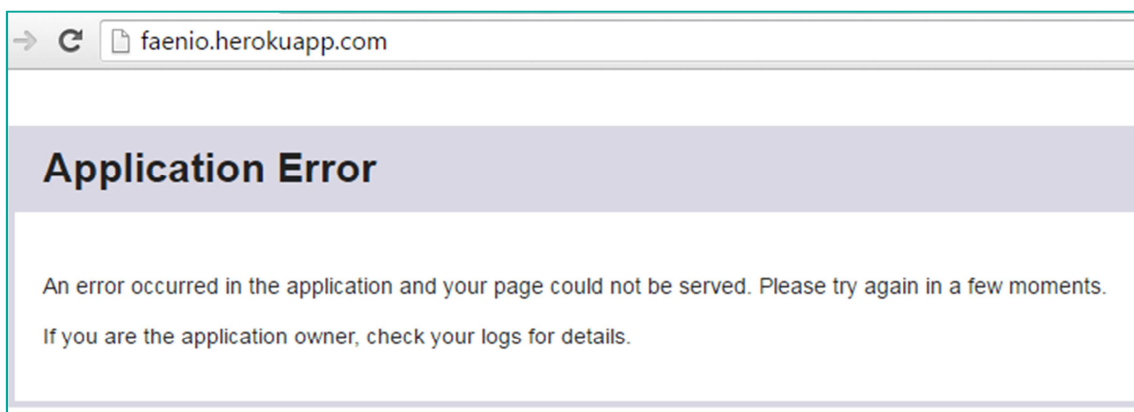


Figura 27. Error a l'executar l'aplicació al servidor

El missatge que es mostra per pantalla es molt poc descriptiu i no dona cap pista sobre quin problema hi ha. Per a solucionar-ho cal accedir al panell de control de Heroku i veure els logs generats en l'execució, que sí que donaran més pistes sobre el problema.

En el meu cas particular hi havia un error en que l'aplicació no trobava la base de dades MongoLab, que no estava correctament especificada en els arxius de configuració.

19. Bugs

A mesura que s'avançava en el desenvolupament de l'aplicació s'han descobert errors i mancances que comprometien el seu correcte funcionament. A continuació es mostren els *bugs*¹⁹ detectats i l'estat en el que es troben en el moment de lliurar el projecte.

Ref.	Descripció	Estat
1	Error a l'iniciar sessió/regar amb Twitter	Resolt
2	Error a l'iniciar sessió/regar amb Facebook	Resolt
3	A vegades al iniciar sessió no es mostra l'avatar i el nom d'usuari fins que es refresca la pàgina o es va a una de nova	Resolt
4	El booleà " <i>isPriceHour</i> " del model Post no s'envia correctament	Resolt
5	L'aplicació envia a l'usuari a les pàgines que requereixen inici de sessió i després mostra la finestra modal, mostrant d'aquesta manera informació que hauria de ser restringida	Resolt
6	Problemes d'adaptació <i>responsive</i> en tota l'aplicació	Parcial
7	Al publicar un nou anunci i anar a qualsevol llistat d'anuncis publicats aquest no apareix si no es refresca la pàgina. Cal trobar la manera de que es refresqui el llistat automàticament. <i>Bug</i> similar a la referència 3	Resolt
8	En ocasions cal refrescar la pàgina perquè es mostri el mapa amb la localització dels serveis	Pendent
9	Al crear un nou xat no es desen els missatges	Resolt
10	El botó "Troba'm" (per detectar la localització actual) no funciona des de dispositius mòbils	Pendent

Taula 3. Bugs

Alhora de prioritzar les activitats s'ha optat per ajornar la resolució d'aquells *bugs* que no s'han considerat imprescindibles per al bon funcionament de l'aplicació i es resoldran en una futura versió posterior a l'entrega del projecte juntament amb aquelles funcionalitats secundaries que, degut al gran abast i l'ajustat temps disposat, no s'han portat a terme. Aquestes són:

- Sistema per a recordar la contrasenya de l'usuari
- Possibilitat que els usuaris puguin afegir la seva imatge de perfil
- Incloure els mètodes de pagament als serveis
- Compartir perfil/publicació per e-mail

¹⁹ Un *bug* és un terme que s'utilitza per a referir-se a errors de programari que desencadenen a un resultat indesitjat

20. Projectió a futur

L'aplicació que s'entrega juntament amb aquesta memòria deixa lloc per a moltes millores i funcionalitats per a fer-la més completa però que en el marc temporal d'aquest projecte no han estat possibles de dur a terme. En cas que es consideri continuar amb aquest projecte en un futur, algunes de les funcionalitats que es preveuen són les següents, per ordre de prioritat:

- Resolució de *bugs* pendents
- Que funcioni per a més ciutats (actualment es troba limitat a Barcelona) mitjançant l'addició de localitzacions quan s'afegeix un nou servei
- Posar un cercador a la capçalera de l'aplicació
- Escriure contingut real per a "Sobre faenio", "Ajuda" i pàgines legals
- Incloure avís de *cookies* al accedir a l'aplicació
- Possibilitat d'afegir imatges personalitzades a les publicacions de serveis com a exemple/mostra del servei que s'està oferint
- Possibilitat d'eliminar el perfil d'usuari
- Millorar els algorismes de cerca per a que es retornin els resultats en menys temps i que, en cas de no trobar coincidències per a paraules clau, ofereixi els resultats més rellevants
- Possibilitat de crear una alerta per a rebre una notificació en el cas que es publiqui una oferta que coincideixi amb els interessos de l'usuari
- Fer que quan s'assigni un servei el client rebi un missatge (via xat) avisant que ja pot emetre la seva valoració
- Desar publicacions en un llistat de "preferits"
- Possibilitat d'activar/desactivar els serveis i que només es mostrin en les llistes de resultats aquells que estiguin marcats com a "actius"
- Afegir una paginació dels resultats
- Distingir entre missatges llegits i no llegits i enviar notificacions de missatges nous
- Afegir la opció de pagament del servei via App (amb la consegüent obtenció de beneficis, destinant un percentatge de la operació per a **faenio**)
- Canviar les rutes (*permalinks*) del contingut d'usuaris i serveis per a que siguin òptimes per al SEO i posicioni millor en els resultats de cercadors
- Aplicació multiidioma amb traduccions al castellà (en una primera fase), anglès (segona fase) i altres a determinar si la acollida amb usuaris reals fos prou bona
- Fer una versió nativa de l'aplicació per a Android i iOS

21. Estudi de viabilitat

El present estudi compren el pressupost, l'anàlisi de mercat i el micro-pla d'empresa i acaba amb unes breus conclusions sobre la viabilitat del projecte.

21. 1. Pressupost

Recursos humans

L'equip humà estaria format bàsicament per tres perfils diferents. Per una banda hi hauria un **gestor de projectes** que s'encarregaria, entre altres tasques, de la definició del projecte, l'estudi de viabilitat, el pressupost, la planificació i la gestió de l'equip.

Per altra banda caldria un **perfil orientat al disseny** que s'encarregaria del *branding*, del disseny de la interfície (prototips de baixa i d'alta definició) i del llibre d'estil. I per últim, caldria un **perfil de full stack developer** (desenvolupador "tot terreny") que s'encarregaria de la part tècnica: definició dels requeriments tècnics i decisions de les tecnologies a utilitzar, desenvolupament del *frontend*, desenvolupament del *backend*, gestió de la base de dades i administració del servidor web.

En el cas d'aquest projecte concret els diferents rols els realitza la mateixa persona i per tant s'ha considerat un sol preu hora independentment del tipus d'activitat a realitzar.

Tal com es senyala en la planificació, el projecte es du a terme en **280 hores** estimades de producció. Aquestes tenen un cost de **25€** bruts per hora, així que el cost total dels recursos humans seria de **7000€** (+IVA).

A aquest cost s'hi hauria de sumar un manteniment per tal d'assegurar el bon funcionament de la plataforma, així com de tenir la possibilitat d'afegir noves funcionalitats, nous idiomes, l'adaptació a aplicació nativa per a Android i iOS, etc. El preu del manteniment mensual s'hauria d'establir segons la quantitat d'hores que el nostre potencial client/patrocinador hi volgués dedicar. Es proposa un manteniment de 5 hores amb un cost de 125€ mensuals.

Cal aclarir que en un cas real s'hauria destinat més recursos a dissenyar, maquetar i (sobretot) programar per tal de tenir un producte amb una qualitat òptima, però a causa de la gran quantitat d'activitats i del poc temps dispostat, s'ha retallat hores de tots els apartats.

Concepte	Import
Costos humans de desenvolupament	7000€
Manteniment mensual (5 hores)	125€/mes

Taula 4. Recursos humans

Recursos materials

Pel que fa als recursos materials, l'únic cost que caldria assumir en un primer moment és el del nom del domini (www.faenio.com) amb un import de 6,99 euros anuals. Durant els primers mesos l'aplicació s'allotjarà a Heroku en la seva versió més bàsica, gratuïta. Un cop s'aconsegueixi captar l'atenció de més usuaris es considerarà canviar de pla dins del propi Heroku o cercar una altra alternativa.

Les llicències del programari utilitzat (bàsicament la llicència del programari d'Adobe) no s'han tingut en compte en el pressupost, ja que venen proporcionades sense cost addicional com a recurs educatiu de la UOC.

Concepte	Proveïdor	Import
Nom de domini faenio.com	Hostalia	6,99€ (+IVA) anuals
Servidor web (1 dyno)	Heroku	0€

Taula 5. Recursos materials

En cas que l'aplicació s'acabi adaptant com a aplicació nativa per a mòbils caldria assumir el cost de la llicència d'Apple (99€/any²⁰). La llicència per a Android, en canvi, no té cap cost.

21.2 Anàlisi de mercat

Audiència potencial

Els perfils dels usuaris potencials de l'aplicació poden ser:

- Persones que disposen de connexió a Internet
- Persones que volen obtenir uns diners addicionals amb feines extres
- Persones sense feina o sense una feina fixa

²⁰ Apple Developer Program <https://developer.apple.com/programs/>

- Persones amb una formació professional que vulguin treure rendiment econòmic amb les seves habilitats
- Persones no professionals amb alguna habilitat que volen explotar econòmicament
- Persones que utilitzin les noves tecnologies per a trobar persones que els solucionin una necessitat

Competència

En el mercat de la oferta i demanda laboral hi ha molts competidors, com ara grans projectes consolidats com són **Infojobs**, **Nubelo**, **Job&Talent**, **JobToday**, **LinkedIn** (que té la seva pròpia aplicació amb ofertes de feina), **Cornerjob**... Però totes elles estan basades en un model en el que una empresa penja un anunci de feina i els candidats envien els seus currículums.

Heus aquí alguns exemples:

Nubelo

Nubelo va néixer com una aplicació web on únicament es publicaven demandes de professionals autònoms per a projectes concrets, publicats tant per empreses com per persones físiques que no podien realitzar alguna tasca ells mateixos. Actualment apart d'haver-se adaptat al mòbil amb una aplicació específica, també permeten publicar ofertes de feina en règim d'assalariat i, fins i tot, es dediquen a cercar ells mateixos candidats per a les empreses que compten amb els seus serveis. Aquest servei de cerca activa de candidats és un dels seus models de negoci, estalviant a les empreses (a canvi d'una suma econòmica) el farragós procés de selecció de perfils.

Per altra banda també compten amb un servei *premium* que permet als candidats veure les ofertes abans que aquells que no ho són, de manera que poden contactar amb els autors i fer-se un lloc abans. El servei *premium* també té altres avantatges com per exemple presentar un nombre il·limitat de candidatures a ofertes/projectes de feina (sense *premium* hi ha un màxim de 15 candidatures).

Nubelo no segueix el mateix concepte que **faenio**, ja que no deixa de ser un "es busca candidat", però al menys sí que té el toc personal de que qualsevol usuari pugui publicar una oferta demanant les habilitats de tercers.

cornerjob®

Cornerjob és una *startup* catalana que es promociona com “La *app* N°1 per a cerca de feina” i tot i que té pàgina web, aquesta serveix bàsicament per a promocionar l'aplicació mòbil.

El concepte d'aquesta aplicació és el tradicional de “Es busca candidat”, que no és la filosofia de **faenio**. El que sí que té com a tret similar es que mostra les ofertes de feina per proximitat geogràfica, aprofitant la geolocalització del telèfon mòbil.

Se'n desconeix el model de negoci: per ara sembla que tot es gratuït. I s'ho poden permetre gracies als 10 milions de dòlars de finançament²¹.



Hulajob és una aplicació web (no té una versió específica per a mòbils) amb un concepte diferent: en aquest cas l'usuari omple el seu perfil amb les seves habilitats, formació i disponibilitat i espera a que les empreses contactin amb ells. Tot i que a priori pot semblar un concepte similar a **faenio**, en realitat no ho és ja que la finalitat de **faenio** es que usuaris particulars (qualsevol persona) pugui veure els professionals que té a prop i cercar-los segons les seves necessitats, però en la plataforma **Hulajob** els

professionals no es poden veure a menys que es tingui un perfil d'empresa. I aquí ve el seu model de negoci²²:

Una de les seves fonts d'ingressos és revelar les dades dels candidats si s'està realitzant una cerca activa, pagant una certa quantitat per cada candidat mostrat. Per altra banda també compten amb un compte *premium* per empreses que, per una quantitat mensual, poden veure les dades de tots els candidats de l'aplicació de manera il·limitada. Finalment la tercera via d'ingressos és la de permetre que empreses publiquin anuncis i cobrar-los una comissió sobre el que es vulgui pagar al treballador.

²¹ “CornerJob alcanza los 10 millones de dólares de financiación con la participación de La Caixa” <http://www.caixacapitalrisc.es/es/cornerjob-alcanza-los-10-millones-de-dolares-de-financiacion-con-la-participacion-de-la-caixa/>

²² Model de negoci de HulaJob explicat en video pel seu CFO Jaime Durán (octubre 2014): <https://www.facebook.com/250592791754846/videos/384166941730763/>

InfoJobs

No es pot fer un llistat d'aplicacions relacionades amb la cerca de feina sense parlar d'**InfoJobs**, que és sense cap dubte la borsa de treball líder a Espanya (i que tot i tenir gairebé 20 anys se la considera una de les 5 *startups* espanyoles més influents).

InfoJobs ha estat sempre molt coneguda per la seva borsa de feines en règim d'assalariat, tot i que també té un portal especialitzat dedicat a ofertes de treball autònom. En qualsevol cas, cap dels dos conceptes és el que segueix **faenio**, que es centra en mostrar els professionals disponibles i no un aparador d'ofertes d'empresa.

Tot i que sembla impossible competir amb **InfoJobs**, apart de basar-se en un concepte totalment oposat també hi ha el fet que està tant massificat que és molt difícil destacar per sobre d'altres candidats. Si una oferta té 500 persones apuntades és gairebé com buscar l'agulla al paller.

21.3 Micro-pla d'empresa

Nom Comercial

faenio

Resum comercial

faenio és una plataforma gratuïta que serveix d'aparador per a que persones que vulguin treure rendiment econòmic d'alguna habilitat publiquin anuncis oferint els seus serveis i serveix com a catàleg de professionals a aquelles persones que necessitin una ajuda.

Model de negoci

En la primera fase del projecte el model de negoci és no tenir model de negoci: es tracta d'arribar al màxim de persones possible oferint tots els serveis sense cap cost (i per tant sense benefici per a l'empresa).

En una segona fase del projecte s'introduiran millores per aquells professionals que necessitin unes característiques especials i per tant s'avinguin a contractar una tarifa mensual per un pla "premium".

Competència

Hi ha moltes aplicacions tant d'escriptori com mòbil dedicades a la recerca de feina (Nubelo, Job&Talent, Cornerjob, HulaJob, Infojobs,...), però cap que segueixi el mateix concepte de "catàleg de professionals" que té **faenio**.

Estratègia de màrqueting

La promoció de l'aplicació disposa de limitacions evidents de manca de pressupost inicial, per tant es faria una promoció basada en la teòrica de cost zero, utilitzant totes les eines disponibles que no tinguin un cos associat a la plataforma, com ara la promoció a les diverses xarxes socials exercint el paper forçat de *Community Manager*.

Inversió inicial i costos a curt i mitjà terminis

La inversió inicial seria tan minsa com 7€ al any. I una bona quantitat d'hores de feina. A mitjà termini caldria costejar un allotjament web amb millors prestacions, que costaria aproximadament 240€ a l'any, que caldria millorar en funció de l'acollida dels usuaris.

DAFO

DEBILITATS

1. Marca nova al mercat
2. Capacitat financera limitada
3. Manca d'experiència empresarial

FORTALESES

1. Servei demandat i necessari
2. Relació qualitat-preu
3. Motivació

AMENACES

1. Molts projectes dins del mateix mercat en els últims temps
2. Competència consolidada
3. Equip limitat

OPORTUNITATS

1. Originalitat en l'enfocament
2. Producte sense competència real
3. Mercat en creixement
4. Alta demanda

21.4 Conclusions de viabilitat

Tot i que existeixen moltes aplicacions tant web com mòbil enfocades a la cerca de feina, sembla que encara pot haver-hi lloc per una més, sobretot si l'enfocament d'aquesta és al contrari al que estilen la resta d'aplicacions.

Caldria posar en pràctica la primera fase del projecte i veure l'acceptació del producte entre els usuaris per veure si realment el model de negoci previst tindria èxit i si es podrien estudiar altres vies d'ingressos.

Els costos de producció de la primera fase de desenvolupament que s'han vist en l'apartat de pressupost són molt assumibles, especialment tenint en compte que els únics costos econòmics reals serien els de manteniment del servidor, ja que a menys que es compti amb un inversor, ningú no pagarà per les hores de treball invertides.

Així que, tot i que encara s'hi hauran de dedicar més hores de desenvolupament per acabar de fer un producte rodó, no se'n descarta la viabilitat en un mercat ansiós per a seguir creixent.

22. Conclusions

Un dels objectius principals alhora d'afrontar aquest projecte era el d'aprendre a utilitzar noves tecnologies de desenvolupament i considero que l'he assolit amb escreix, adquirint coneixements sobretot d'Angular, Node i MongoDB, totes tres amb un potencial enorme del que només n'he vist la punta de l'iceberg, i que m'han deixat amb més ganes de seguir aprenent.

Dur a terme aquest projecte ha estat molt dur, tant per mi com per les persones al meu voltant per a qui he estat pràcticament absent durant els darrers tres mesos, però considero que tant pel resultat obtingut com sobretot per els coneixements adquirits ha valgut molt la pena i m'ha aportat moltes satisfaccions.

Per altra banda, la realització d'aquest treball m'ha servit per a consolidar molts dels coneixements adquirits en el transcurs d'aquests 5 anys estudiant el Grau Multimèdia, especialment pel que fa a la gestió de projectes i a l'enginyeria de programari.

He pogut comprovar la importància d'una bona planificació i anàlisi alhora de desenvolupar un projecte d'aquestes dimensions. En el meu cas, considero que la planificació que vaig fer va estar prou bé però vaig ser massa optimista en l'anàlisi i descripció de funcionalitats a desenvolupar de manera que el projecte se m'ha fet massa gran pel temps dispost. No obstant això, gracies al seguiment diari de tasques amb Trello²³ he pogut ajustar-me prou a la idea inicial del projecte, cosa que amb una planificació més laxa (i amb menys força de voluntat) no hauria pogut assolir.

S'entrega una versió final estable de **faenio** en la que l'usuari pot registrar-se, donar d'alta serveis, cercar-los per categoria, localització i paraules clau, mantenir converses en temps real amb altres usuaris i deixar valoracions sobre serveis rebuts, entre altres. Tot i ser un bon punt de partida, el producte encara té marge per a seguir evolucionant, així que posteriorment al lliurament final s'hi afegiran gradualment tant les funcionalitats que han quedat com les que s'han llistat com a futuribles amb l'objectiu de convertir-lo en un projecte amb usuaris reals que en facin un ús real.

²³ Veure referència a l'apartat "7.1 Programari" de la memòria del projecte

Annex 1. Lliurables del projecte

S'entreguen els arxius que es llisten a continuació:

- **Memòria:** El present document amb tota la informació referent a l'aplicació desenvolupada
- **Presentació en vídeo**
- **Presentació en diapositives (PPT)**
- **Presentació en vídeo en format anunci**
- **Codi font de l'aplicació:** Directori "faenio" amb tot el contingut de l'aplicació tant a nivell de servidor com de client

El producte final també es pot trobar online a la següent adreça: www.faenio.com

Annex 2. Codi font (extractes)

En les properes pàgines es mostren alguns fragments de codi de l'aplicació una mica representatius de les diferents parts que la componen. Els fragments van acompanyats d'una breu descripció del seu funcionament a mode de referència.

Aplicació client: encaminador del perfil

L'arxiu `profile.js` forma part de l'aplicació del client (Angular) i s'encarrega de definir el format de la ruta cap a una pàgina de perfil d'un usuari format per **profile** i seguit d'un paràmetre que serà l'identificador de l'usuari.

```
angular.module('faenio')
  .config(function ($stateProvider) {
    $stateProvider
      .state('profile', {
        url: '/profile/:profileId',
        views: {
          'header': {
            templateUrl: 'app/components/header/header.html',
            controller: 'HeaderCtrl'
          },
          'main': {
            templateUrl: 'app/profile/profile/profile.html',
            controller: 'ProfileCtrl'
          },
          'footer': {
            templateUrl: 'app/components/footer/footer.html',
            controller: 'FooterCtrl'
          }
        },
        data : {
          cssClassnames : 'profile-page not-home',
          requireLogin: false
        }
      })
  })
```

A continuació es defineix la vista o vistes que formaran part d'aquest estat, indicant l'adreça on es troba el *template* i quin controlador se n'encarrega. D'aquesta manera es pot modularitzar codi que s'ha de repetir al llarg de les pàgines de l'aplicació, i es pot decidir també si en algun moment donat es vol canviar la capçalera per una altra diferent, o no incloure el peu de pàgina, per exemple.

Els diferents objectes definits dins de “*views*” es col·locaran a la pàgina principal de l'aplicació, és a dir a l'arxiu *index.html*:

```
<div data-ui-view="header"></div>
<div data-ui-view="main"></div>
<div data-ui-view="footer"></div>
```

Dins l'objecte *data*, a *cssClassnames* s'hi indica una o diverses classes que seran col·locades a l'etiqueta *body* (que du l'atribut *route-css-classnames*) i d'aquesta manera es poden aplicar estils personalitzats a cada pàgina.

Finalment, *requireLogin* defineix si la pàgina necessita que l'usuari estigui autenticat o no és necessari. Així per exemple per a accedir a un perfil *requireLogin* serà *false*, mentre que per enviar un missatge a un altre usuari, *requireLogin* serà *true*.

API

Com ja s'ha comentat, l'aplicació està construïda per a que pugui actuar com una API REST. En principi només és per a servir-se a ella mateixa, però estaria oberta a l'escalabilitat.

Per a simplificar les crides de funcions s'han creat diversos fitxers de factoria d'API dins de l'aplicació client (*commentsAPI.js*, *locationsAPI.js*, *messagesAPI.js*, *postsAPI.js* i *usersAPI.js*), totes elles amb una estructura similar a la següent:

```
angular
  .module('faenio')
  .factory('usersAPI', function ($http) {

    return {
      getProfile: getProfile,
      updateProfile: updateProfile,
    }

    function getProfile(userId) {
      return $http.get('/api/users/' + userId);
    }

    function updateProfile(user) {
      return $http.put('/api/users/' + user._id, user);
    }

  })
```

En l'exemple sobre aquestes línies es veu *usersAPI.js* que només té dues funcions: *getProfile*, que recull el perfil d'un usuari per mètode GET passant només la id i *updateProfile*, que actualitza el perfil passant per mètode PUT l'objecte usuari.

Al servidor, al controlador *index.js* del mòdul d'usuaris (*server/api/user/*), s'hi implementa un encaminador (similar al que s'ha vist del costat de client) que recull la crida i crida la funció que té assignada:

```
var express = require('express');
var controller = require('./user.controller');
var router = express.Router();
var config = require('../../config/environment');
var auth = require('../../auth/auth.service');

router.post('/', controller.create);
router.get('/:id', controller.getProfile);
router.put('/:id', auth.isAuthenticated(), controller.update);

module.exports = router;
```

Nota: Al costat de client no s'ha implementat una funció de factoria per a crear usuaris, perquè es creen per una altra banda (des de *auth.Srv.js*, dins de la carpeta *app/components/auth*)

La funció que s'ha cridat des de *index.js* es troba a l'arxiu *user.controller.js* (dins de la mateixa carpeta */user/*). Per exemple, la de *getProfile* seria així:

```
/**
 * Get a single user
 */
exports.getProfile = function(req, res, next) {
  var userId = req.params.id;

  User
    .findById(userId)
    .populate('comments._user comments._post')
    .exec(function (err, user) {
      if(err) { return res.send(500, err); }
      if(!user) { return res.send(404); }
      return res.json(user);
    });
};
```

Enviament de missatges en temps real

La llibreria *socket.io* proporciona una comunicació bidireccional entre client i servidor, així que permet que un usuari pugui escriure un missatge a un altre i que el segon el rebí sense necessitat de refrescar el navegador.

Per una banda, a l'arxiu *app.js* de l'aplicació servidor s'hi importa la llibreria, se li indica que treballi sobre el servidor Express declarat prèviament i se li diu que, quan es rebí un esdeveniment "nou missatge" cridi a la funció "missatge":

```
var socketio = require('socket.io');
var app = express();
var server = http.createServer(app);
var io = socketio(server);

io.on('connection',function(socket){
  socket.on('nou missatge', function(msg, user){
    io.emit('missatge', {msg: msg, user:user});
  });
});
```

Per part del client, concretament al controlador de missatges (*messages.controller.js*), es defineix una funció "sendMsg" que s'executa quan l'usuari envia el formulari amb el fragment de conversa. Aquesta funció per una banda emmagatzema el missatge a la base de dades i per altra emet l'esdeveniment "nou missatge".

```
$scope.sendMsg = function() {

  socket.emit('nou missatge', $scope.text, $scope.currentUser);
  var message = {
    msg : $scope.text,
    _chatRoom : $scope.chatRoom,
  }

  messagesAPI.newMessage(message)
  .then(function(data) {
    $scope.text = "";
  })
  .catch(function(err) {
    console.log(err);
    $scope.text += " (no s'ha desat)";
  });
}
```

Com s'ha indicat al servidor, a l'emetre "nou missatge" es crida la funció "missatge", que en aquest cas construeix el text del missatge i el mostra per pantalla.

```
socket.on('missatge', function(data){
  // check if is the first message
  var chatHtml = chat.html();
  var isFirst = false;

  if (chatHtml.lastIndexOf("chat-first") > 0)
```

```
        isFirst = true;

    var jo = "";
    if (data.user._id == $scope.currentUser._id)
        jo = " (jo)";

    var contentString = "<p><span class='chat-time'>" +
moment().format('L') + " | <span>" + moment().format('LT') +
"</span></span>" + "<strong>" + data.user.firstName + jo +
"</strong>" + " - " + data.msg + "</p>";

    if (isFirst)
        chat.html(contentString);
    else
        chat.append(contentString);

    chat.animate({scrollTop: chat.get(0).scrollHeight},1000);
})
```

En el codi que s'acaba de veure hi ha certs controls per a millorar l'experiència d'usuari: per una banda es controla si és o no el primer missatge de la conversa (es pot mostrar historial o un "Escriu el teu primer missatge") i també si l'usuari "sóc jo mateix" ja que en aquest cas s'hi indica amb un "(jo)" al costat del nom, per evitar confusions (poden parlar dues persones amb el mateix nom).

Finalment, la darrera línia del codi fa que hi hagi un *scroll* cap al missatge més recent , que serà el primer començant per a baix, amb una transició d'un segon.

Aplicació servidor: Cerca de publicacions

Per a realitzar la cerca de publicacions es poden rebre paràmetres de localització (*location*), categoria (*category*) i paraules clau (*query*) i segons aquests paràmetres s'implementarà un filtre o un altre:

- Si la localització està definida, es cercaran els resultats dintre del polígon determinat (cosa que s'aconsegueix gracies a la funció `within()` de Mongoose)
- Si la categoria esta definida, es cercaran els resultats que tinguin el nom de categoria coincident amb el paràmetre rebut
- Si les paraules claus estan definides, es cercaran els resultats que coincideixin ja sigui pel títol de la publicació com per nom de categoria (per exemple, es pot haver posat com a paraula clau "reparacions", que coincideix amb un nom de categoria)

```

exports.searchResults = function(req, res) {
  var location = req.body.location;

  var category = "123456";
  if (req.body.category != undefined)
    category = req.body.category;

  var query = "123456";
  if (req.body.query != undefined)
    query = req.body.query;

  var myQuery = Post.find({});

  if (location.geometry != undefined)
    myQuery =
myQuery.where('location').within().geometry(location.geometry);

  if (req.body.category)
    myQuery = myQuery.where('category.name', new
RegExp(category, 'i'));

  if (req.body.query)
    myQuery.or([{'category.name': new RegExp(query, 'i')}, {
'name': new RegExp(query, 'i')}]);

```

En primer lloc s'han fet les validacions. Després s'agafen tots els serveis amb `Post.find({})` i a partir d'aquí es van filtrant els resultats.

Un cop s'ha construït “*myQuery*” (“la meva consulta”), s'executa i a més s'indica que els serveis es mostrin ordenats per ordre de creació invers. I finalment, amb el *plugin* de *Mongoose* anomenat “*Deep Populate*”, es retornen les dades de l'objecte **User** associat a cadascun dels articles:

```

myQuery
.sort({
  createTime: -1
})
.deepPopulate('_user')
.exec(function(err, posts) {
  if (err) {
    console.log(err);
    return res.send(500, err);
  }
  if (!posts) {
    console.log("nothing found");
    return res.send(404);
  }
  return res.status(200)
.json(posts);
});
};

```

Annex 3. Llibreries externes

Quan es treballa amb aplicacions Javascript és gairebé inevitable dependre de llibreries de tercers. En aquest projecte s'ha utilitzat **NPM** (Node Package Manager) per a la gestió de llibreries del costat de servidor i **Bower** per a les llibreries del costat del client, amb els arxius **package.json** i **bower.json** respectivament com a "índex" de les llibreries que en depenen.

A continuació es llistaran els mòduls i llibreries utilitzats amb una breu descripció de la seva funcionalitat com a referència.

Servidor

- **angular-route (ngRoute)**

Mòdul que proporciona serveis d'encaminament i serveis i directives d'enllaços profunds per aplicacions desenvolupades en AngularJS.

URL: <https://docs.angularjs.org/api/ngRoute>

- **body-parser**

Middleware que s'utilitza per analitzar sintàcticament (en anglès, "parse") els objectes enviats per POST i comprovar que estan correctament escrits, i retorna un resultat accessible a través de la variable *req.body*

URL: <https://www.npmjs.com/package/body-parser>

- **method-override**

És un *middleware* que realitza una verificació sobre la variable *req.body* i serveix per a sobreesciure els mètodes dels formularis

URL: <https://www.npmjs.com/package/method-override>

- **composable-middleware**

Permet una sèrie de funcions de *middleware* per a ser utilitzades com si fossin una sola peça. Així un paquet de *middleware* es converteix en un sol objecte de primera classe.

URL: <https://www.npmjs.com/package/composable-middleware>

- **connect-mongo**

Permet establir la connexió amb MongoDB a través de Express i emmagatzemar-ne la sessió.

URL: <https://www.npmjs.com/package/connect-mongo>

- **cookie-parser**

S'utilitza per al tractament de *cookies*, facilitant la creació i consulta dins de l'aplicació

URL: <https://www.npmjs.com/package/cookie-parser>

- **debug**
Utilitat per a treballar en desenvolupament i mostrar els errors per consola de comandes
URL: <https://www.npmjs.com/package/debug>
- **express**
És una llibreria d'alt nivell per a desenvolupament web amb Node.js dissenyada per desenvolupar aplicacions "single-page", "multi-page" i híbrides.
URL: <https://www.npmjs.com/package/express>
- **express-jwt**
Mòdul de *middleware* que permet autenticar peticions HTTP utilitzant *JsonWebTokens* en aplicacions Node.js
URL: <https://github.com/auth0/express-jwt>
- **express-session**
Middleware d'Express per al tractament de sessions
URL: <https://www.npmjs.com/package/express-session>
- **lodash (Lo-Dash)**
Llibreria per a manipular estructures JSON de manera simplificada. Inicialment va ser un fork de la llibreria Underscore.js, per això és similar, però ha afegit més opcions de funcionalitat
URL: <https://www.npmjs.com/package/lodash>
- **mongoose**
És una llibreria per a treballar amb Node.js i MongoDB, oferint una solució senzilla basada en esquemes per a modelar dades d'aplicació. Inclou conversió de tipus, validació, generació de consultes, lògica de negoci i més.
URL: <https://www.npmjs.com/package/mongoose>
- **mongoose-deep-populate**
Plugin per a Mongoose que permet habilitar "populates" profunds en models niats
URL: <https://www.npmjs.com/package/mongoose-deep-populate>
- **morgan**
Middleware registrador de sol·licituds HTTP per a Node.js, nomenat "Morgan" per la sèrie televisiva "Dexter"
URL: <https://www.npmjs.com/package/morgan>
- **passport**
Middleware per a l'autenticació amb Node.js. S'utilitza en aquesta aplicació per a iniciar sessió o registrar un usuari mitjançant Twitter, Facebook o amb un usuari i paraula de pas
URL: <https://www.npmjs.com/package/passport>

- **passport-facebook**
Estratègia de Passport per a l'autenticació amb Facebook mitjançant l'API OAuth 2.0.
URL: <https://github.com/jaredhanson/passport-facebook>
- **passport-local**
Estratègia de Passport per a l'autenticació mitjançant un nom d'usuari i paraula de pas
URL: <https://www.npmjs.com/package/passport-local>
- **passport-twitter**
Estratègia de Passport per a l'autenticació amb Twitter mitjançant l'API OAuth 1.0
URL: <https://www.npmjs.com/package/passport-twitter>
- **requirejs**
Adaptador de Node per a RequireJS, per a la càrrega de mòduls AMD
URL: <https://www.npmjs.com/package/requirejs>
- **serve-favicon**
Middleware de Node.js per a servir el *favicon* de l'aplicació
URL: <https://github.com/expressjs/serve-favicon>
- **socket.io**
Llibreria de Node.js per a treballar a temps real. En aquesta aplicació s'utilitza per als missatges de xat
URL: <https://www.npmjs.com/package/socket.io>

Client

- **angular**
AngularJS és un *framework* de desenvolupament MWC creat per Google que s'utilitza per a crear i mantenir aplicacions web d'una sola pàgina
URL: <https://github.com/angular/bower-angular>
- **bootstrap**
És una llibreria creada per Twitter que permet maquetar llocs web i aplicacions de manera ràpida oferint un conjunt d'eines CSS i Javascript que simplifiquen les tasques d'una maquetació des de zero
URL: <http://getbootstrap.com/getting-started/>
- **jquery**
Llibreria que permet la selecció d'elements del DOM de forma simplificada mitjançant el símbol \$ i ofereix opcions de manipulació, animació, simplificació de crides ajax, etc.
URL: <https://jquery.com/>

- **angular-ui-router**
Llibreria que simplifica l'encaminament de vistes amb AngularJS
URL: <https://github.com/angular-ui/ui-router>
- **angular-socket-io**
Component per a utilitzar la llibreria socket.io al costat del client amb AngularJS
URL: <https://github.com/btford/angular-socket-io>
- **socket.io-client**
Enviament i recepció d'esdeveniments a temps real. Llibreria del costat del client.
URL: <http://socket.io/docs/client-api/#>
- **angular-moment**
Directives d'angular per al plugin Moment.js
URL: <https://github.com/urish/angular-moment>
- **moment**
Llibreria que analitza, valida, manipula i mostra dates amb Javascript de manera amigable
URL: <http://momentjs.com/>
- **angular-resource (ngResource)**
Mòdul que proporciona suport d'interacció amb els serveis RESTful
URL: <https://docs.angularjs.org/api/ngResource>
- **ngGeolocation**
Proporciona suport per la API de geolocalització de HTML5 per tal d'oferir una manera fàcil i ràpida de consumir informació geogràfica proporcionada per navegadors moderns amb HTML5 en aplicacions desenvolupades amb AngularJS
URL: <https://github.com/ninjatronic/ngGeolocation>
- **noty**
Plugin de jQuery que simplifica el fet de mostrar missatges d'alerta, d'èxit i d'error
URL: <http://ned.im/noty/>
- **animate.css**
Estils CSS que afegixen animacions. Utilitzat per noty en el moment de mostrar i amagar les notificacions
URL: <https://daneden.github.io/animate.css/>
- **angular-animate (ngAnimate)**
Mòdul que proporciona animacions basades en CSS (keyframes i transicions) i Javascript per a aplicacions en AngularJS
URL: <https://docs.angularjs.org/api/ngAnimate>
- **angular-cookies (ngCookies)**

Mòdul que proporciona un contenidor per a la lectura i escriptura de galetes del navegador

URL: <https://docs.angularjs.org/api/ngCookies>

- **angular-touch (ngTouch)**

Mòdul que ofereix esdeveniments tàctils i altres ajudes per a dispositius amb el sensor tàctil habilitat. L'aplicació es basa en la gestió d'esdeveniments de JQuery Mobile

URL: <https://docs.angularjs.org/api/ngTouch>

- **angular-bootstrap**

Components de bootstrap reescrits per a adaptar-se a AngularJS. A l'aplicació s'utilitza per exemple per a les finestres modals d'inici de sessió i registre d'usuaris

URL: <https://angular-ui.github.io/bootstrap/>

- **bootstrap-star-rating**

Plugin de JQuery basat en Bootstrap per a la valoració de serveis i mostrar el resultat en forma d'estrelles

URL: <http://plugins.krajee.com/star-rating>

- **modernizr**

Llibreria que ajuda a donar suport a versions antigues de navegadors

URL: <https://modernizr.com/>

- **geolib**

Llibreria amb recursos per a tractar amb dades geogràfiques, com per exemple trobar el centre entre un *array* de coordenades

URL: <https://www.npmjs.com/package/geolib>

Annex 4. Captures de pantalla

Procés de prototipat

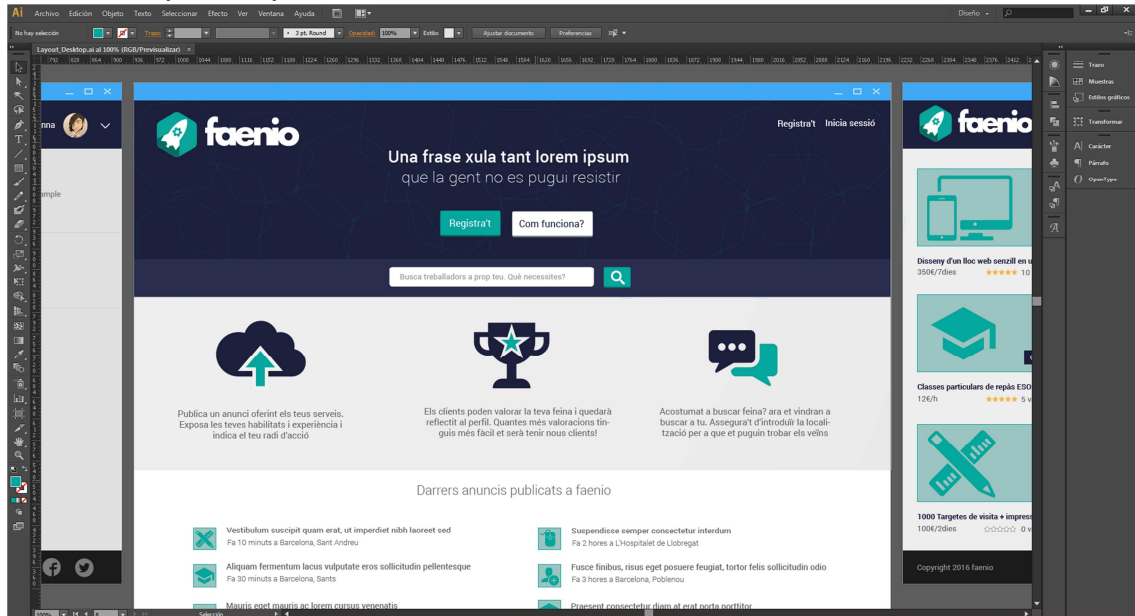


Figura 28. Disseny de les diferents pantalles de l'aplicació

Procés de desenvolupament

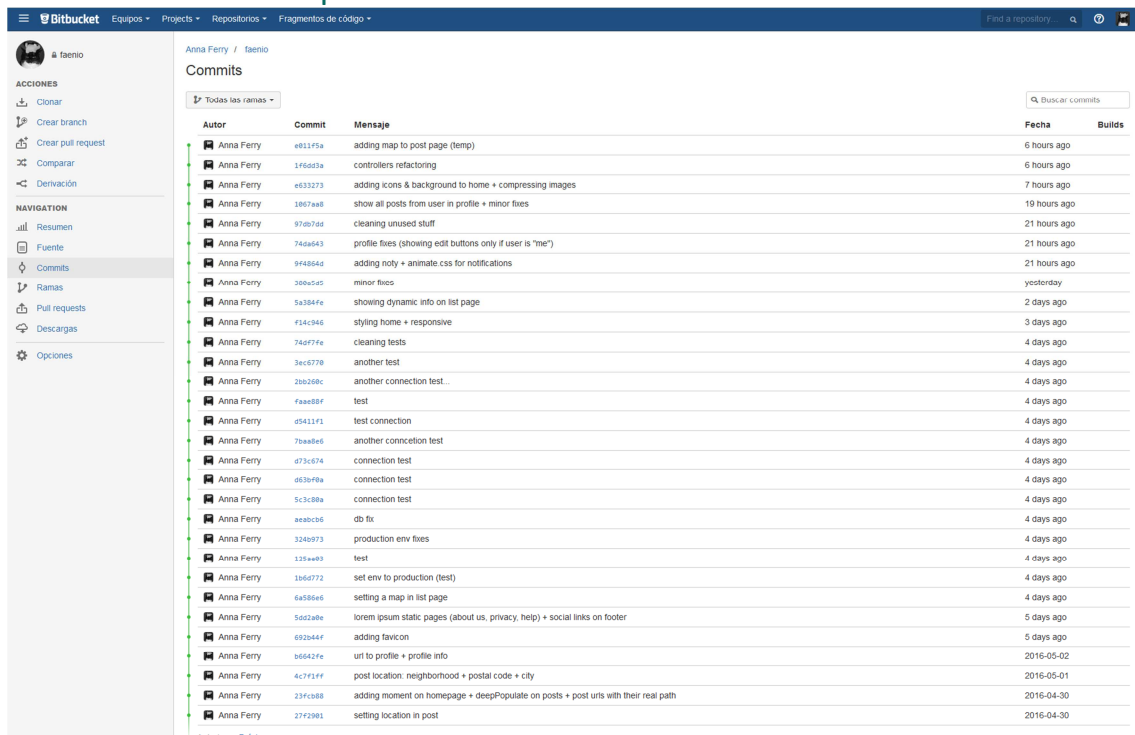
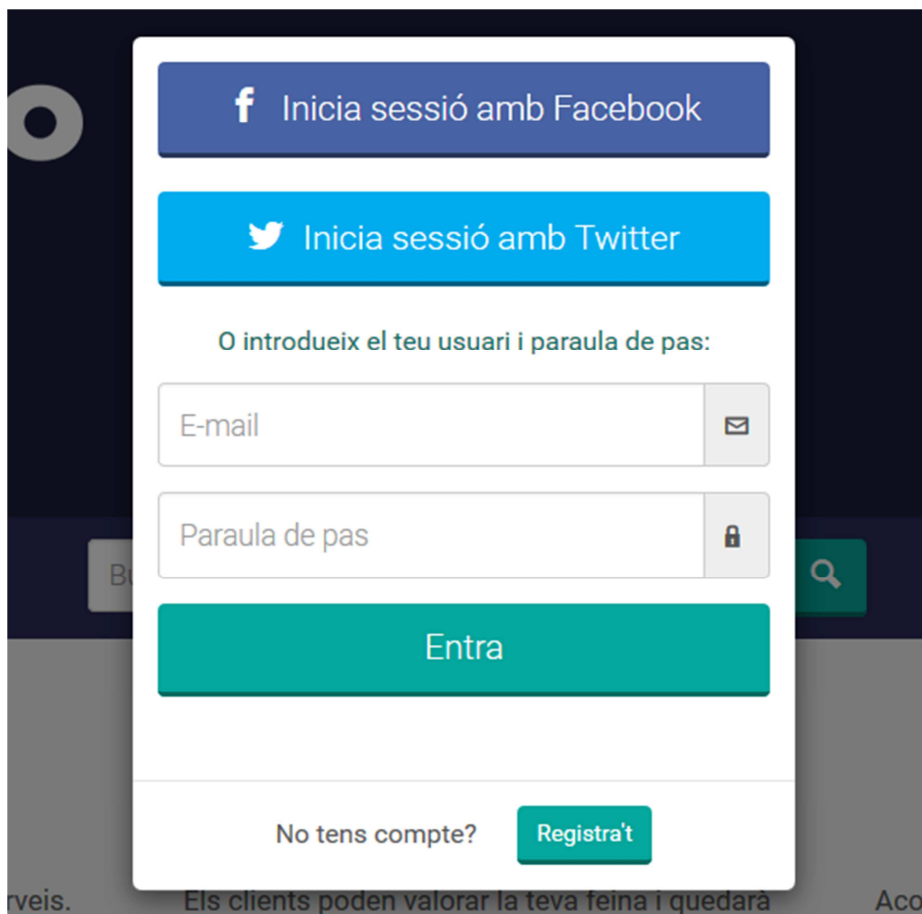


Figura 29. Historial de commits a Bitbucket

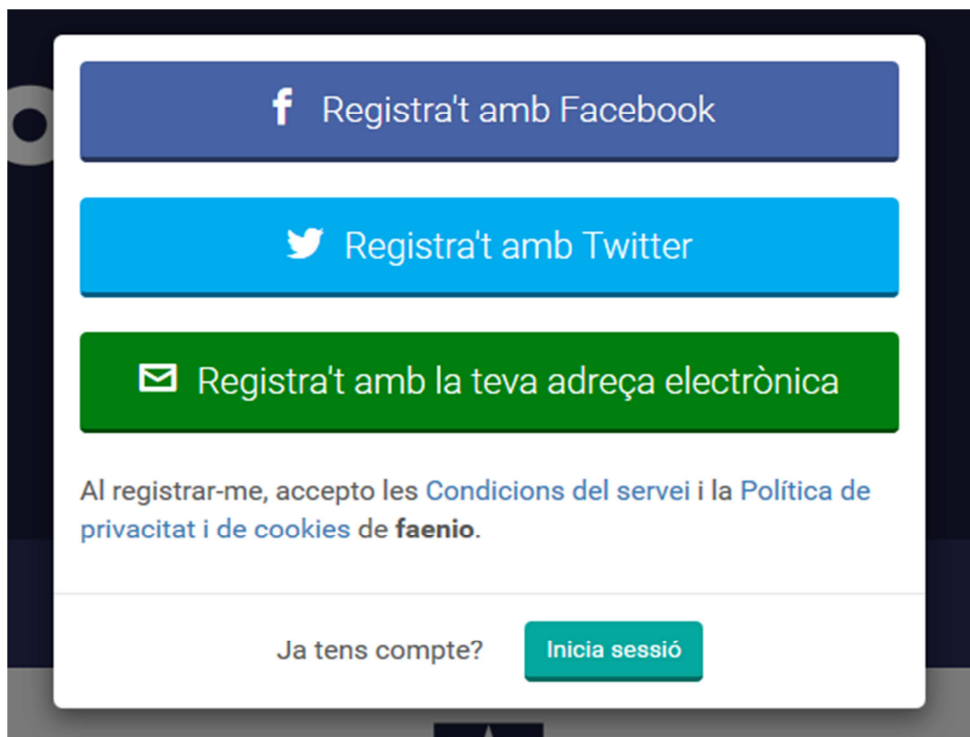
Interior de l'aplicació



Formulari d'inici de sessió amb opcions de registre i login:

- Inicia sessió amb Facebook
- Inicia sessió amb Twitter
- O introdueix el teu usuari i paraula de pas:
- E-mail
- Paraula de pas
- Entra
- No tens compte? [Registra't](#)

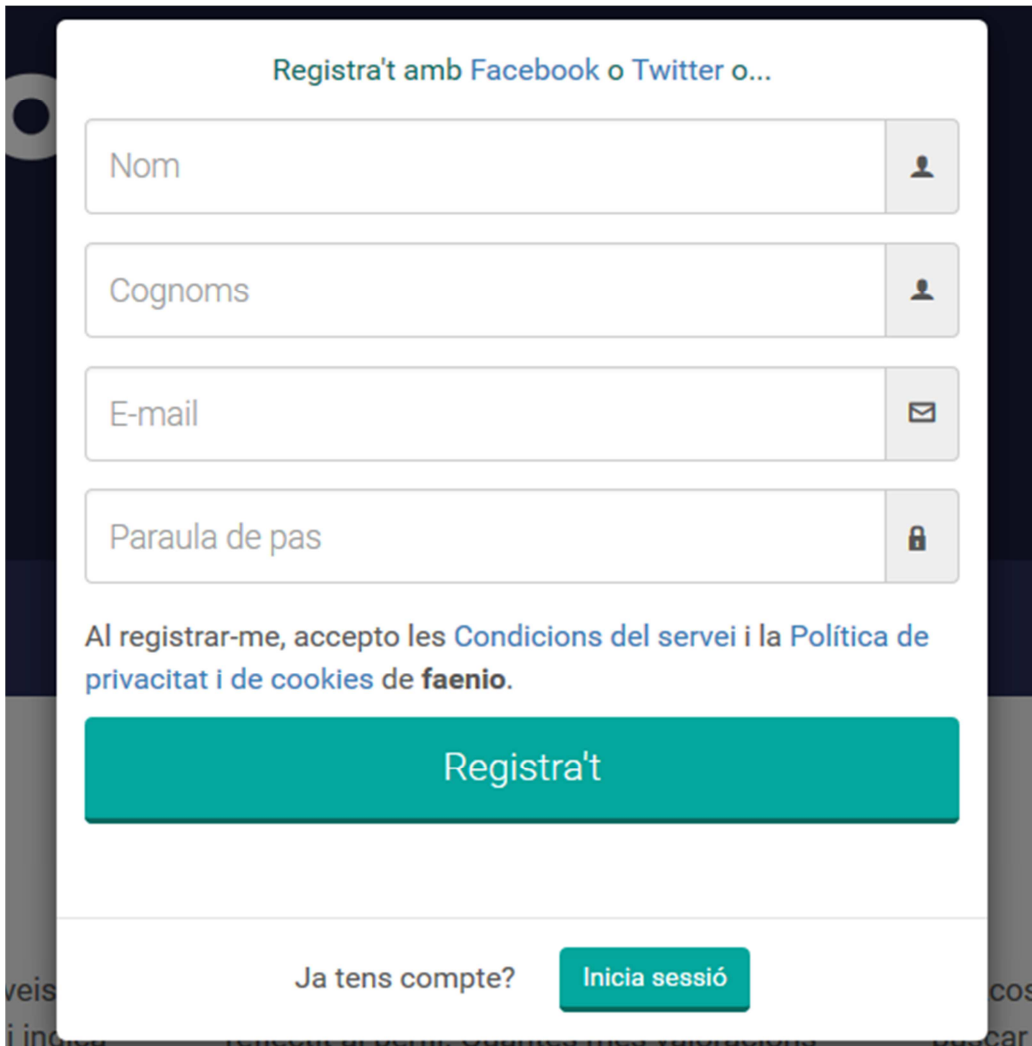
Figura 30. Formulari d'inici de sessió



Formulari de registre amb opcions de registre i login:

- Registra't amb Facebook
- Registra't amb Twitter
- Registra't amb la teva adreça electrònica
- Al registrar-me, accepto les [Condicions del servei](#) i la [Política de privacitat i de cookies](#) de **faenio**.
- Ja tens compte? [Inicia sessió](#)

Figura 31. Formulari de registre



Registra't amb Facebook o Twitter o...

Nom

Cognoms

E-mail

Paraula de pas

Al registrar-me, accepto les [Condicions del servei](#) i la [Política de privacitat i de cookies de faenio](#).

Registra't

Ja tens compte? **Inicia sessió**

Figura 32. Formulari de registre amb correu electrònic



Comentari

Valora i deixa un comentari sobre la teva experiència amb l'usuari

★★★★☆

Comentari del servei

Acceptar **Cancelar**

Figura 33. Valoració de serveis amb comentari i puntuació numèrica



Figura 34. Pàgina d'inici: filtre per categories

faenio Anna ▼

Nova publicació

Introdueix a continuació les dades del servei que vols oferir. Com més informació posis més atreurà als teus possibles clients, així que no et deixis res!

Nom:

Descripció:

Preu: € Preu/Hora Preu/Servei

Termini entrega:

Categoria:

Zona:

Figura 35. Nova publicació de servei

faenio Publica una oferta Anna

[← Torna a l'article sense desar](#)

Editar publicació

Introdueix a continuació les dades del servei que vols oferir. Com més informació posis més atreurà als teus possibles clients, així que no et deixis res!

Nom:

Descripció:

Preu: € Preu/Hora Preu/Servei

Termini entrega:

Categoria:

Zona: Troba'm

Eliminar publicació Desar modificacions

Figura 36. Editar publicació de servei

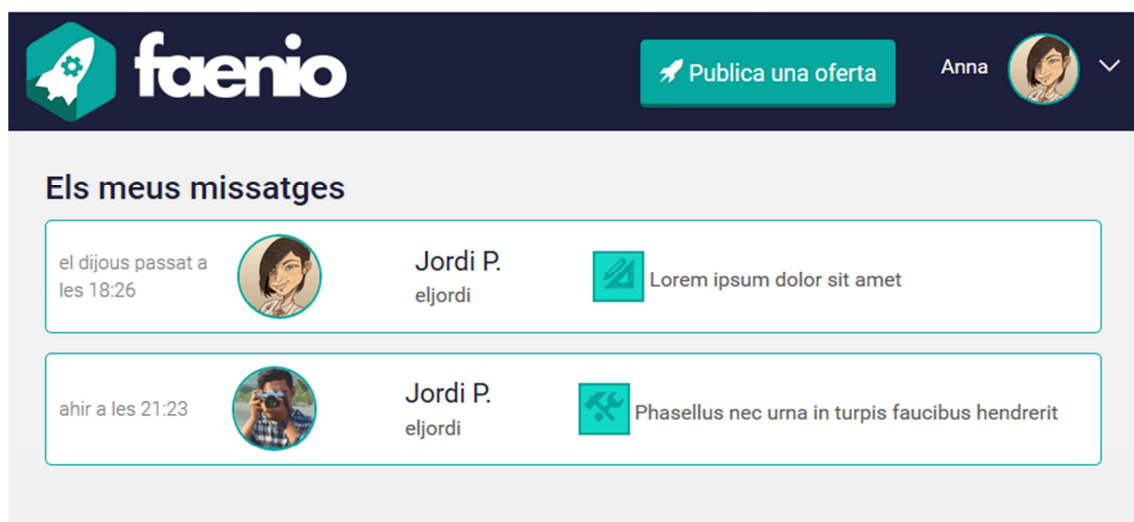


Figura 37. "Els meus missatges" (converses amb usuaris)

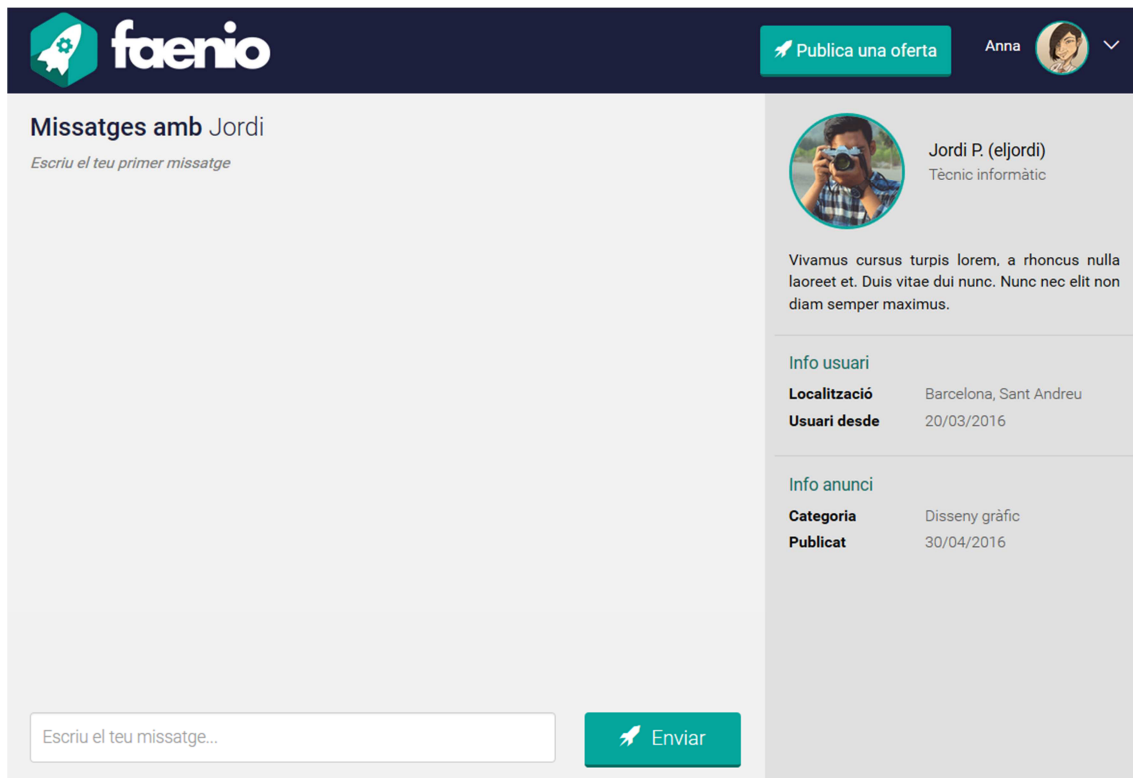


Figura 38. Nova conversa amb un altre usuari

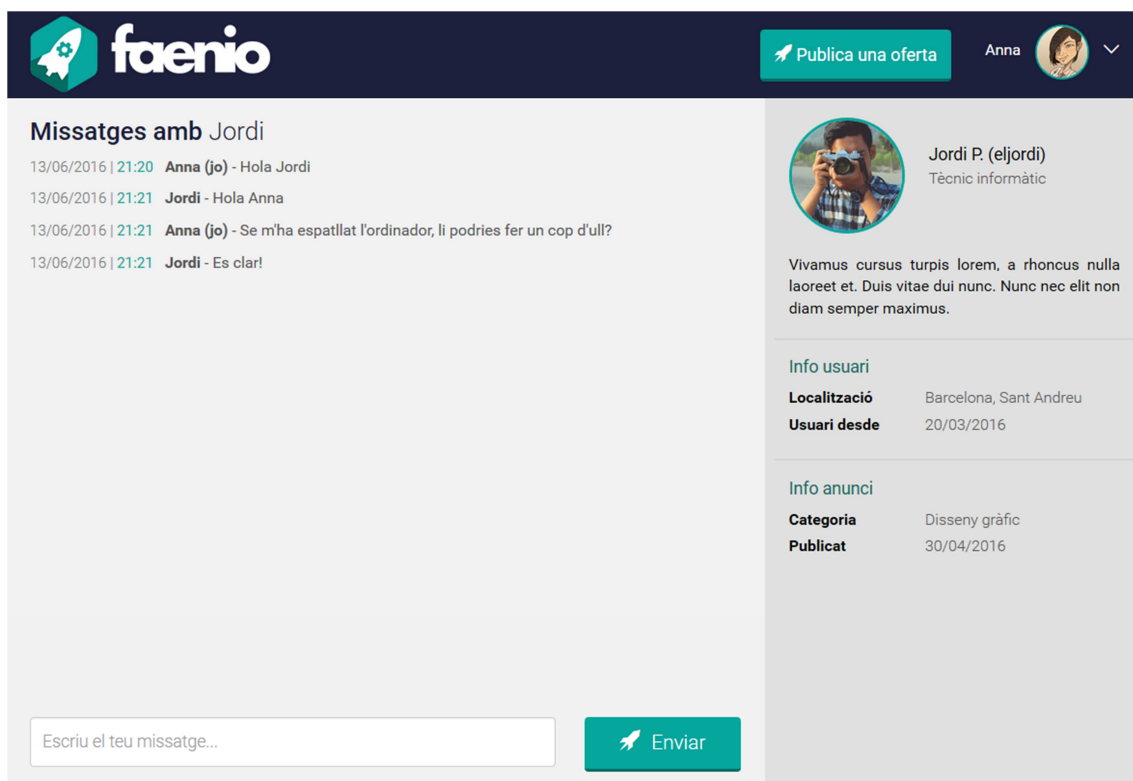



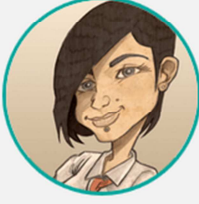
Figura 39. Conversa amb en Jordi sobre un servei

faenio [Publica una oferta](#) Anna 

[← Torna al perfil sense desar](#)

Editar Perfil

Edita la informació del teu perfil. Com més informació posis més confiança donaràs als teus possibles clients, així que no et deixis res! (Tingues present que tots els camps que omplis seran visibles publicament)



Nom

Cognoms

Nom d'usuari

Descripció breu


Biografia

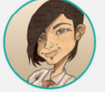
Localització [Troba'm](#)

E-mail

[Desar canvis](#)


Figura 40. Editar perfil d'usuari

faenio [Publica una oferta](#) Anna 


 **Anna Ferry** ★★★★★ ☆
 Multimediera
 Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus nec urna in turpis faucibus hendrerit eget sit amet nibh. Duis vitae dui nunc. Nunc nec elit non diam semper maximus.

[Edita el perfil](#)


Valoracions d'usuaris (1)

 **Jordi P.** ★★★★★ ☆
 Molt bé 30/03/2016


Serveis publicats (1)

 **Lorem ipsum dolor sit amet** [Editar](#)
 Publicat 30/04/2016 Categoria Disseny gràfic
 Estat actiu Vist 43 vegades [Eliminar](#)

Contactes rebuts (1)

 **Jordi P.**
 (Lorem ipsum dolor sit amet) [Assignar servei](#)

Serveis rebuts (1)

 **Phasellus nec urna in turpis faucibus hendrerit** [Votar](#)
 Publicat 30/03/2016 Categoria Reparacions
 Estat actiu Vist 23 vegades

Info

Localització Barcelona, Sant Andreu

Usuari desde 20/03/2016

Última connexió avui a les 21:19

Activitat

Serveis adjudicats 0

Serveis rebuts 1

Comparteix el perfil

[f](#) [t](#) [G+](#) [in](#)

© 2016 faenio [Sobre faenio](#) [Política de Privacitat](#) [Ajuda](#) [Segueix-nos a](#) [f](#) [t](#)

Figura 41. Perfil d'usuari

Annex 5. Guia d'usuari

Executar l'aplicació

L'aplicació es troba online allotjada a la plataforma *Heroku*. S'hi pot accedir tant des de <http://faenio.herokuapp.com> com des del domini <http://www.faenio.com>.

També es poden executar els fitxers entregats de manera local, tot i que per això cal tenir NodeJS i MongoDB instal·lats a l'equip. Mitjançant la terminal s'ha d'accedir a la carpeta on es troben tots els fitxers i executar “**npm install**” per instal·lar tots els mòduls dels quals depèn l'aplicació i que es troben definits al fitxer **package.json**.

Un cop fet això i amb MongoDB en execució, al teclejar “**grunt**” a la consola de comandes arrenca el servidor local i s'hi pot accedir des de <http://localhost:5000/>.

Iniciar sessió

A la part superior dreta de la pàgina d'inici hi ha un enllaç per a iniciar sessió que, al polsar-lo, obre una finestra emergent que permet iniciar sessió mitjançant Twitter, Facebook o utilitzant un correu electrònic i contrasenya. Les dues primeres opcions fan que l'autenticació i/o registre es facin en un sol clic.

Existeixen dos usuaris de prova (amb un perfil complet, articles publicats i valoracions rebudes) amb les credencials següents:

Usuari: test@test.com

Paraula de pas: 1234

Usuari: jordi@test.com

Paraula de pas: 1234

El fet d'entrar amb aquests dos usuaris permetrà veure com funciona una relació client-proveïdor, ja que es mostra per una banda l'usuari que publica serveis i rep contactes (amb la opció d'assignar un servei) i per l'altra banda l'usuari que ha rebut algun servei i té l'opció d'emetre una valoració.

Navegació per l'aplicació

A continuació es fa un petit passeig per l'aplicació per a veure les funcionalitats desenvolupades i què és el que s'ofereix en cadascun dels apartats.

A l'iniciar sessió, la capçalera canvia mostrant la informació de l'usuari. Fent clic sobre la fletxa que es troba al costat de la imatge d'usuari s'obre un menú des del que es pot anar al perfil, a la secció de missatges o tancar la sessió.

En el moment d'iniciar sessió, a la pàgina d'inici canvia el botó de "registrar" per un de "publicar anunci", que porta a una pàgina nova des de la que es pot crear un nou anunci.

Quan es crea un nou anunci, aquest apareix automàticament tant en la pàgina d'inici, com a la pàgina de llistat, com a la pàgina de perfil de l'usuari que ha publicat l'anunci i es mostra amb la imatge de la categoria a la que correspon.

A la pàgina de perfil d'un usuari es mostren els seus detalls, la llista de les seves publicacions i la llista de valoracions que han rebut.

Si la pàgina de perfil és la del propi usuari es mostren els botons d'edició de perfil i edició/eliminació de publicacions, però si es la d'un usuari diferent lògicament aquests botons no apareixen.

A la pàgina d'una publicació es mostra la informació relativa al servei incloent el mapa de la zona on es durà a terme i el llistat de valoracions rebudes. També es mostra informació bàsica de l'autor de l'anunci amb la possibilitat d'accedir al seu perfil.

Un usuari pot necessitar més informació sobre un servei determinat, així que ha de fer clic sobre el botó "Envia-li un missatge" (acció que només es pot realitzar si s'ha iniciat sessió prèviament) i anirà a una pàgina de xat, en la que ambdós usuaris poden intercanviar missatges en temps real.

Si ambdós usuaris arriben a un acord i s'acaba realitzant l'operació, l'usuari proveïdor del servei pot assignar-li el servei al seu client des del seu perfil, que significarà que el tracte s'ha realitzat. A partir d'aquell moment, l'usuari client veu al seu perfil aquell servei en la llista de serveis rebuts i en pot emetre una valoració.

Annex 6. Guia d'estil

A continuació es defineix la línia gràfica bàsica del projecte.

Logotip



Figura 42. Logotip sobre fons fosc



Figura 43. Logotip sobre fons clar

Paleta tipogràfica i mida de fonts

COS DE TEXT

Tipus de lletra: Roboto

Estil: Regular

Color: #404041

Mida de lletra escriptori: 16px

Mida de lletra smartphone: 14px

TITULARS SECUNDARIS (H3, h4)

Tipus de lletra: Roboto

Estil: Regular

Color: #086c62

Mida de lletra escriptori: 18px

Mida de lletra smartphone: 18px

TITULARS PRINCIPALS (H1, H2)

Tipus de lletra: Roboto

Estil: Regular / Bold

Color: #404041 / #1c1e3c

Mida de lletra escriptori: 26px

Mida de lletra smartphone: 20px

Paleta de colors

Els colors de **faenio** estan descrits amb colors RGB i en el seu valor equivalent en el sistema hexagesimal. Els colors principals es mostren a continuació:

#404041	#1c1e3c	#2d3051	#086c62	#05a69c
64,64,65	28,30,60	45,48,81	8,108,98	5,166,156

Retícula



Figura 44. Retícula de 12 columnes

S'ha optat per utilitzar una graella de 12 columnes, ja que és un format que permet molta flexibilitat. Per altra banda actualment és gairebé un Standard, ja que els frameworks fontend més utilitzats, com Bootstrap o Foundation, estan basats en el sistema de 12 columnes.

És un tipus de graella amb la que s'obté un ordre molt subtil, ja que una secció del web pot ocupar tres blocs, la següent 4 blocs de tres columnes i a continuació mostrar elements en dos columnes (de 6 i 6) però seguir mantenint una coherència.

MIDES

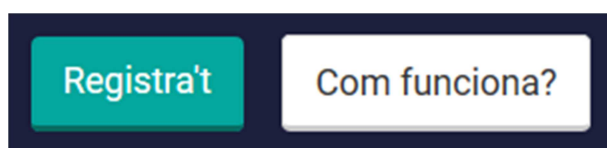
L'amplada d'una columna de la retícula equival a un 8,33% de l'amplada total. No obstant, qualsevol objecte que es col·loqui dins la retícula pot ocupar entre 1 i 12 columnes, i en aquests casos les relacions d'amplades serien les següents:

1 columna:	8,33333333%	7 columnes:	58,33333333%
2 columnes:	16,66666667%	8 columnes:	66,66666667%
3 columnes:	25%	9 columnes:	75%
4 columnes:	33,33333333%	10 columnes:	83,33333333%
5 columnes:	41,66666667%	11 columnes:	91,66666667%
6 columnes:	50%	12 columnes:	100%

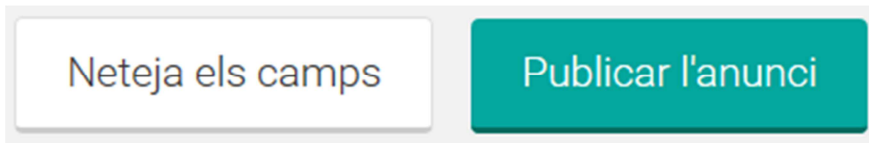
Entre columna i columna hi ha **una separació de 30px** (15px a cada banda de la columna, que es sumen als de la següent).

Elements d'interacció

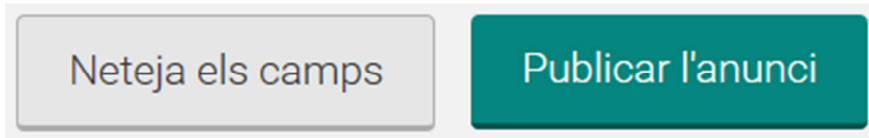
Botons de tipus principal (esquerra) i secundari (dreta):



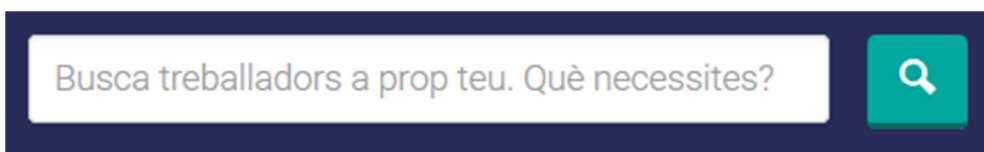
Botons en estat normal:



Botons amb el mouse a sobre (hover):



Cercador pàgina d'inici:



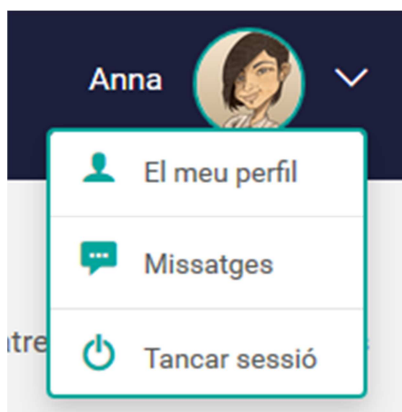
Etiqueta de categoria:



Elements de formulari:

Nom:	<input type="text" value="Nom"/>
Descripció:	<input type="text" value="Descripció"/>
Preu:	<input type="text" value="Preu"/> € <input type="radio"/> Preu/Hora <input type="radio"/> Preu/Servei
Categoria:	<input type="text" value="Reparacions"/> ▼
Zona:	<input type="text" value="Zona del servei"/> <input type="button" value="Troba'm"/>

Menú d'usuari:



Annex 7. Glossari

- **API.** Interfície de programació d'aplicacions. Representació del sistema de comunicació entre aplicacions o sistemes de programari.
- **Bug.** Terme que s'utilitza per a referir-se a errors de programari que desencadenen a un resultat indesitjat.
- **Framework.** Estructura conceptual i tecnològica que serveix per a la organització i desenvolupament de programari.
- **Git.** Control de versions. Recolzament sistemàtic per al manteniment i gestió d'arxius de codi font.
- **JSON.** Notació literal d'objectes Javascript. És un format per a l'intercanvi de dades.
- **MEAN Stack.** Infraestructura de l'aplicació. Formada per *MongoDB*, *ExpressJS*, *AngularJS* i *NodeJS*.
- **Middleware.** Programari que ajuda a una aplicació a interactuar o comunicar-se amb altres aplicacions, paquets de programes, xarxes, maquinari i/o sistemes operatius.
- **Mobile First** és un concepte que consisteix en dissenyar les interfícies pensant en els terminals mòbils i posteriorment adaptar-los a pantalles més grans.
- **MVC (Model-Vista-Controlador).** Patró d'arquitectura de programari que separa les dades i la lògica de negoci d'una aplicació de la interfície d'usuari i el mòdul encarregat de gestionar els esdeveniments i les comunicacions.
- **NoSQL.** Sistema de gestió de bases de dades no relacionals (no basades en relacions entre taules).
- **REST (Representational Estate Transfer).** Descriu qualsevol interfície entre sistemes que utilitzi directament HTTP per a obtenir dades o indicar la execució d'operacions sobre les dades.
- **UML (Unified Modeling Language).** Llenguatge per a representar i documentar sistemes informàtics.
- **Wireframe.** Representació esquemàtica d'una pàgina web sense elements gràfics, on es representa l'arquitectura d'informació.

Annex 8. Bibliografia

- **Hassan, Y.** (2009). *Diseño Centrado en el Usuario (DCU)*. [online] NoSoloUsabilidad. Obtingut de: <http://www.nosolousabilidad.com/manual/3.htm>
- **Henry, S.** (2005). *Web Content Accessibility Guidelines (WCAG) Overview*. [online] W3C. Obtingut de: <https://www.w3.org/WAI/intro/wcag.php>
- **Hom, J.** (1996). *Evaluación Heurística (Heuristic evaluation)*. [online] Sidar. Obtingut de: <http://www.sidar.org/recur/desdi/traduc/es/visitable/inspeccion/Heur.htm>
- **Nielsen, J.** (1994) *Enhancing the explanatory power of usability heuristics*. Proceedings on the ACM CHI'94 Conference (pàgines 152-158)
- **Ornbo, G.** (2013) *Programación NODE.JS*. Madrid: Anaya Multimedia
- **Sevilleja, C.** (2013) *Easy Node Authentication: Facebook*. [online] Scotch.io. Obtingut de: <https://scotch.io/tutorials/easy-node-authentication-facebook>
- *Getting Started on Heroku with Node.js* [online] Heroku. Obtingut de: <https://devcenter.heroku.com/articles/getting-started-with-nodejs#introduction>
- *Google Maps Geocoding API*. [online] Google. Obtingut de: <https://developers.google.com/maps/documentation/geocoding/intro#Geocoding>

Recursos

També s'han utilitzat diverses fonts de recursos, entre elles les principals:

- **Platzi**. Pàgina de videotutorials frontend i backend <https://platzi.com/>
- **Devcode**. Pàgina de videotutorials per a desenvolupadors <http://devcode.la>
- **Udemy**. Pàgina de cursos online de temes diversos <https://www.udemy.com/>
- **Stackoverflow**. Forum de desenvolupadors <http://stackoverflow.com>
- **AngularJS**. Documentació Angular <https://angularjs.org/>
- **MongoDB**. Documentació de MongoDB <https://docs.mongodb.com/>
- **Bootstrap**. Documentació de Bootstrap <http://getbootstrap.com/>
- **NPM**. Documentació de paquets de Node Package Manager <https://www.npmjs.com/>
- **W3C**. Referència sobre HTML, CSS, Javascript, etc <http://www.w3schools.com/>
- **Stock Snap**. Repositori d'imatges gratuït i lliure de drets, utilitzat per a les imatges de perfil dels usuaris ficticis de la plataforma <https://stocksnap.io/>
- **Icomoon**. Repositori d'icones <https://icomoon.io/>