

NEARBY JOBS - APLICACIÓ WEB D'OFERTES DE FEINA PER PROXIMITAT

Nom Estudiant (Autor): DANIEL COMAS FERNÁNDEZ

Grau d'enginyeria informàtica

TFG - Desenvolupament Web

Nom Consultor/a: GREGORIO ROBLES MARTÍNEZ

Nom Professor/a responsable de l'assignatura: SANTI CABALLE LLOBET

Data Lliurament: 12/06/2020



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

FITXA DEL TREBALL FINAL

Títol del treball:	<i>NearBy Jobs - Aplicació web d'ofertes de feina per proximitat</i>
Nom de l'autor:	<i>Daniel Comas Fernández</i>
Nom del consultor/a:	<i>Gregorio Robles Martínez</i>
Nom del PRA:	<i>Santi Caballe Llobet</i>
Data de lliurament (mm/aaaa):	<i>06/2020</i>
Titulació o programa:	<i>Grau d'Enginyeria Informàtica</i>
Àrea del Treball Final:	<i>TFG - Desenvolupament Web</i>
Idioma del treball:	<i>Català</i>
Paraules clau	<i>Ofertes, feina, proximitat</i>
Resum del Treball (màxim 250 paraules): <i>Amb la finalitat, context d'aplicació, metodologia, resultats i conclusions del treball</i>	
<p>La finalitat del projecte consisteix en obtenir una aplicació web que permeti als usuaris consultar ofertes de feina per proximitat i a les companyies cercar professionals. Aquesta ha d'estar composta per un portal web el qual obté i envia les dades a partir d'uns serveis web i on l'autorització d'aquests es realitza a partir de JWT tokens donada la seva característica d'operar en entorns distribuïts.</p> <p>Conscient del temps i dels recursos disposats el lector podrà veure la intenció des del primer moment de mostrar el coneixement i posar en pràctica el conjunt de tècniques apreses durant el grau, en diferents fases i de forma àmplia, com la planificació, pla de riscos, definició de requisits, dissenys de components, casos d'ús i prototipatges entre d'altres.</p> <p>Durant la implementació, es pren consciència de la complexitat d'aquest tipus de projectes i de la importància que tenen tot els processos previs d'anàlisis per minimitzar l'impacte dels canvis quan es produeixen.</p> <p>Finalment, s'obté l'aplicació proposada i s'indiquen tasques perquè es pugui continuar amb el projecte.</p>	

Abstract (in English, 250 words or less):

The project's purpose resides in to obtain a web application to allow users to look for job opportunities nearby and companies for professionals. In addition, it must be composed of a web site and a set of an authorized web services to send and receive data. The authorization must be made with JWT token due its capacity to operate in distributed Systems.

Aware of the time and resources available the reader will be able to see, from the first moment, the intention to show the knowledge and to use specific techniques learned in this study level, in different project stages and in extensively way such as planning, risk plan, requirements definition, component designs, use cases and page prototyping among other things.

During the software implementation, we become aware of the complexity of this type of projects and how important are the initial analytical processes that help to minimize the impact of changes when they occur.

Finally, the application proposed is obtained and some next steps are registered to be able to continue with the project.

Agraeixo a tots els que han fet possible que aquest projecte s'hagi pogut realitzar tal i com s'ha fet. Aquest, és el resultat d'un llarg recorregut formatiu on en cada moment sempre hi ha hagut algú que ha aportat el seu gra de sorra i que m'ha ajudat tant en aconseguir els coneixements com la determinació per fer-ho. Una mica de cada un d'ells està present aquí i desitjo que us ajudi d'una forma o altre a vosaltres també. Gràcies.

Índex

1. Títol del treball	1
2. Descripció breu.....	1
3. Objectius	1
3.1. Principals.....	1
3.2. Secundaris	1
4. Tecnologies que s'utilitzen	1
5. Planificació temporal	2
6. Avaluació de riscos	4
7. Requisits del projecte	6
8. Disseny del projecte	6
8.1. Esquema invariant punt de vista de la informació	6
8.2. Diagrama de casos d'ús	8
8.3. Estructura global del projecte	9
8.4. Diagrama de components.....	9
8.5. User Journey	11
8.6. Mockups / Prototipatge	12
8.6.1. Login.....	12
8.6.2. User Register	13
8.6.3. Company Register	14
8.6.4. User Home.....	15
8.6.5. User Home - Show job	16
8.6.6. Company Home.....	17
8.6.7. Company Jobs.....	18
8.6.8. Company Jobs – New Job	19
9. El frontend	20
9.1. Estudi del <i>framework</i>	20
9.2. Punts específics per aquest projecte	21
9.3. Pàgines.....	25
9.3.1. Login.....	25
9.3.2. User Register	26
9.3.3. Company Register	29
9.3.4. User Home.....	30
9.3.5. User Home - Show job	31
9.3.6. Company Home.....	32
9.3.7. Company Jobs.....	33
9.3.8. Company Jobs – New Job	34
10. El servidor.....	36
10.1. Base de dades	36
10.2. Els serveis web	38
10.2.1. Autenticació i autorització d'usuaris.....	38
10.2.2. Logout	40
10.2.3. POST /oauth/token	41
10.2.4. POST /oauth/token/revoke	42
10.2.5. GET /users/me.....	43

10.2.6. GET /professions	44
10.2.7. POST /users	45
10.2.8. POST /users/:id/jobs.....	46
10.2.9. DELETE /users/:id/jobs/:jobid.....	47
10.2.10. GET /users/:id/jobs	47
10.2.11. GET /users/jobs	48
10.2.12. GET /users	49
11. Tasques següents	51
12. Impacte dels riscos.....	51
13. Resultats i conclusions.....	52
14. Glossari.....	53

1. Títol del treball

Nearby Jobs - Aplicació web d'ofertes de feina per proximitat

2. Descripció breu

El projecte consisteix en una plataforma web que permet a usuaris cercar ofertes de feina per proximitat a través de coordenades gps i les companyies poden fer el mateix però per cercar professionals propers.

3. Objectius

3.1. Principals

- Obtenir una aplicació web que permeti als usuaris consultar ofertes de feina per proximitat i a les companyies cercar professionals. L'aplicació aconseguida consisteix en un portal el qual obté les dades a partir d'uns serveis web i on l'autorització d'aquests serveis es fan fa a partir de JWT tokens els quals permeten treballar de forma distribuïda.

3.2. Secundaris

- Documentar i realitzar les fases d'un projecte de desenvolupament web
- Posar en pràctica els coneixements adquirits de les assignatures del grau

4. Tecnologies que s'utilitzen













Frontend: Angular 9





Base de dades: PostgreSQL

Servidor: Serveis web amb nodejs, express

Programes: MS Project per la planificació, MagicDraw pels diagrames, VS Code per la programació de codi

5. Planificació temporal

Id		Nombre de tarea	Duración	Comienzo	Fin	Predecesoras
1		Primera fita	28 días?	lun 09/03/20	vie 10/04/20	
2		Diagrama vista de informació	2 días	lun 09/03/20	mar 10/03/20	
3		Diagrama components frontend i serveis web	2 días	lun 09/03/20	mar 10/03/20	
4		Casos d'ús	4 días	mié 11/03/20	lun 16/03/20	
5		Mockups/Prototipatge	4 días	mié 18/03/20	dom 22/03/20	
6		Frontend	15 días?	lun 23/03/20	mié 08/04/20	
7		Creació plantilla	1 día	lun 23/03/20	lun 23/03/20	
8		Login + serveis per connectar als web services	2 días	mar 24/03/20	mié 25/03/20	7
9		Registre usuari + serveis per connectar als web services	2 días?	jue 26/03/20	vie 27/03/20	8
10		Registre empresa + serveis per connectar als web services	2 días	sáb 28/03/20	lun 30/03/20	9
11		Home usuari + serveis per connectar als web services	2 días	mar 31/03/20	mié 01/04/20	10
12		Home empresa + serveis per connectar als web services	2 días	jue 02/04/20	vie 03/04/20	11
13		Empresa + serveis per connectar als web services	2 días	sáb 04/04/20	lun 06/04/20	12
14		Empresa - Ofertes de feina + serveis per connectar als web services	1 día?	mié 08/04/20	mié 08/04/20	13
15		Memòria final - Documentar part realitzada	28 días?	lun 09/03/20	vie 10/04/20	
16		Segona fita	39 días?	lun 13/04/20	vie 29/05/20	
17		Crear base de dades	3 días	lun 13/04/20	mié 15/04/20	
18		Oauth - Token request	1 día?	jue 16/04/20	jue 16/04/20	17
19		Autorització dels web services	1 día?	jue 16/04/20	jue 16/04/20	
20		Web services - Obtener usuari autenticat	1 día?	vie 17/04/20	vie 17/04/20	19
21		Web services - Crear usuaris	3 días	sáb 18/04/20	lun 20/04/20	20
22		Web services - Crear empreses	3 días	mar 21/04/20	jue 23/04/20	21
23		Web services - Crear ofertes de feina	3 días	vie 24/04/20	dom 26/04/20	22
24		Web services - Llistat d'ofertes per empresa	3 días	lun 27/04/20	mié 29/04/20	23

Id		Nombre de tarea	Duración	Comienzo	Fin	Predecesoras
25		Web services - Llistat d'ofertes a partir de la professió	3 días	jue 30/04/20	lun 04/05/20	24
26		Web services - Llistat de professionals (usuaris) a partir de la professió	3 días	mar 05/05/20	jue 07/05/20	25
27		Conectar Frontend amb els web services realitzats	17 días	jue 07/05/20	vie 29/05/20	
28		Memòria final - Documentar part realitzada	39 días?	lun 13/04/20	vie 29/05/20	
29		Memòria final - Documentar parts restants i resultats finals	10 días	lun 01/06/20	vie 12/06/20	

6. Avaluació de riscos

Codi	Nom	Causa	Descripció	Conseqüència	Probabilitat	Impacte	Nivell
R01	Instal·lació i configuració dels programes essencials pel desenvolupament del projecte	Es trigui més temps del compte	De vegades, les instal·lacions i les configuracions dels programaris poder no ser trivials i requereixen un temps que es pot sobrepassar del previst	Endarreriment del començament de les fases	Mitjana	Mitjà	Mitjà
R02	Pèrdua de fitxers o de la última sessió	Més feina	Si es perden fitxers que és el principal valor pel projecte caldrà tornar a realitzar el codi perdut	No finalitzar el projecte o tindre que tornar a fer codi ja realitzat en temps rècord	Mitjana	Alt	Alt
R03	Canvis de requisits	Més feina, tornar a analitzar, i dissenyar i canviar codi	De vegades, durant el projecte i a tot projecte de software es pot donar el cas que s'ha de canviar els requisits	Adaptar el codi existent a aquests canvis	Mitjana	Mitjà	Mitjà
R04	Temps de realització d'una tasca més alts	Es trigui més temps del compte	-	Endarreriment de les tasques següents	Mitjana	Mitjà	Mitjà

Mesures correctores

Codi	Acció	Tipus	Risc Residual	Data límit
R01-A1	Instal·lació del programari abans de començar el projecte	Mitigadora	Baix	Fins la fase 2 inclosa
R02-A1	Utilitzar repositoris de projecte al núvol o copiar fitxers en algun servei cloud. Sincronitzar al final de cada sessió	Mitigadora	Baix	Fins a la fase 2 inclosa
R03-A1	Analitzar els canvis a realitzar quan es produeixin i canviar els temps en la planificació	Mitigadora	Baix	Durant tot el projecte
R04-A1	Canviar els temps en la planificació	Mitigadora	Baix	Durant tot el projecte

7. Requisits del projecte

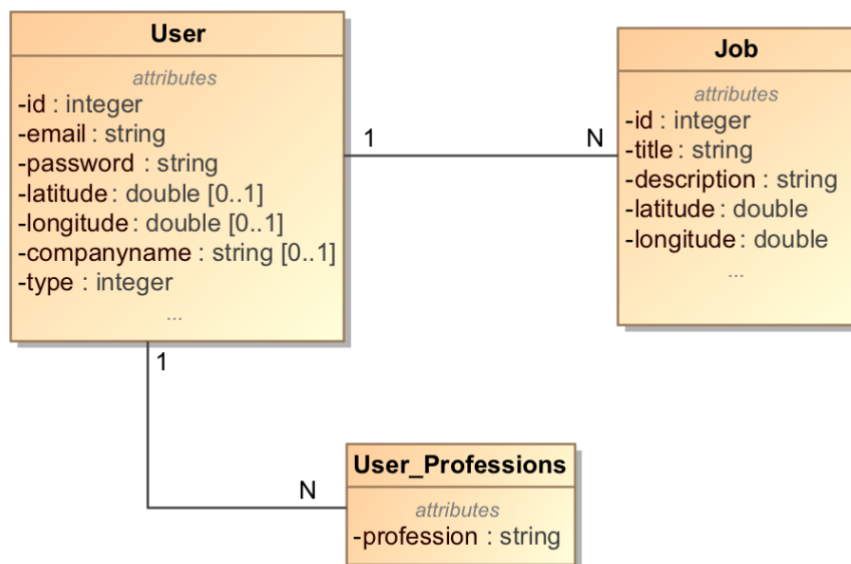
Anotem els requisits que l'aplicació ha de complir:

- S'ha de poder accedir des del navegador
- Han de poder entrar dos tipus d'usuaris, l'usuari estàndard com el de tipus empresa
- S'ha d'utilitzar autenticació i autorització d'usuaris perquè aquests puguin el realitzar processos de forma segura
- Els usuaris estàndard han de poder:
 - o Registrar-se al portal indicant el email, contrasenya, les diferents professions en les que s'ha especialitzat i una localització gps que és la posició o zona propera per on li agradaria trobar una feina
 - o Entrar al portal
 - o Cercar ofertes de treball a partir d'una zona delimitada per un mapa i una professió
 - o Veure el detall d'una oferta de treball seleccionada
 - o Poder desconnectar la sessió amb seguretat
- Els usuaris empresa han de poder:
 - o Registrar-se al portal indicant nom d'empresa, email i contrasenya per entrar
 - o Cercar professionals a partir d'una zona delimitada per un mapa i una professió
 - o Donar d'alta ofertes de treball de l'empresa, indicant la professió, una descripció i una posició gps de la ubicació de la feina
 - o Veure ofertes donades d'alta per l'empresa
 - o Eliminar ofertes de treball de l'empresa
 - o Poder desconnectar la sessió amb seguretat

8. Disseny del projecte

8.1. Esquema invariant punt de vista de la informació

El primer pas, és el de realitzar l'esquema invariant del punt de vista de la informació, segons el model de referència pel processament obert i distribuït (RM-ODP) el qual ens permet representar com es guarden les dades i com es relacionen entre elles.

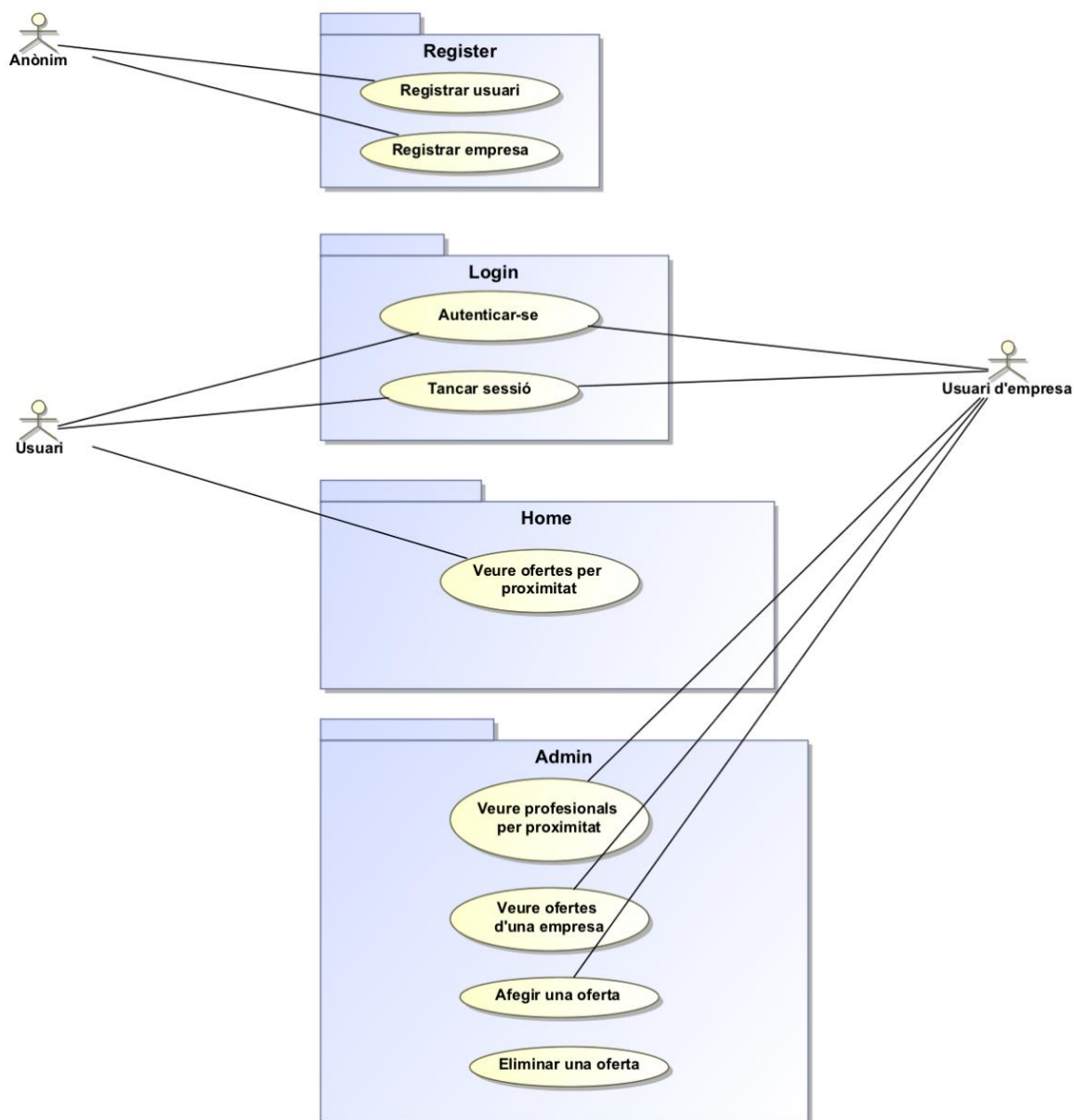


Com veiem, tenim tres entitats:

- User s'ha modelat amb els camps email, password i companyname com String. Id i type com enters i latitude i longitude com a double. Tan latitude com longitude i companyname permeten nulls. Aquesta classe està relacionada amb N User_Professions i N Job. A més a més, s'ha decidit que tan els usuaris d'empresa com els estàndard estiguin centralitzats en un mateix punt per tal d'estalviar consultar a altres entitats quan es vulgui realitzar determinades tasques com validacions de login o canvis de password, és a dir, menys objectes a gestionar per tal de reduir complexitat.
- User_Professions s'ha modelat amb un camp profession com a String
- Job s'ha modelat amb un camp id com a enter, tan title (professió) i description com a string i latitude i longitude com a double

8.2. Diagrama de casos d'ús

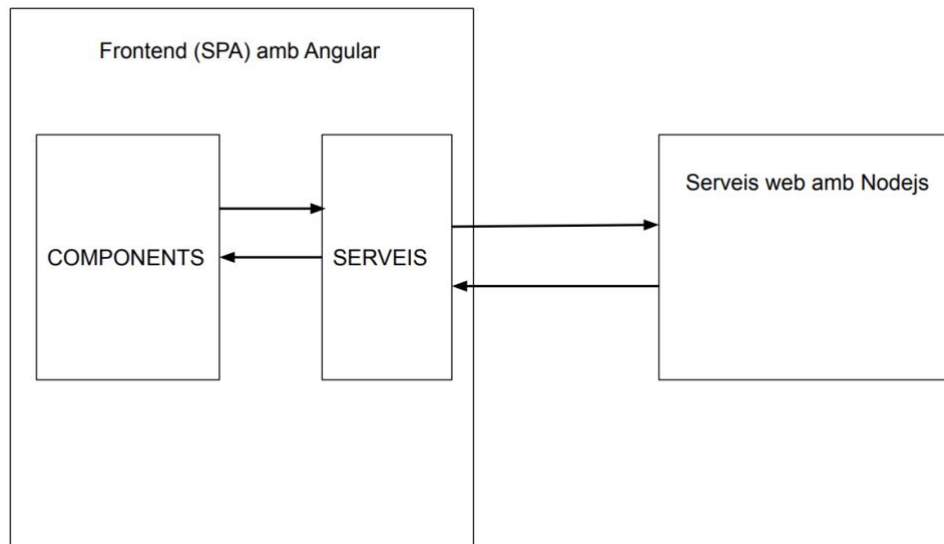
Un altre pas ahora de dissenyar el projecte és organitzar per paquets les accions que cada rol fa, d'aquesta manera ens podem fer una idea de quins components pot requerir l'aplicació. El diagrama que ens permet això, és el diagrama de casos d'ús:



8.3. Estructura global del projecte

El projecte a nivell estructural estarà dividit en dos subprojectes. Per una banda, un projecte *frontend*, encarregat de presentar i permetre que l'usuari pugui interaccionar (part visible) i per l'altre, el servidor, encarregat de la d'obtenció i processament de la informació (part no visible).

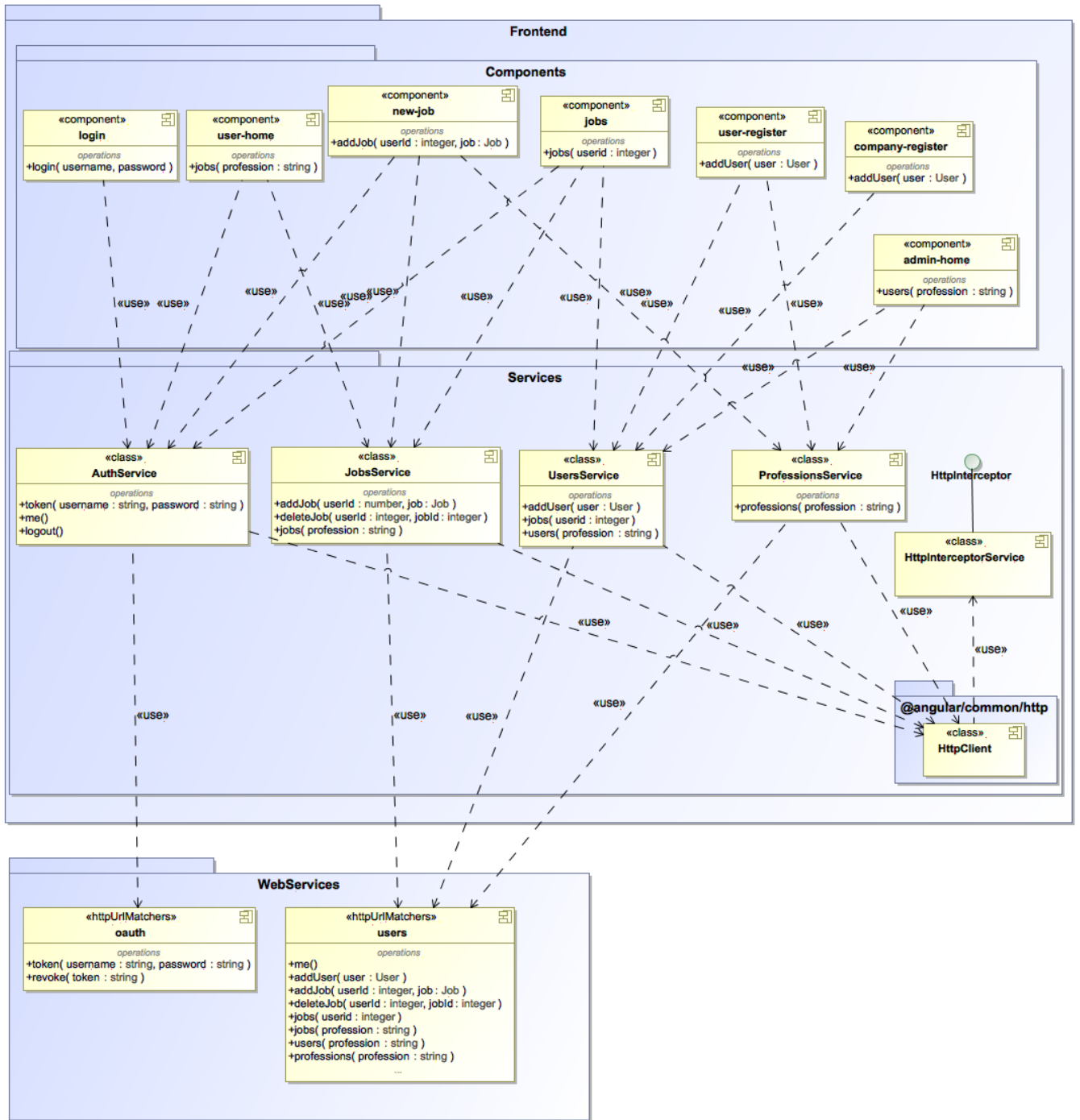
El següent gràfic il·lustra de quina forma es comuniquen:



Comunicacions entre els subprojectes

8.4. Diagrama de components

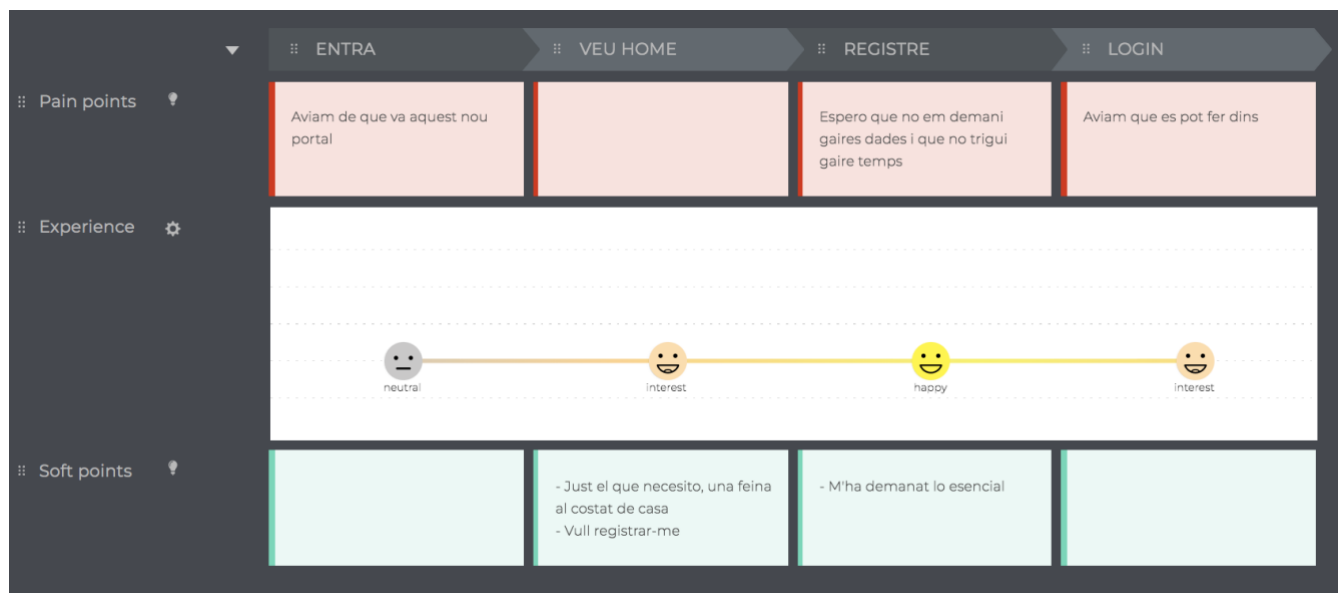
Vista una visió general, ha arribat el moment de decidir els objectes i components en que estarà compost tot el projecte i com s'han de relacionar per tal de reduir les complexitats del codi i l'acoblament. Un diagrama que ens ajuda a representar aquests objectes, components, les seves operacions i les relacions entre ells és el diagrama de components:



8.5. User Journey

Un cop especificat a nivell tècnic com es farà la implementació del codi. És el moment de centrar-nos en com presentem la informació, és per això que usem la tècnica de l'usuari journey. Aquesta tècnica ens serveix per extreure informació de quines emocions poden tenir els usuaris al veure pàgines del nostre web i que ens pot ajudar a encarar els dissenys.

En aquest cas, s'elabora el que creiem que és el més destacat, és la pàgina principal i el registre, és a dir el punt d'entrada. Partim de la base que un usuari anònim entra per primer cop i es registra:



Captem els pensaments que pot tindre i extrèiem les següents conclusions per elaborar com han de ser aquestes dues pàgines:

- A la primera pàgina que s'entra ha d'informar clarament el que fa l'aplicació
- La pàgina de registre ha de tindre el camps essencials, un cop l'usuari és dins podem demanar que empleni més informació

8.6. Mockups / Prototipatge

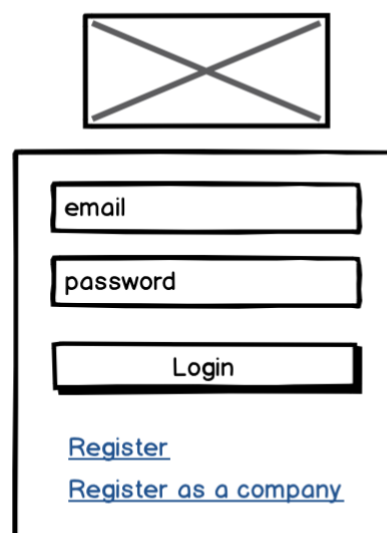
Continuem amb el disseny de pàgines, ara amb la realització de prototipatges o mockups. Aquesta és una tècnica que ens ajuda a prendre decisions en relació als camps i controls a utilitzar i quins criteris usem per a distribuir-los.

En aquest projecte s'usen de forma destacada els formularis i es segueix un criteri d'ordenació per importància.

8.6.1. Login

En el cas de la pàgina de login, es divideix en dos seccions, la part esquerra on es mostra informació clara del que es pot fer a la web i a la dreta el formulari per entrar a l'aplicació, amb el logo a sobre que serveix d'identificació i dos enllaços, un per anar a la pàgina de registre d'usuari i l'altre pel de l'empresa.

- **Select a location where you want to work**
- **Select your profession**
- **Job found!**



The mockup shows a login form with the following elements:

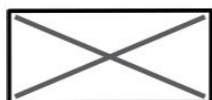
- A logo consisting of a rectangle with an 'X' inside, positioned at the top.
- An input field labeled "email".
- An input field labeled "password".
- A "Login" button.
- Two links: "Register" and "Register as a company", both in blue text.

8.6.2. User Register

Pel registre d'usuari, tenim un formulari amb els camps distribuïts de major a menor importància, tot i que tots són requerits. Per una banda, les dades d'accés al portal, com el correu electrònic i la contrasenya, i per l'altre, les diferents professions i el mapa per seleccionar la posició d'on es vol treballar, amb un text informatiu per millorar la comprensió.

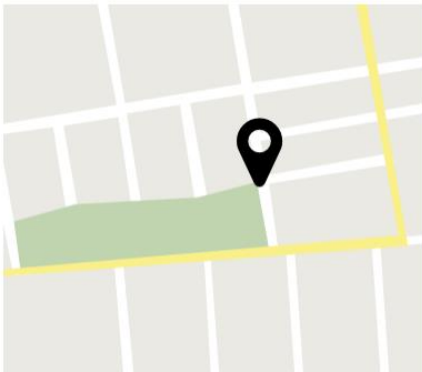
Cal afegir, que el control de professions usat, és un control anomenat d'etiquetes o *tags* en anglès que ens permet afegir-ne més d'una i se li pot donar la capacitat d'auto emplenar. És a dir, el fem servir per seleccionar professions que s'hagin afegit anteriorment en el sistema mentre que estem escrivint. En canvi, sinó hi és, queda escrit i s'afegirà dins del sistema un cop registrats.

El logo queda posicionat a dalt de tot com a identificació i per poder tornar a la pàgina de login:



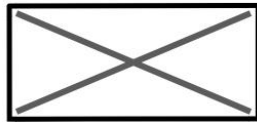
User register

Professions

Mark on the map a position
where you would like to
work or nearby


8.6.3. Company Register

L'altre pàgina de registre, la de registre d'empresa, manté l'estructura i la distribució de camps per importància com l'anterior d'usuari, però en aquest cas el contingut del formulari es diferent, es requereixen menys dades, com són: el nom de l'empresa i les credencials per a entrar a l'aplicació, aquestes últimes amb un text informatiu per indicar perquè es demanen.

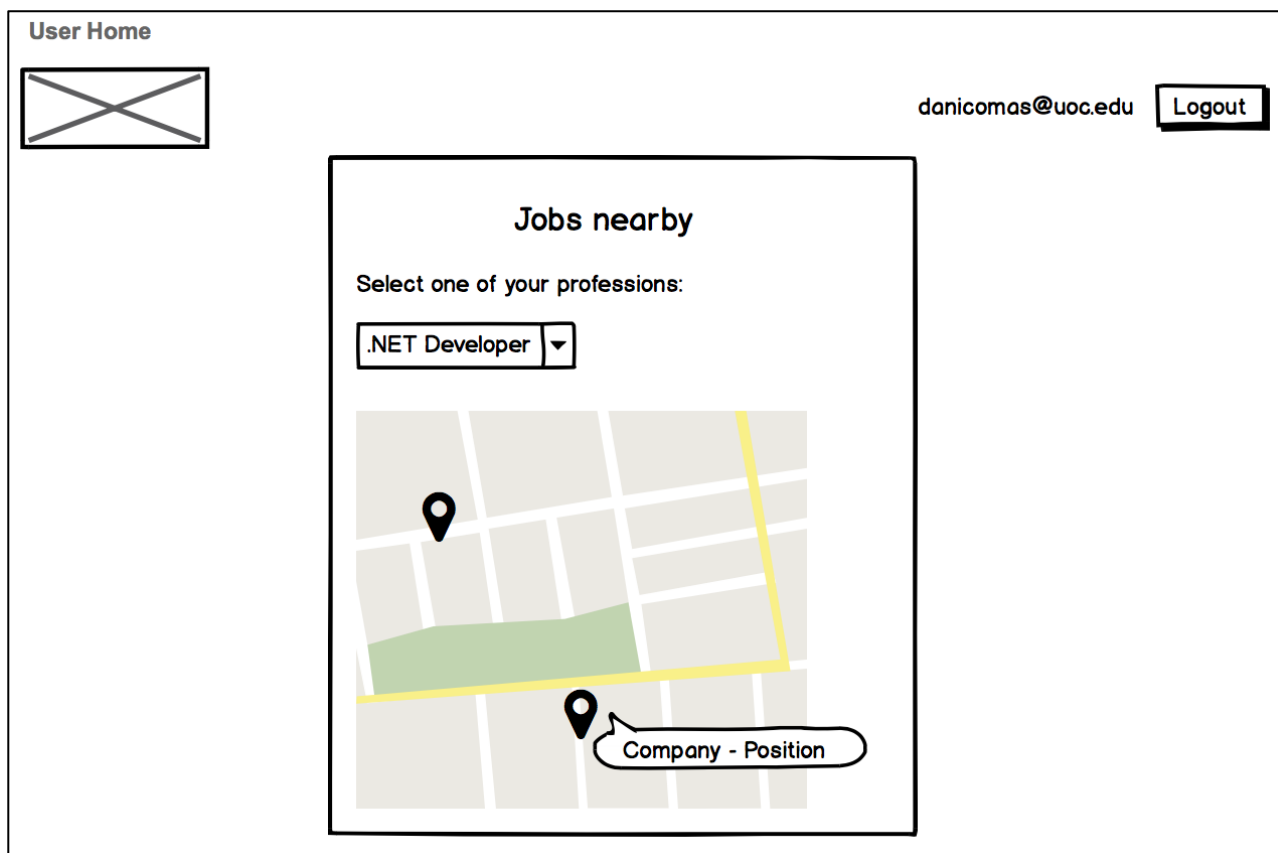


Company register

Create an user who will be the administrator of the company:

8.6.4. User Home

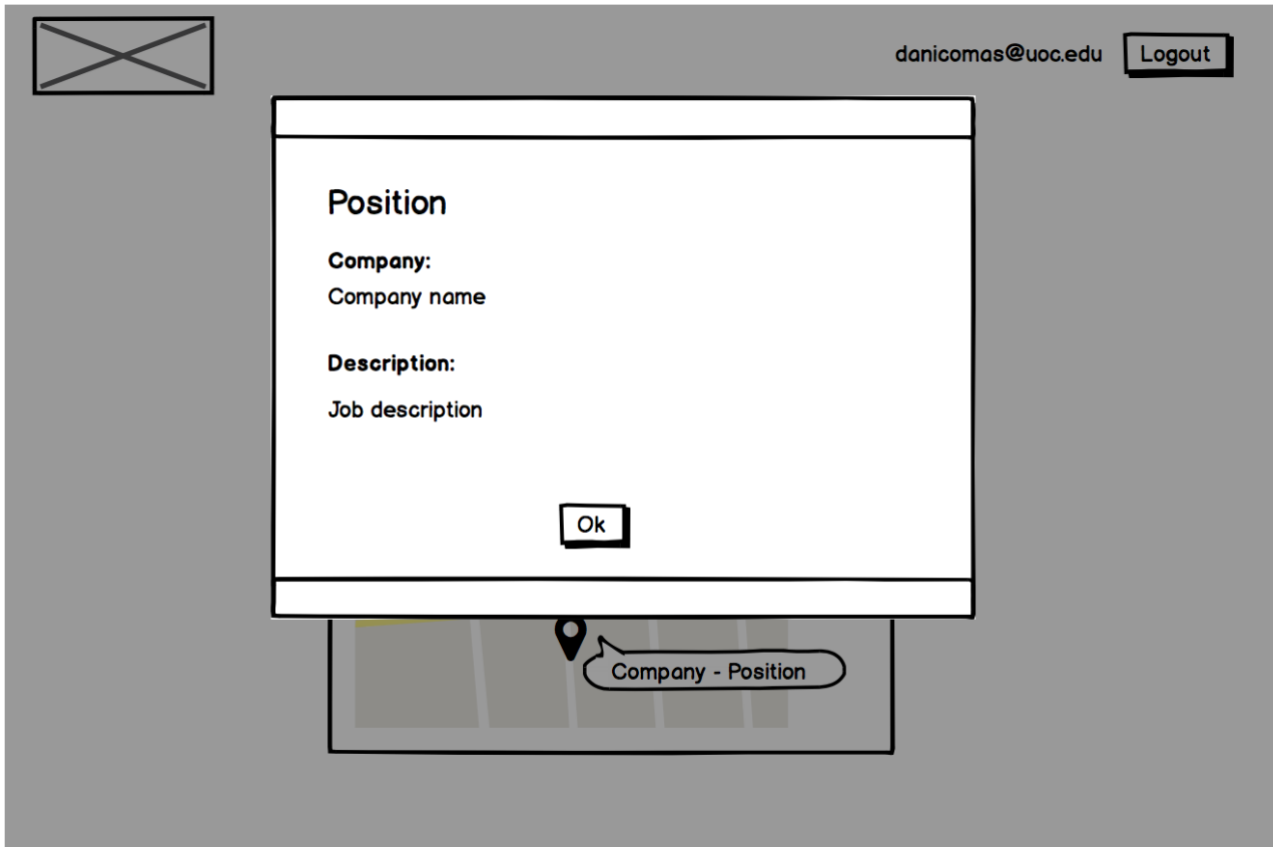
Ara, treballem amb les pàgines un cop l'usuari s'ha validat en el sistema. La primera que veu és la pàgina principal o home. Aquesta conté -a part de la capçalera amb el logo, l'usuari validat i un botó de tancar sessió que és una màxima en totes les pàgines internes- el títol de la pàgina que serveix per posicionar-nos i per últim, el mapa i el camp per cercar feines per professió.



Cal destacar, que s'ha afegit un text informatiu com indicador que s'ha de seleccionar alguna professió del desplegable abans de que apareguin ofertes de feina en el mapa. Si fem clic en una d'elles, se'ns obrirà la pàgina modal que es mostra en el següent apartat.

8.6.5. User Home - Show job

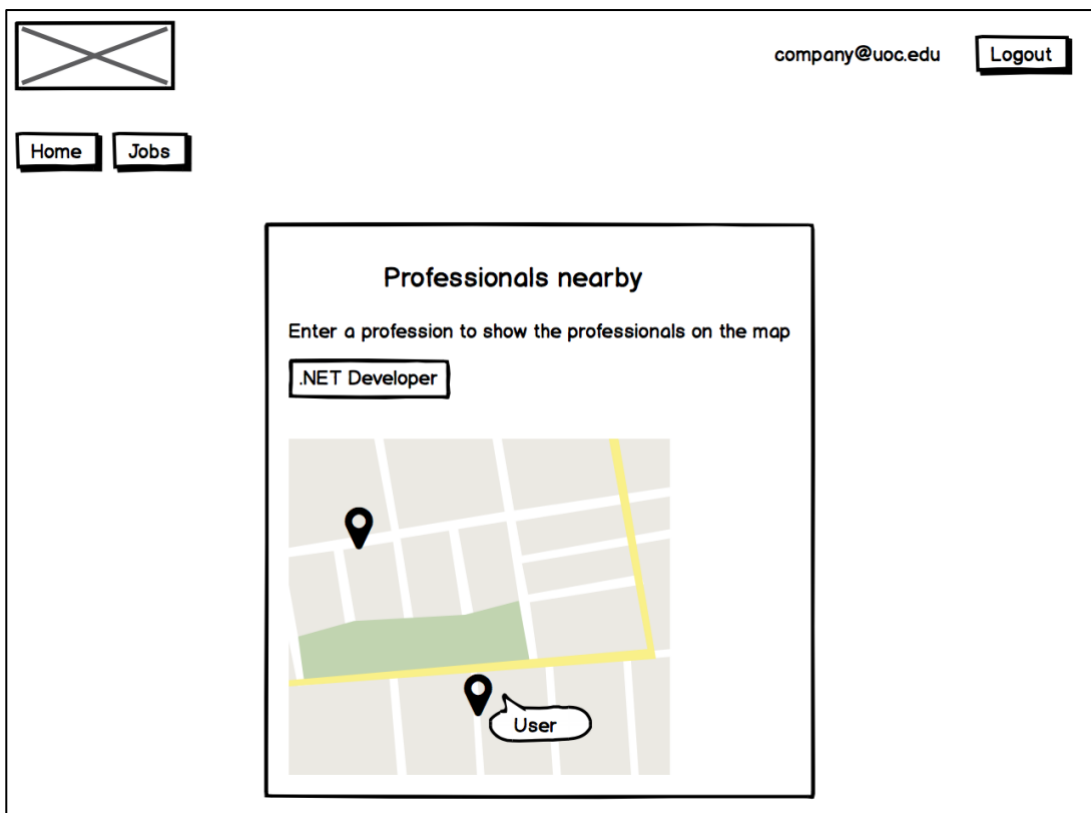
Un cop s'ha seleccionat una oferta en el mapa, s'obre aquest modal que conté: el títol de la oferta, l'empresa que la ofereix i una descripció.



8.6.6. Company Home

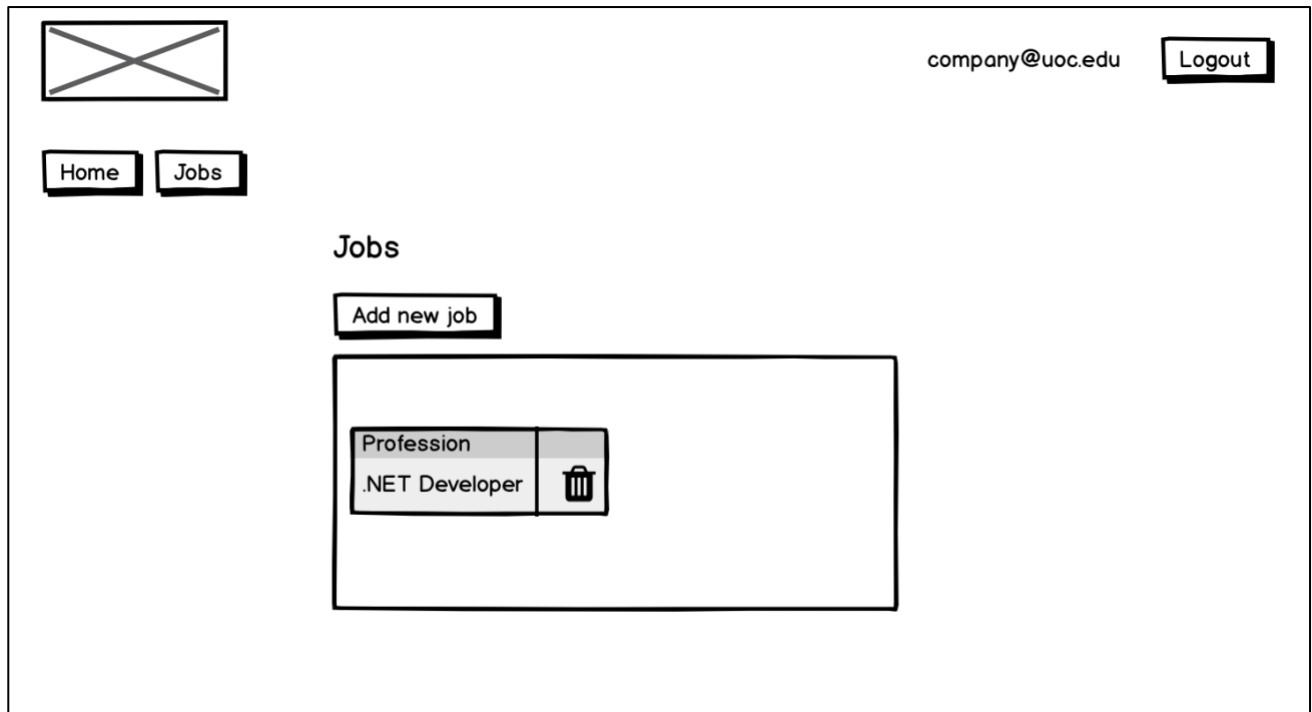
Quan un usuari es valida a l'aplicació, va a parar a una pàgina principal, hem vist la que s'accedeix quan no és del tipus empresa, ara ens queda explicar exactament aquesta. En aquest cas, el contingut que veurà un cop entrat és la cerca de professionals propers mitjançant la selecció d'una professió.

Cal afegir que donat que l'empresa tindrà l'opció d'accedir a un altre pàgina -la gestió d'ofertes- s'afegeix un menú de navegació.



8.6.7. Company Jobs

A l'entrar a la pàgina de gestió d'ofertes de l'empresa, veiem el llistat de totes les ofertes que s'han registrat anteriorment i amb l'opció d'eliminar-les o donar d'alta de noves.



Al prémer el botó d'afegir nova oferta s'obrirà un modal que s'explica en el següent apartat.

8.6.8. Company Jobs – New Job

En el modal per afegir una nova oferta de treball, l'empresa indica el tipus de professional que s'està buscant, una descripció i posiciona en el mapa on es realitzarà la feina:

The image shows a web application interface for adding a new job. It features a modal window titled "New Job" with the following elements:

- A "Profession" text input field.
- A "Description" text area.
- A map showing a street grid with a yellow highlighted path and a black location pin.
- An "Add" button located below the map.

The modal is displayed on a grey background. In the top right corner, there is a user email "company@uoc.edu" and a "Logout" button. In the top left corner, there is a placeholder box with an 'X' and two navigation buttons labeled "Home" and "Jobs".

Arribats en aquest punt, ja coneixem com seran totes les pàgines. No obstant, no és de més comentar que en una aplicació en el món real continuàrem amb el procés de disseny, fent prototipatges d'alt nivell i realitzant tests amb diferents usuaris per provar la interfície. Nosaltres, però, ara continuarem amb la definició de l'estructura del frontend, que és el sub projecte que englobarà totes aquestes pàgines.

9. El frontend

Tècnicament, el frontend consistirà en una sola pàgina on dins d'aquesta es carregaran totes les que acabem de dissenyar. El motiu de fer-ho d'aquesta forma és que es millora la interacció amb l'usuari i el processament del servidor. Aquest concepte d'una sola pàgina, s'anomena *Single Page Application (SPA)*.

Per tal efecte, s'ha decidit que la tecnologia a utilitzar és el conjunt de llibreries o *framework* d'Angular, concretament la versió 9. El qual, a partir del llenguatge de programació Microsoft Typescript ens permet implementar el codi seguint un dels conceptes essencials de la programació, la orientació a objectes.

No hi ha que deixar de banda el Angular CLI, el qual conté un conjunt de comandes que ens ajuden a realitzar nombroses tasques, com per exemple, selecció de plantilles predefinides, creació de classes, tasques d'execució o publicació entre d'altres.

Per tancar el conjunt tecnològic a utilitzar en aquest frontend, cal anomenar l'editor de codi seleccionat, l'IDE Microsoft Visual Studio Code en aquest cas, que ens aporta una bona integració amb aquest *framework*.

Ara continuem amb el coneixement obtingut sobre l'estudi del *framework*, ja que no es pot començar a programar sense conèixer els elements bàsics i com funciona.

9.1. Estudi del *framework*

En un projecte amb el *framework* d'Angular trobem els següents elements i conceptes bàsics:

- Un gestor de mòduls que gestiona les llibreries que s'utilitzen en el codi. Webpack o systemjs en són un d'aquests
- Components: Formats per un html, estils i codi de programació. En el codi del component se'l hi referencia un selector que no és més que un element html on aquest es carregarà. Per entendre-ho millor, una pàgina web pot està formada per varis elements, com per exemple: `<div></div>`, `` o també ens en podem inventar de nous com `<prova></prova>`. Si referenciem a `<prova>`, el component es carregarà dins d'aquest.
- Concepte d'arbre de components

Lligant amb l'element anterior, cal entendre que Angular es basa amb el concepte o jerarquia d'arbre. És a dir, existeix un node superior on d'aquest pengen o ramifiquen altres nodes. Es carrega per tant, un primer component, que en molts casos s'anomena `app.component` i després els seus fills.

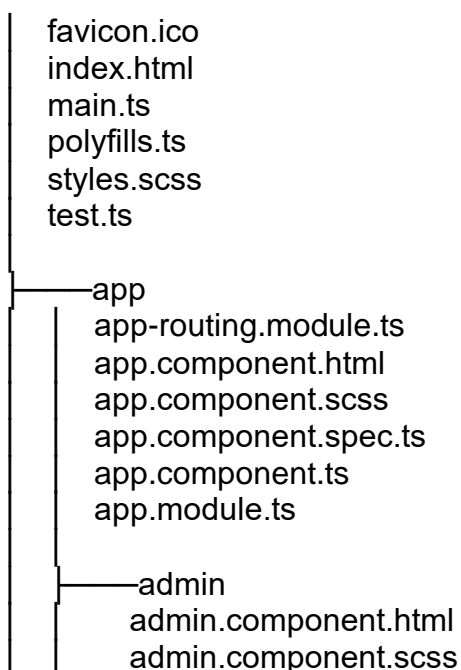
- Un fitxer index.html on es carregarà el primer component, el component arrel
- Serveis: Són classes(1*) que contenen certa lògica com per exemple peticions al servidor. Utilitzen el patró singleton (2*).
- Models: Són classes. Es poden utilitzar per tipar (3*) les respostes que venen del servidor.
- Mòduls: És una forma d'encapsular un conjunt de components, serveis o altres mòduls. Normalment, pel mòdul inicial s'utilitza el nom app.module.ts
- Main.ts: És el primer fitxer en executar-se, abans de que es formi l'arbre de components. Serveix de configuració (estableix entorns de desenvolupament o de producció) i executa el primer mòdul (L'AppModule).

Apresos els elements i conceptes bàsics del *framework*, toca el torn d'adaptar-lo al projecte que es vol realitzar.

9.2. Punts específics per aquest projecte

En aquest projecte s'afegeix la llibreria *Angular Material* la qual conté nombrosos components seguint el concepte de disseny *Google Material Design*. En aquesta, trobem: camps de text i d'etiquetes, desplegable, llistats, funcionalitats d'auto emplenar i drag & drop(4*) i també ens permet editar els seus temes i plantilles.

En aquest punt toca realitzar l'estructura del projecte sempre seguint les especificacions definides en punts anteriors. Organitzem tots els fitxers i pàgines que formen el projecte frontend de la següent manera:



admin.component.spec.ts
admin.component.ts

admin-home
admin-home.component.html
admin-home.component.scss
admin-home.component.spec.ts
admin-home.component.ts

auth
auth.component.html
auth.component.scss
auth.component.spec.ts
auth.component.ts

company-register
company-register.component.html
company-register.component.scss
company-register.component.spec.ts
company-register.component.ts

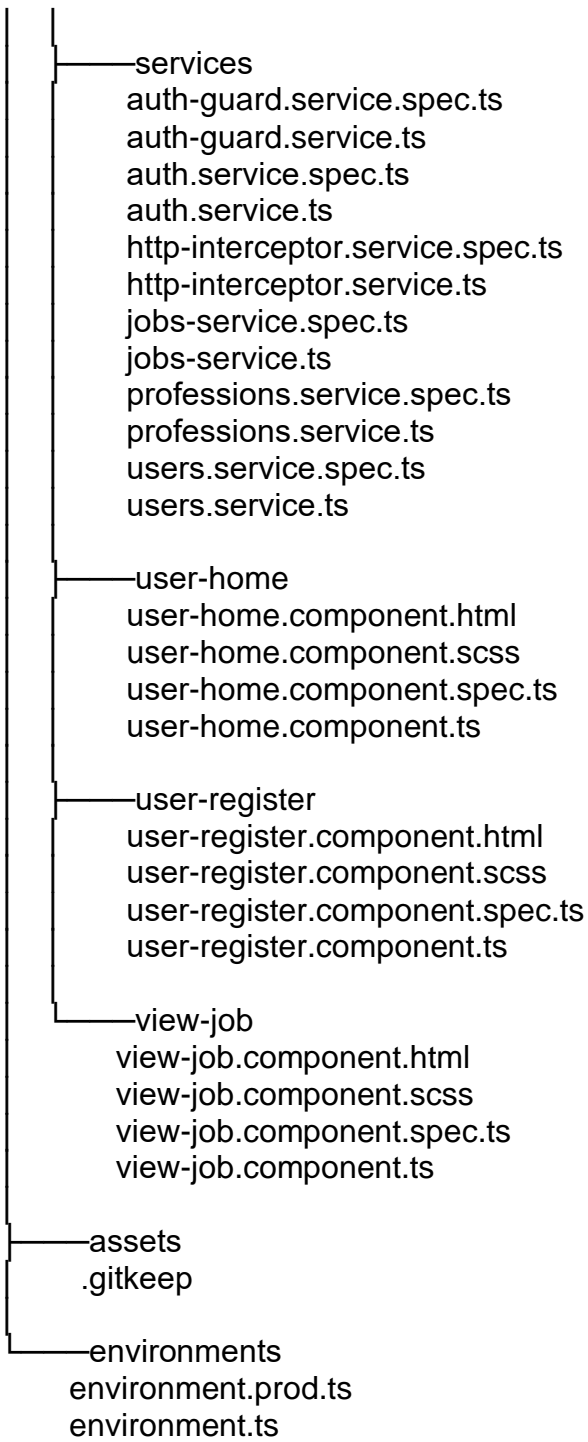
jobs
jobs.component.html
jobs.component.scss
jobs.component.spec.ts
jobs.component.ts

login
login.component.html
login.component.scss
login.component.spec.ts
login.component.ts

logo
logo.component.html
logo.component.scss
logo.component.spec.ts
logo.component.ts

models
job.model.spec.ts
job.model.ts
user.model.spec.ts
user.model.ts

new-job
new-job.component.html
new-job.component.scss
new-job.component.spec.ts
new-job.component.ts



Un altre punt específic que s'ha afegit és el d'enrutament. Aquest serveix per enllaçar les rutes del navegador amb un component. Posant un exemple s'entendrà. Posem que tenim el següent:

```
{ path: 'user-register', component: UserRegisterComponent }
```

Això es tradueix en que quan el navegador detecti la url `http://<prefix_portal_web>/user-register` es carregarà tot l'arbre de components fins arribar a `UserRegisterComponent`, aquest, també s'inclou. D'aquesta forma, quan un usuari vagi a aquesta url veurà la pàgina de registre. Hi ha vegades, però, que l'arbre pot estar parcialment carregat, en aquest cas, només es carreguen els components que resten. El fitxer que conté aquests enllaços és el **app-routing.module.ts**

Per últim, anomenar el paper del servei **auth-guard.service.ts** el qual verifica que els permisos per entrar a un component determinat siguin els correctes. En cas que no ho siguin, redirigeix a la pàgina home que pertoca o al login.

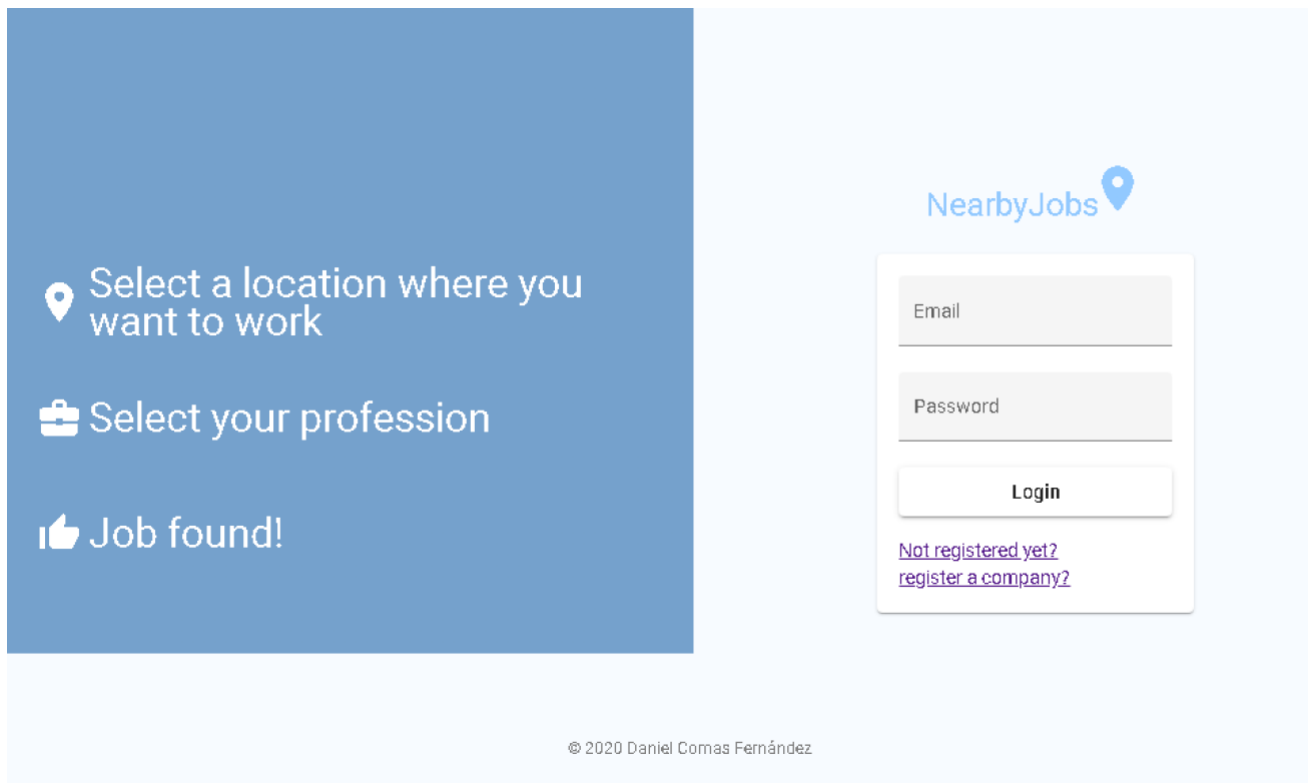
Ara, ja ens podem ficar a programar les pàgines.

9.3. Pàgines

Totes les pàgines s'han realitzat a partir dels mockups i seguint el concepte de disseny *Google Material Design*. A continuació s'indiquen els passos que s'han fet en aquest punt:

- Escollit color de fons i títols per totes les pàgines i s'ha creat el logo
- Elaborat l'html de cada pàgina
- Programat la interacció
- Programat els serveis frontend que els quals connecten els components amb els serveis web del servidor. Deixem preparada la comunicació
- Comprovat que es compleix amb les especificacions

9.3.1. Login



9.3.2. User Register



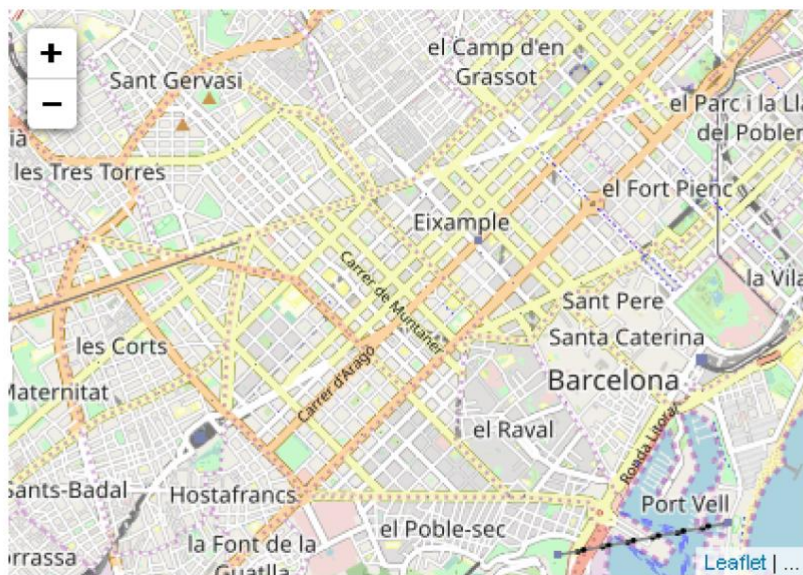
User register

Email

Password


Profession (Start to write your profession's name but try to select existence or more used ones. Write a new one in case it doesn't exist). You can write more than one

Profession...



Register

Els formularis si contenen errors marquen els camps i mostren textos informatius en vermell. Com en el pas següent d'haver clicat el botó de "Register" sense afegir cap dada:

NearbyJobs 

User register

Email

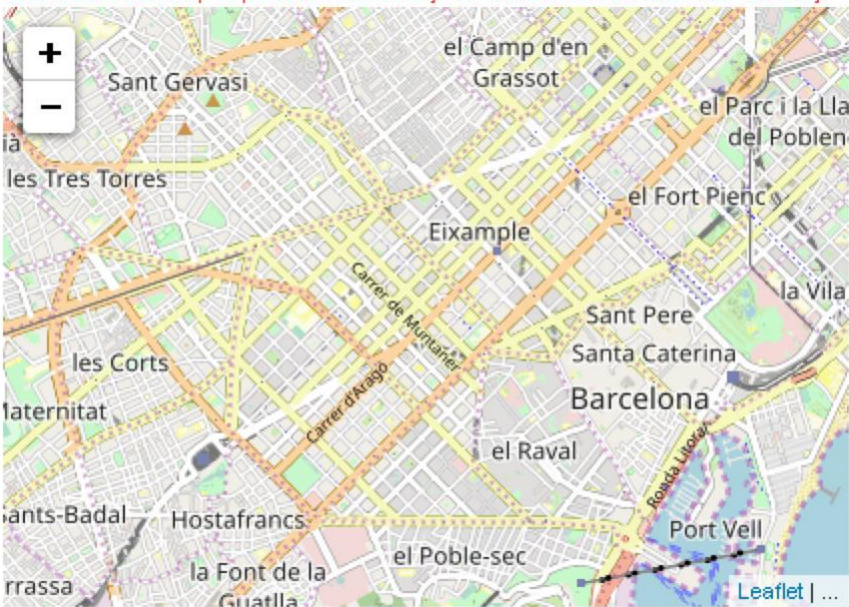
Password

Profession (Start to write your profession's name but try to select existence or more used ones. Write a new one in case it doesn't exist). You can write more than one

Add at least one profession

Profession...

Mark on the map a position where you would like to work or nearby



Register

User register

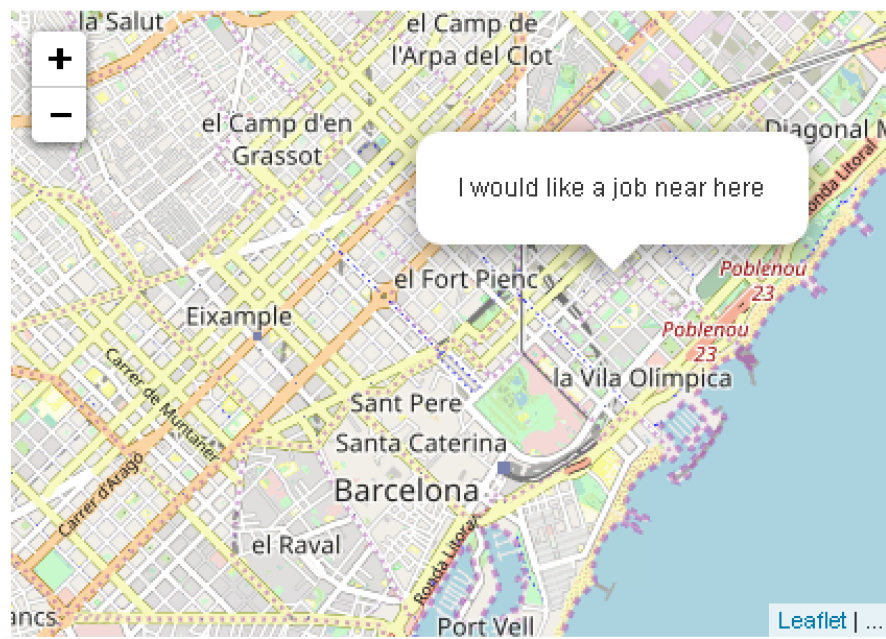
Email
danicommas@uoc.edu

Password

Profession (Start to write your profession's name but try to select existence or more used ones. Write a new one in case it doesn't exist). You can write more than one

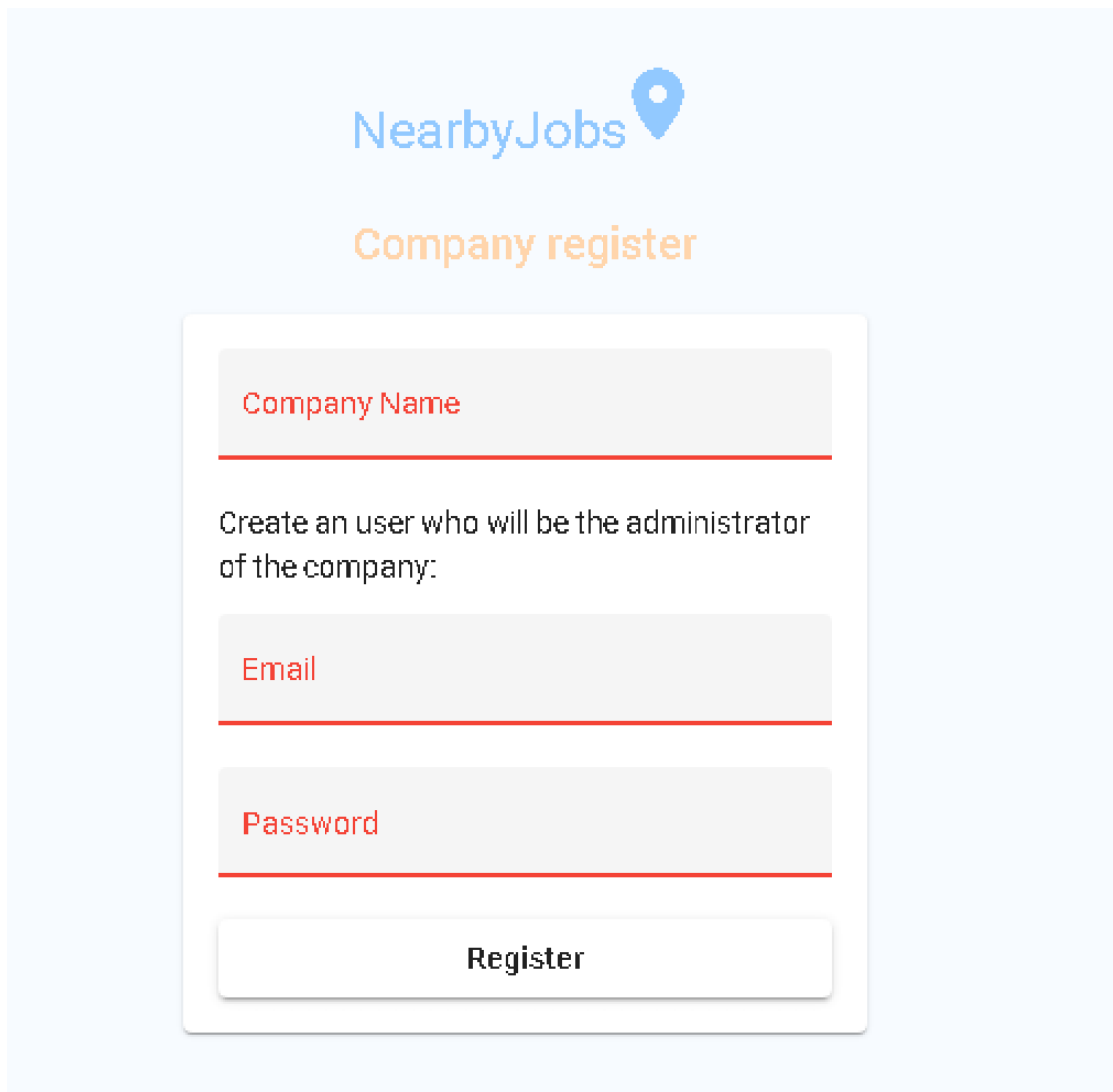
.NET Developer ✕ Profession...

Mark on the map a position where you would like to work or nearby




Register

9.3.3. Company Register



The screenshot shows a registration form for a company. At the top, the 'NearbyJobs' logo is displayed in blue, featuring a location pin icon. Below the logo, the title 'Company register' is written in orange. The form itself is a white rounded rectangle with a thin grey border. It contains three input fields, each with a red label and a red underline: 'Company Name', 'Email', and 'Password'. Below these fields is a white button with a grey border and the text 'Register' in bold black font. The entire form is centered on a light blue background.

NearbyJobs 

Company register

Company Name

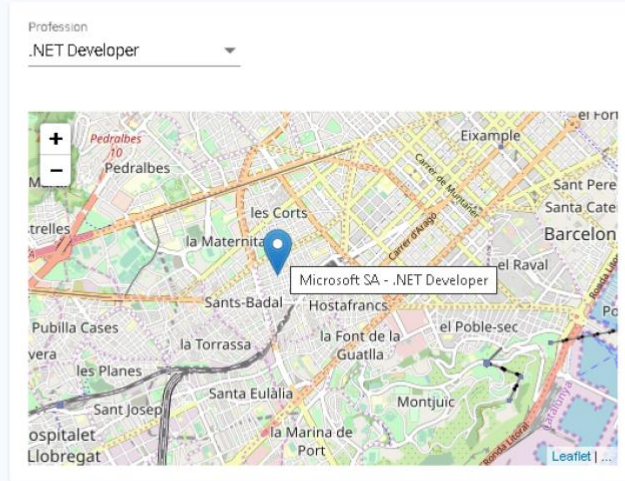
Create an user who will be the administrator of the company:

Email

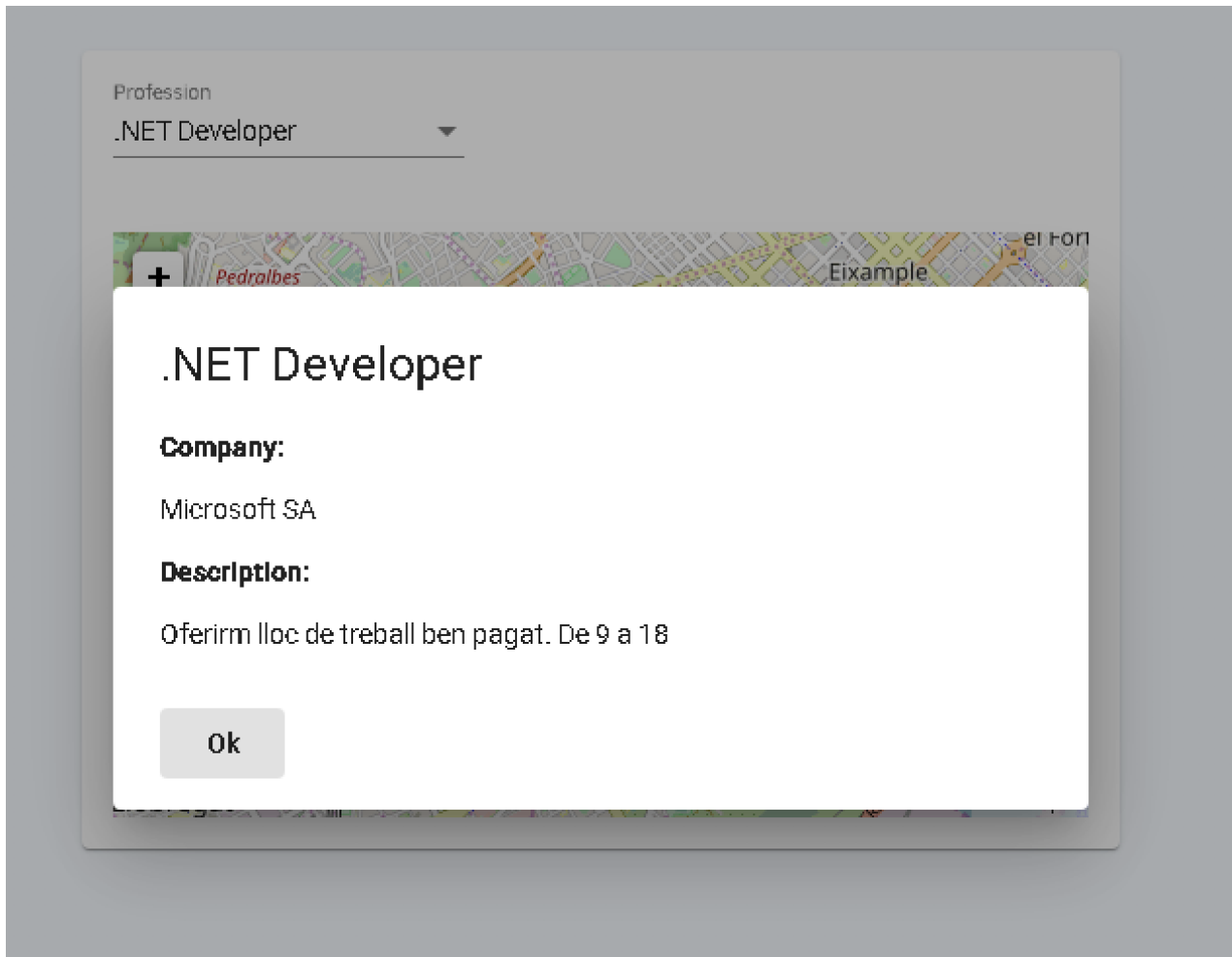
Password

Register


9.3.4. User Home



9.3.5. User Home - Show job



9.3.6. Company Home


NearbyJobs 

company1@uoc.edu (Microsoft SA) [Logout](#)

[Home](#) [Jobs](#)

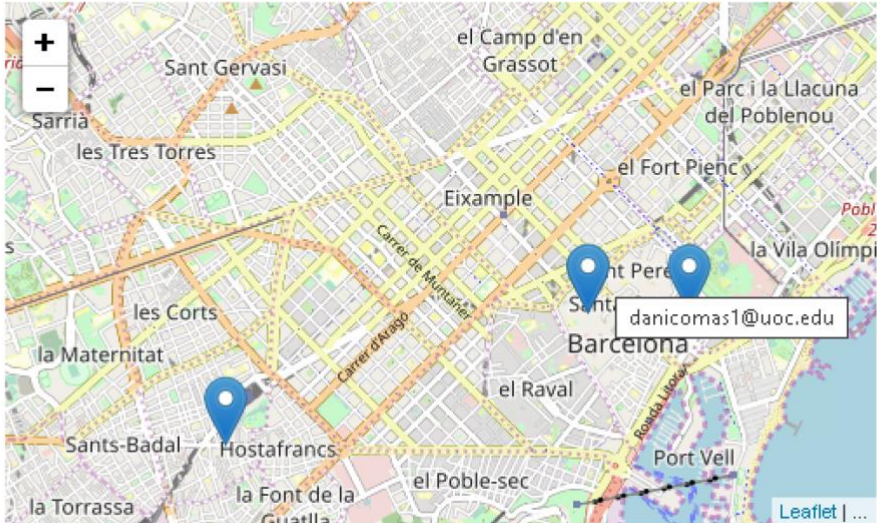
Enter a profession to show the professionals on the map

Profession...



Enter a profession to show the professionals on the map

.NET Developer



9.3.7. Company Jobs

The screenshot shows the 'NearbyJobs' application interface. At the top left is the logo 'NearbyJobs' with a location pin icon. At the top right, the user is logged in as 'company1@uoc.edu (Microsoft SA)' with a 'Logout' button. Below the logo are two navigation buttons: 'Home' and 'Jobs'. The 'Jobs' page is active, displaying a table of job listings. Above the table is an 'Add new job' button. The table has two columns: 'Profession' and 'Description'. One job is listed with the profession '.NET Developer' and the description 'Ofertim lloc de treball ben pag...'. To the right of the description is a trash icon. Below the table, there is a pagination control showing 'Items per page: 25', '1 - 1 of 1', and navigation arrows.

company1@uoc.edu (Microsoft SA) Logout

Home Jobs

Jobs

Add new job

Profession	Description	
.NET Developer	Ofertim lloc de treball ben pag...	

Items per page: 25 1 - 1 of 1 |< < > >|

© 2020 Daniel Comas Fernández

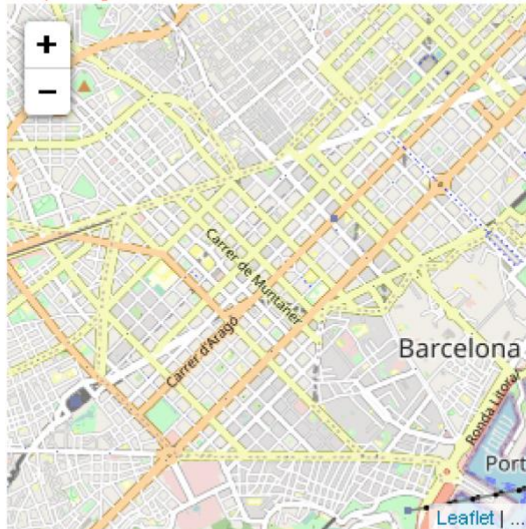
9.3.8. Company Jobs – New Job

Enter a profession. Try to select a profession shown in the autocomplete while you are writing. However, let the text you entered if the profession not exists:

Profession (eg: .NET Developer)

Description (Objectives, Duties, Working hours, Benefits, Salary...)

Mark on the map the job location



Add

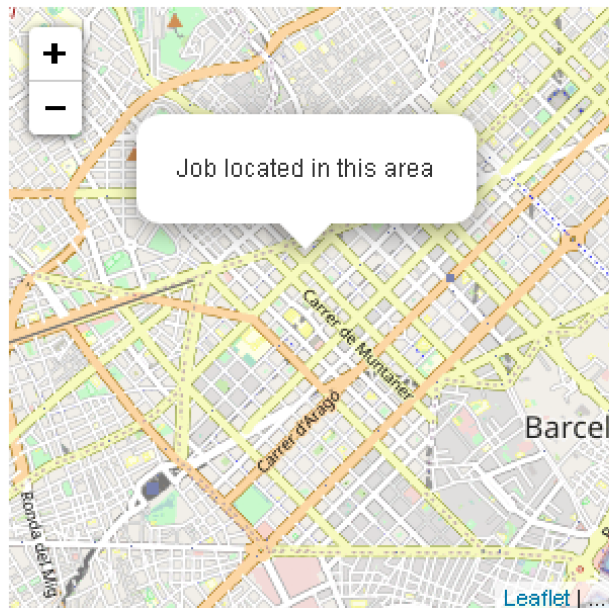
Enter a profession. Try to select a profession shown in the autocomplete while you are writing. However, let the text you entered if the profession not exists:

.NET Developer

Description (Objectives, Duties, Working hours, Benefits, Salary...)

It is required a Microsoft Azure Cloud Developer to develop our cloud products.

Mark on the map the job location



Add

Ja tenim les pàgines programades i també les comunicacions amb el servidor. Ara, cal centrar-nos en aquesta última part.

10. El servidor

Les tasques a realitzar en el servidor són per una part les referents a la creació de base de dades i per l'altre la programació dels serveis web. Per l'usuari que fa ús de l'aplicació tot aquesta part és transparent.

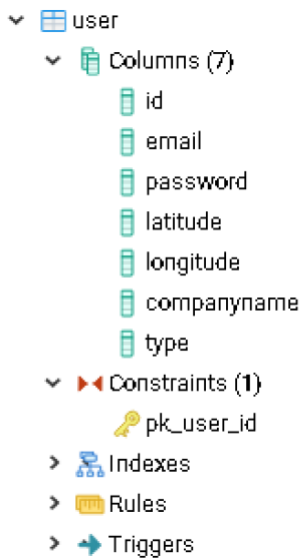
10.1. Base de dades

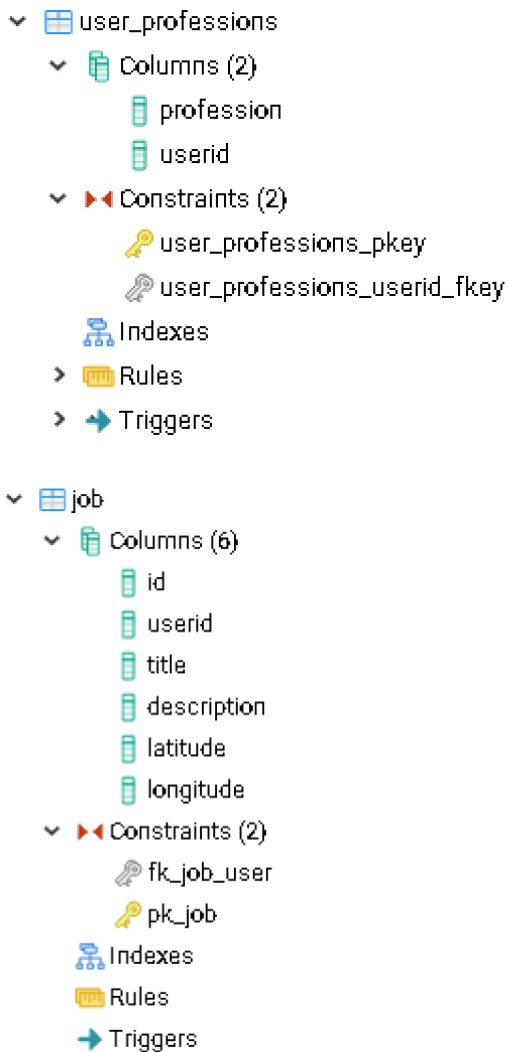
Per crear la base de dades es segueix l'esquema invariant del punt de vista de la informació fet en els passos de disseny del projecte. Per una banda, s'escull un tipus de base de dades relacional(5*) perquè és un concepte que ens pot donar un bon rendiment en l'execució de consultes i consistència de les dades.

En el mercat hi ha nombrosos i diversos sistemes gestors de bases de dades(6*). Per aquest projecte s'escull PostgreSQL que és open source(7*) i amb una gran comunitat al darrere. A part de la creació de bases de dades relacions amb els seus elements bàsics, ens permet també:

- crear esquemes
- usuaris per administrar la base de dades establint restriccions entre ells
- crear i editar: vistes, funcions, índex, disparadors i seqüències

Ja podem crear les taules i relacions entre aquestes:





Per fer tot aquest punt, s'han pres les següents decisions:

- L'atribut id de la taula User és una clau primària(8*) d'aquesta
- Els atributs profession i userid de la taula user_professions formen una clau primària composta, d'aquesta forma no es repeteixen parells.
- L'atribut userid de la taula user_professions és una clau forània(9*) amb l'atribut id d'User.
- L'atribut id de la taula job és la clau primària d'aquesta

10.2. Els serveis web

Ja tenim el frontend i la base de dades i ara ens manca la part on es processen tots els enviaments, és a dir la part servidora. Aquesta part és l'únic punt d'accés a la base de dades anterior.

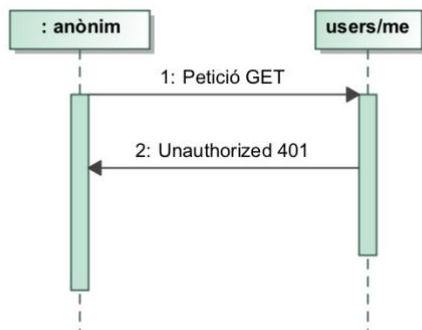
Es decideix fer la part servidora seguint l'arquitectura de software API REST. Aquesta conté un seguit de restriccions i metodologies que ens permeten d'estandarditzar les comunicacions entre client i servidor.

Les API REST es poden comunicar en formats de documents de dades com XML o JSON, en el nostre cas utilitzem aquest últim pel rendiment i per la facilitat d'accedir i treballar amb les dades.

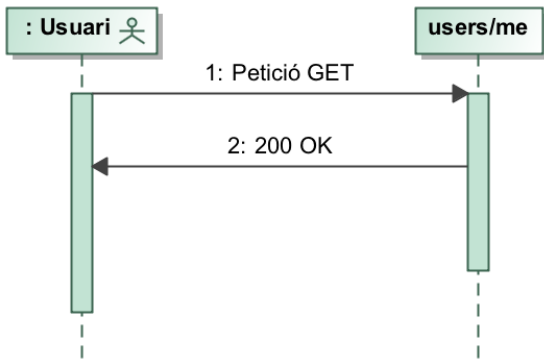
L'stack tecnològic^(10*) utilitzat per aquest fi és el runtime^(11*) NodeJS amb el servidor web Expressjs que ens permet crear aquests serveis amb JavaScript -un dels llenguatges més utilitzats- i disposem de nombroses llibreries al nostre abast per tal d'afegir-li les capacitats requerides.

10.2.1. Autenticació i autorització d'usuaris

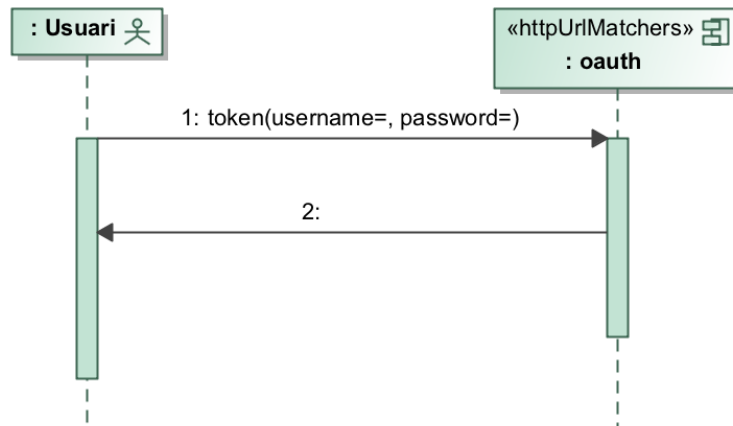
La seguretat és un requisit del projecte, és per això, que els serveis web s'han de restringir i processar i retornar dades depenent de l'usuari que hagi fet l'acció. Posem un exemple, si un usuari anònim fa una petició per obtenir un perfil ha de retornar que no té accés, com il·lustra el diagrama d'interacció:



En canvi, ha de retornar dades si l'usuari està autenticat:



Com ho fem? En aquest cas, treballarem sobre el concepte d'OAUTH2 grant-type password(12*). L'usuari omple les dades de validació en el portal web, s'envien al servidor i si són correctes es retorna un *token* (un codi):



Procés d'obtenció de token

Aquesta és la petició i la resposta a nivell tècnic:

```

curl --location --request POST 'http://localhost:3000/oauth/token' \
--header 'Content-Type: application/json' \
--data-raw '{
  "grant-type": "password",
  "username": "company1@uoc.edu",
  "password": "12345aA"
}'

{"token": "<token>"}
  
```

El *token* obtingut s'envia en totes les peticions com a capçalera *HTTP Authorization: Bearer <Token>*(13*) per tal d'autoritzar l'usuari i realitzar totes les tasques que se li permetin fer, com per exemple l'obtenció del perfil.

Veiem la petició i resposta tècnicament:

```

curl -H 'Content-type: application/json' -H "Authorization: Bearer <TOKEN>" http://localhost:3000/users/me

{"id":9,"email":"company1@uoc.edu","password":"12345aA","latitude":null,"longitude":null,"companyname":"Microsoft SA","type":2,"professions":[]}
  
```

Cal destacar, que s'han escullit els *JSON Web Tokens (JWT)* el qual ens dona la capacitat d'eliminar per complet o reduir l'acoblament(14*) del sistema alhora de fer l'autorització del usuari. És a dir, es poden publicar serveis utilitzant els mateixos *tokens* sense requerir de tots els algorismes que fa servir l'emissor de *tokens* (també anomenat servidor d'autorització).

Si entrem al web <https://jwt.io> es pot comprovar el contingut del JWT. Enganxem un generat pel sistema:



The image shows a JWT token and its decoded structure. On the left, a pink box contains the token: `eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpYXQiOiJlOTAzMzU0ODksImV4cCI6MTU5MDMzOTA0S2wic3ViIjoiOSJ9.bP6Zo4XH9YGKR2co5lwW9yK7g2aFNEfgvpAWd3BSSpA`. On the right, a table shows the decoded structure:

HEADER: ALGORITHM & TOKEN TYPE
<pre>{ "alg": "HS256", "typ": "JWT" }</pre>
PAYLOAD: DATA
<pre>{ "iat": 1590335489, "exp": 1590339089, "sub": "9" }</pre>

Veiem que a la secció *payload* apareix el contingut del *token* desxifrat i l'atribut "sub" que conté l'identificador (id) de l'usuari. Amb aquest id ja és suficient per realitzar tasques sobre aquest usuari.

Per tal d'afegir més seguretat, signem el JWT amb un algorisme de clau pública per tal de comprovar la seva validesa quan s'enviïn en les peticions.

10.2.2. Logout

Un altres dels requisits és que es pugui desconnectar la sessió de manera segura.

El repte que ens hem trobat ha estat amb la naturalesa dels JWT tokens, els quals són vàlids fins que no es superen la seves dates d'expiració. És per això, que s'ha cregut òptim tindre un *llistat de tokens revocats* d'aquesta manera ja no poder ser utilitzats en properes peticions.

El procés que es segueix un cop l'usuari clica a logout és el següent:

- 1- Es revoca el token
- 2- Es borren les dades de l'usuari que estaven guardades en el local storage del navegador
- 3- Es redirigeix al login

Vista la validació, autorització i la desconnexió segura, detallem els serveis web de l'aplicació.

10.2.3. POST /oauth/token

Objectiu:

Retornar un *token* per poder utilitzar els serveis que requereixen de permisos especials

Paràmetres:

Nom	Descripció	Tipus	Requerit
grant-type	Tipus de petició per obtenir el token. Es contempla únicament el tipus "password"	string	SI
username	Nom d'usuari	string	SI
password	Contrasenya de l'usuari	string	SI

Es requereix *token* d'autorització? NO

Resposta:

Retorna una resposta amb HTTP STATUS 200 OK i amb un objecte amb la següent estructura:

```
{
  "token": "<JWT_TOKEN>"
}
```


10.2.4. POST /oauth/token/revoke

Objectiu:

Inutilitzar el *token* perquè ja no es pugui fer servir per accedir a recursos restringits, és a dir ja no serà vàlid pel seu ús.

Paràmetres:

Nom	Descripció	Tipus	Requerit
token_type_hint	Tipus de token. Es contempla únicament el tipus "access_token"	string	SI
token	El token	string	SI

Es requereix *token* d'autorització? NO

Resposta:

Retorna una resposta amb HTTP STATUS 200 OK

10.2.5. GET /users/me

Objectiu:

Retornar les dades de l'usuari autenticat

Paràmetres:

Sense paràmetres

Es requereix *token* d'autorització? SI

Resposta:

Retorna una resposta amb HTTP STATUS 200 OK i amb un objecte amb la següent estructura:

```
{
  "id": 2,
  "email": "danicomas@uoc.edu",
  "password": "12345aA",
  "latitude": 41.3850639,
  "longitude": 2.1734035,
  "companyname": null,
  "type": 1,
  "professions": [
    ".NET Developer"
  ]
}
```

10.2.6. GET /professions

Objectiu:

Retorna les professions que existeixen en el sistema

Paràmetres - Querystring:

Nom	Descripció	Tipus	Requerit
profession	Valor que permet filtrar les professions. Per exemple: Si fem 'Dev' pot retornar '.NET Developer' i 'JAVA Developer'	string	SI però pot ser buit

Es requereix token d'autorització? NO

Resposta:

Retorna una resposta amb HTTP STATUS 200 OK i amb un objecte amb la següent estructura:

```
[  
  "Java EE",  
  ".NET Developer"  
]
```

10.2.7. POST /users

Objectiu:

Crear un usuari, tant estàndard com d'empresa

Paràmetres:

Nom	Descripció	Tipus	Requerit
email	Correu utilitzat per validar-se al sistema	string	SI
password	Contrasenya utilitzada per validar-se al sistema	string	SI
latitude	Coordenada juntament amb longitude que determinen la zona per on els hi agradaria treballar	double	Requerit si el tipus és 1
longitude	Coordenada juntament amb latitude que determinen la zona per on els hi agradaria treballar	double	Requerit si el tipus és 1
professions	Professions de l'usuari	array de strings	Requerit si el tipus és 1. Almenys una professió
companyname	Nom de l'empresa	string	Requerit si el tipus és 2
type	1 per usuari estàndard, 2 per empresa	number	SI

Es requereix *token* d'autorització? NO

Resposta:

Retorna una resposta amb HTTP STATUS 200 OK i amb un objecte amb la següent estructura:

```
{
  "email": "fran.perexz@toshiba.es",
  "password": "toshi123!",
  "companyname": "Toshiba SA",
  "type": 2
}
```

10.2.8. POST /users/:id/jobs

Objectiu:

Crear una oportunitat laboral per a una empresa

Paràmetres:

Nom	Descripció	Tipus	Requerit
title	El títol de la oportunitat, és la professió que s'oferta.	string	SI
description	Contrasenya utilitzada per validar-se al sistema	string	SI
latitude	Coordenada juntament amb longitude que determinen la zona per on s'oferta la posició	double	SI
longitude	Coordenada juntament amb latitude que determinen la zona per on s'oferta la posició	double	SI

Es requereix *token* d'autorització? SI

Retorna una resposta amb HTTP STATUS 200 OK i amb un objecte amb la següent estructura:

```
{
  "title": "Project Manager",
  "description": "It is required a project manager to do the tracking of an internal cloud project",
  "latitude": "41.389105",
  "longitude": "2.1446393"
}
```

10.2.9. DELETE /users/:id/jobs/:jobid

Objectiu:

Eliminar una oportunitat laboral d'una empresa

Paràmetres:

Sense paràmetres

Es requereix *token* d'autorització? SI

Retorna una resposta amb HTTP STATUS 200 OK

10.2.10. GET /users/:id/jobs

Objectiu:

Retorna totes les oportunitats laborals vigents d'una empresa

Paràmetres:

Sense paràmetres

Es requereix *token* d'autorització? SI

Retorna una resposta amb HTTP STATUS 200 OK i amb un objecte amb la següent estructura:

```
[
  {
    "id": 1,
    "userid": 9,
    "title": ".NET Developer",
    "description": "Oferim lloc de treball ben pagat. De 9 a 18",
    "latitude": 41.3782223,
    "longitude": 2.1325527
  }
]
```

10.2.11. GET /users/jobs

Objectiu:

Retorna totes les oportunitats laborals vigents d'una empresa

Paràmetres - Querystring:

Nom	Descripció	Tipus	Requerit
title	Valor que permet filtrar les oportunitats per la professió.	string	SI

Es requereix *token* d'autorització? SI

Retorna una resposta amb HTTP STATUS 200 OK i amb un objecte amb la següent estructura:

```
[
  {
    "id": 1,
    "userid": 9,
    "title": ".NET Developer",
    "description": "Oferim lloc de treball ben pagat. De 9 a 18",
    "latitude": 41.3782223,
    "longitude": 2.1325527,
    "user": {
      "id": 9,
      "email": "company1@uoc.edu",
      "password": "12345aA",
      "latitude": null,
      "longitude": null,
      "companyname": "Microsoft SA",
      "type": 2
    }
  }
]
```

10.2.12. GET /users

Objectiu:

Retorna els usuaris que tenen la professió

Paràmetres - Querystring:

Nom	Descripció	Tipus	Requerit
profession	Valor que permet filtrar els usuaris per la professió.	string	SI

Es requereix *token* d'autorització? SI

Retorna una resposta amb HTTP STATUS 200 OK i amb un objecte amb la següent estructura:

```
[
  {
    "id": 2,
    "email": "danicomas@uoc.edu",
    "password": "12345aA",
    "latitude": 41.3850639,
    "longitude": 2.1734035,
    "companyname": null,
    "type": 1
  },
  {
    "id": 8,
    "email": "test1@uoc.edu",
    "password": "12345aA",
    "latitude": 41.3850639,
    "longitude": 2.1834035,
    "companyname": null,
    "type": 1
  },
  {
    "id": 16,
    "email": "danicomas1@uoc.edu",
    "password": "12345aA",
    "latitude": 41.3850639,
    "longitude": 2.1734035,
    "companyname": null,
    "type": 1
  },
],
```



```
{
  "id": 17,
  "email": "testn@uoc.edu",
  "password": "12345aA",
  "latitude": 41.3752406749478,
  "longitude": 2.13776220573146,
  "companyname": null,
  "type": 1
}
```

Un cop els serveis s'han especificat i programat, cal provar que tant el frontend com aquesta part s'estan comunicant tal i com s'havia definit, de no ser així, cal fer modificacions.

Arribats en aquest punt l'aplicació està completa.

11. Tasques següents

A mesura que es va avançant amb el projecte o un cop ha conclòs, es troben tasques que no estaven en la planificació però que es podrien realitzar en altres estadis. Aquestes són les que s'han recopilat:

- Refresc de *JWT tokens* per augmentar la seguretat. Posar els *tokens* amb una duració de 5 minuts i anar-los refrescant per tal d'evitar que si un usuari malintencionat obtingués el *token* veiés reduït el seu temps d'acció
- Fer un procés automàtic per eliminar el llistat de *JWT tokens revocats*
- Ficar una data d'inici de la oportunitat laboral
- Fer que els serveis web de cercar professionals i oportunitats filtrin per zones en comptes de tornar-ho tot
- Que els usuaris puguin pujar el seu currículum
- Mostrar més dades de l'usuari quan una empresa cerca professionals, per exemple el nom, cognom, la professió i el CV
- Realitzar un vídeo de presentació en el login
- Localitzar textos del frontend per a poder traduir l'aplicació a altres idiomes

12. Impacte dels riscos

El projecte s'ha desenvolupat sense incidències destacables i complint les dates proposades en la planificació inicial inclús avançant tasques de fases posteriors. També, s'han anat fent còpies en cada sessió de desenvolupament, edició de documents, i no s'ha requerit de recuperar estats previs.

Cal anotar per altres projectes semblants que la part servidora es pot fer abans que el frontend així s'evita un seguit de modificacions posteriors en el frontend.

13. Resultats i conclusions

Des de l'inici es va planificar el projecte per ser completat en dues fases amb uns objectius específics per cada una. La primera, d'anàlisi, disseny general i desenvolupament del frontend i l'altre pel desenvolupament de la part servidora. En elles i també, en la planificació inicial es van aplicar tècniques de diferents assignatures cursades durant el grau. La planificació, identificació dels recursos i pla de riscos juntament amb les seves mesures correctores i la forma d'estructurar el document, de gestió de projectes, la definició de requisits segons stakeholders o usuaris que n'han faran ús, d'enginyeria de requisits.

Abans de començar a codificar les pàgines de l'aplicació es van usar conceptes d'interacció persona-ordinador, com l'*user journey* per tindre en compte les emocions que li poden sorgir a l'usuari al navegar pel web i els *mockups* per la forma de presentar la informació, com quins camps i controls utilitzar i com organitzar-los.

Continuant, per dissenyar el programari i per tal de només seguir les pautes marcades durant la implementació, es van utilitzar tècniques de l'assignatura d'Enginyeria del programari de components i sistemes distribuïts, com l'esquema invariant del punt de vista de la informació, els casos d'ús i el diagrama de components.

Amb tot això i durant tota la fase de desenvolupament, s'ha pres consciència de les complexitats que hi ha en aquest tipus de projectes, la importància de la realització prèvia de les especificacions, els diagrames adients, l'estudi i selecció dels conceptes, de les tecnologies i dels patrons de disseny del programari per tal de minimitzar l'impacte dels canvis que sempre s'acaben produint. Tot es canviant i adaptable i l'actitud per detectar i trobar solucions segons el context determinant.

Arribats a aquest punt, complíem els objectius secundaris a la vegada que aconseguíem el principal, obtenir una aplicació web que permetés als usuaris consultar ofertes de feina per proximitat i a les companyies cercar professionals.

14. Glossari

*1 classes: és un element bàsic en la programació orientada a objectes. Com a exemple si tenim una classe Gos amb les propietats color i mida, aquest element per ara només és una definició, però un cop s'està executant el codi, ens permet crear objectes diversos com per exemple new Gos("negrós", "petit") o new Gos("marronós", "gran").

2* *patró singleton*: s'inicialitza (es crea) un objecte el primer cop que es fa servir per tal de que en els següents usos es faci servir aquest mateix.

3* *tipar o typing en anglès*: és un terme utilitzat per definir que un valor és d'un tipus en concret, per exemple "hola" és del tipus string o true és boolean.

4* *drag & drop*: és un mètode d'interacció que permet arrossega i deixa anar elements.

5* *base de dades relacional*: és un tipus de base de dades on les seves taules mantenen les relacions per mantenir la consistència de les dades i així evitar duplicitats o registres orfes a la vegada que s'augmenta el rendiment de les consultes.

6* *sistemes gestor de bases de dades*: creació, modificació i administració d'una base de dades.

7* *open source*: un programa és open source o codi lliure quan està disponible per ser consultat, redistribuït i modificat.

8* *clau primària*: en una base de dades de tipus relacional, les taules tenen un camp o una combinació de camps que identifica de forma única a cada fila d'una taula. Quan hi ha una combinació de camps s'anomena clau primària composta.

9* *clau forània*: és una referència entre dos taules. D'aquesta manera queden unides i els seus registres tenen unes restriccions

10* *stack tecnològic*: combinació de productes i de llenguatges de programació

11* *runtime environment*: permet l'execució d'un determinat model

12* *OAuth2 Grant type password*: OAuth2 és un estàndard obert sobre fluxos d'autorització d'aplicacions. Quan hi ha un problema a solucionar és bo seguir estàndards ja que han estat influenciats per varis agents.

13* *HTTP Authorization: Bearer <Token>*. El protocol utilitzat per enviar i rebre peticions web és el HTTP. Aquest protocol en cada petició i resposta conté unes capçaleres o headers. Aquesta és una d'elles.

14* *acoblament*: És el grau de dependència entre diferents algoritmes. En el disseny del programari es cerca un nivell baix.