

**UNIVERSITAT OBERTA DE CATALUNYA**  
**ENGINYERIA TÈCNICA DE TELECOMUNICACIONS TELEMÀTICA**

---

**GangMenu**  
**“Sistema col.laboratiu de publicació de menús de restaurants”**

**Juny-2012**

**Autor: Marc Pursals i Garcia**

**Director de Projecte: Manel Llopart Vidal**

**ABSTRACT:**

GangMenu és un joc. És un joc que intenta donar resposta a un problema que teníem amb els amics cada vegada que intentàvem anar a dinar al voltant de la feina. I dic joc, perquè tot i que en realitat es tracta d'una aplicació mòbil amb totes les seves complexitats, per mi ha estat un joc que engloba tres de les passions que tinc, el hardware, les aplicacions web i els smartphones.

En concret, GangMenu es una aplicació desenvolupada en Phonegap que permet, gràcies a la geolocalització, compartir les imatges dels menús dels restaurants/bar als que anem, i que siguin accessibles per altra gent que vulgui saber què i on menjar.

Aquesta aplicació està basada en una infraestructura de 4 parts: aplicació web, api, aplicació mòbil i infraestructura de hardware. Les dues primeres parts estan desenvolupades amb Ruby on Rails, que es un framework per crear aplicacions web basat en Ruby (fill il·legítim de Perl) i encarat al paradigma MVC (model-view-controller). La tercera, Phonegap, es un altre framework encarat a crear aplicacions multiplataforma per smartphones. I la quarta, la infraestructura, es tracta de maquinari amb linux, apache, passenger i mongoDB.

El resultat es un grup compacte d'aplicació mòbil, aplicació web, api i hardware que té la intenció de donar servei a la gent que vulgui saber quins són els restaurant que hi ha al seu voltant, i poder veure els menús i els preus que en aquell bar s'hi donen.

GangMenu es un juego. Es un juego que intenta dar respuesta a un problema que teníamos con los amigos cada vez que intentábamos ir a comer alrededor de la empresa. Y digo juego porque aunque en realidad se trate de una aplicación móvil con todas sus complejidades, para mi ha sido un juego que engloba tres de las pasiones que tengo: el hardware, las aplicaciones web y los smartphones.

En concreto, GangMenu es una aplicación desarrollada en Phonegap que permite, gracias entre otras cosas a la geolocalización, compartir las imágenes de los menús de los restaurantes/bar a los que vamos, para que sean accesibles a otra gente que quiera saber qué y dónde comer.

Esta aplicación esta basada en una infraestructura de 4 partes: aplicación web, api, aplicación móvil y infraestructura de hardware. Las dos primeras partes estan desarrolladas en Ruby on Rails, que es un framework para crear aplicaciones web basado en Ruby (hijo ilegítimo de Perl) i encarado al paradigma MVC (model-view-controller). La tercera parte, Phonegap, es otro framework encarado a crear aplicaciones multiplataforma para smartphones. Y la cuarta, la infraestructura, se trata de maquinaria con linux, apache, passenger y mongoDB.

El resultado es un grupo compacto de aplicación móvil, aplicación web, api y hardware que tiene la intención de dar servicio a la gente que quiera saber cuales son los restaurantes que hay a su alrededor, poder ver el menú que se da y el precio que tiene.

GangMenu is a game. It is a game that tries to answer to a problem we had with friends every time we tried to eat around the company. For me is a game because although in reality it is a mobile application with all its complexities, for me it was a game that includes three of my passions: the hardware, web applications and smartphones.

Specifically, GangMenu is an application on PhoneGap that allows, thanks to geolocation and other stuff, share images from the menus of the restaurants / bar to be accessible to other people who want to know what and where eat.

This application is based on an infrastructure of 4 parts: web application, api, mobile application and hardware. The first two parts are developed in Ruby on Rails, a framework aimed on rcreating web applications based on Ruby (Perl's

illegitimate son) and approaching the MVC paradigm (model-view-controller).

The third part, PhoneGap is another framework aimed towards creating cross-platform applications for smartphones. And the fourth part is machinery set up with linux, apache, passenger and MongoDB.

The result is a compact group formed by mobile application, web application, api and hardware that is intended to serve people who want to know which restaurants are around him and want to see the menu given and the price.

**RESUM:**

Aquest treball contempla l'execució del projecte GangMenu, que es una aplicació mòbil i web que permet a usuaris compartir els menús dels restaurants on vagin a dinar perquè sigui accessibles per altres usuaris.

El projecte està dividit en tres grans apartats:

- el desenvolupament de una aplicació web
- el desenvolupament de una API
- el desenvolupament de una aplicació mòbil

El primer apartat que desenvoluparé serà l'estat de l'art d'aquest projecte on s'explica els antecedents que han fet que decideixi desenvolupar aquesta aplicació, així com un petit resum de les tecnologies que han influït en la possibilitat de desenvolupar-la.

El segon apartat que desenvoluparé, serà la aplicació mòbil per ser l'aplicació que més treball comporta i més complexitat donada la novetat dels frameworks multiplataforma. En aquest apartat desenvoluparé el disseny de l'aplicació, els seus casos d'ús, els mockups (o diagrames de blocs funcionals) de l'aplicació i la descripció del frameworks que he utilitzat per desenvolupar-la.

En tercer terme desenvoluparé l'apartat corresponent a la aplicació web on, de manera anàloga a la anterior, s'explicarà el disseny, els casos d'ús i la descripció del framework amb el que he desenvolupat l'aplicació.

En quart lloc, s'explicitarà el funcionament de la API que dona cobertura al terminal mòbil per nutrir-lo de serveis del servidor web i de la base de dades, de manera que s'explicarà les diferents funcionalitats que, en aquesta fase inicial del projecte, cobreixen les especificacions de l'aplicació mòbil. Aquesta API, està programada en el mateix framework que l'aplicació web, i per tant, no es desenvoluparà aquest punt en aquest apartat.

En cinquè lloc, es farà una descripció tècnica dels diferents aspectes col·laterals de l'aplicació en general, com son les bases de dades utilitzades, el sistema d'allotjament inicial i el previst, així com un simples càlculs de la escalabilitat que estimo caldrà per dimensionar el hardware.

En sisè lloc, es presentaran i explicaran algunes captures de pantalla de les tres aplicacions, per poder ensenyar el funcionament real així com les problemàtiques que inevitablement ens anirem trobant.

Per últim s'explicaran algunes de les línies de futur del projecte, i les passes següent que es tindrien que desenvolupar, per tal de fer progressar el projecte en els pròxims mesos.

A títol informatiu, crec necessari comentar que tot i que aquest projecte està pensat per complir amb les exigències del TFC, des de un bon principi ha nascut amb una clara vocació de negoci. La intenció es que , una vegada desenvolupada tota la part tècnica, aquest projecte s'ha de completar amb un "bussiness plan" així com amb unes accions comercials, que l'hi donin la empenta que crec que pot tenir. Es un projecte sòlid, que es basa en la col·laborativitat de la gent, en les xarxes socials, en una necessitat clara i que ajuda i millora una situació actual. Per tant crec que pot ser un bon començament per una línia de negoci basada en internet.

## INDEX:

	Abstract	Pàg: 2
	Resum	Pàg: 4
	Index de figures	Pàg: 6
	Index de taules	Pàg: 7
Capítol 1	Introducció	Pàg: 8
Capítol 2	Estat de l'Art	Pàg: 10
Capítol 3	Disseny i implementació	Pàg: 16
3.1	Aplicació mòbil	Pàg: 16
3.1.1	Casos d'ús	Pàg: 17
3.1.2	Mockups	Pàg: 22
3.1.3	Descripció del framework	Pàg: 23
3.2	Aplicació web	Pàg: 26
3.2.1	Casos d'ús	Pàg: 27
3.2.2	Descripció del framework	Pàg: 32
3.3	API	Pàg: 35
3.4	Interacció entre sistemes	Pàg: 38
3.5	Facebook i Twitter	Pàg: 40
3.6	Descripció Tècnica	Pàg: 40
3.6.1	Bases de dades	Pàg: 40
3.6.1.1	Descripció d'alternatives	Pàg: 42
3.6.1.2	Justificació de la elecció	Pàg: 43
3.6.2	Sistema d'allotjament	Pàg: 44
3.6.2.1	Sistema inicial	Pàg: 46
3.6.2.2	Sistema objectiu (virtualització)	Pàg: 46
3.6.3	Escalabilitat	Pàg: 49
Capítol 4	Implementació i demostració	Pàg: 51
4.1	Modificacions i demostració sobre l'aplicació mòbil	Pàg: 51
4.2	Modificacions i demostració sobre l'aplicació web	Pàg: 57
Capítol 5	Conclusions i línies de futur	Pàg: 63
	Bibliografia	Pàg: 65

## INDEX DE FIGURES:

Figura:	1	Logo Foursquare	Pàg:	11
Figura:	2	Logo Google Places	Pàg:	11
Figura:	3	Logo SceneTap	Pàg:	12
Figura:	4	Logo WheretheLadies.at	Pàg:	12
Figura:	5	Logo Spofav	Pàg:	12
Figura:	6	Logo AquiSi	Pàg:	13
Figura:	7	Logo Garrafon	Pàg:	13
Figura:	8	Logo MadridMola	Pàg:	13
Figura:	9	Logo Kekanto	Pàg:	13
Figura:	10	Diagrama de blocs: Interconnexió de sistemes	Pàg:	16
Figura:	11	Pantalla inici aplicació mòbil	Pàg:	17
Figura:	12	Diagrama de blocs: inici aplicació	Pàg:	18
Figura:	13	Pantalles Afegir	Pàg:	18
Figura:	14	Diagrama de blocs: afegir restaurant	Pàg:	19
Figura:	15	Diagrama de blocs: Navigator	Pàg:	19
Figura:	16	Diagrama de blocs: afegir menú	Pàg:	20
Figura:	17	Pantalla afegir comentari	Pàg:	20
Figura:	18	Mockup aplicació mòbil	Pàg:	22
Figura:	19	Pantalla inici aplicació web	Pàg:	27
Figura:	20	Pantalla registre aplicació web	Pàg:	28
Figura:	21	Diagrama de blocs: registrar usuari	Pàg:	28
Figura:	22	Diagrama de blocs: accés usuari	Pàg:	29
Figura:	23	Pantalla inici aplicació web/timeline	Pàg:	29
Figura:	24	Diagrama de blocs: buscar bar	Pàg:	30
Figura:	25	Pantalla cercar bar per direcció	Pàg:	30
Figura:	26	Pantalla enviar menú web	Pàg:	31
Figura:	27	Diagrama de blocs: enviar menú web	Pàg:	31
Figura:	28	Logo Javascript	Pàg:	33
Figura:	29	Logo Ruby on Rails	Pàg:	33
Figura:	30	Logo mongoDB	Pàg:	43
Figura:	31	Disseny del sistema físic	Pàg:	46
Figura:	32	Discs durs virtuals	Pàg:	48
Figura:	33	Gràfica ce escalabliat	Pàg:	49
Figura:	34	Imatges aplicació mòbil 1-13	Pàg:	51-56
Figura:	34	Imatges aplicació web 1-14	Pàg:	57-62

**INDEX DE TAULES:**

Taula:	1	Base de dades: Usuaris	Pàg:	41
Taula:	2	Base de dades: Geoposició	Pàg:	41
Taula:	3	Base de dades: Menú	Pàg:	41
Taula:	4	Base de dades: Comments	Pàg:	41
Taula:	5	Base de dades: Població	Pàg:	41
Taula:	6	Base de dades: Administradors	Pàg:	42
Taula:	7	Base de dades: MongoDB	Pàg:	43
Taula:	8	Permisos de les aplicacions	Pàg:	45

## 1. Introducció (descripció global del projecte)

Només a Catalunya hi ha més bars/restaurants que en tota Finlàndia. Si. I més que en tota Irlanda, Noruega o Dinamarca.

I no es que sigui una situació excepcional de la nostra terra. Al contrari. Segons l'anuari econòmic de la Caixa del 2010<sup>1</sup>, per tota la geografia espanyola es repeteix aquest model de relació habitant-negoci de restauració, que fa del nostre país un *rara habis* dins del món globalitzat.

Podríem dir, per exemple, que segons aquest mateix anuari, les comunitats on hi ha una major densitat de bars/restaurants per habitant serien: (d'aquí endavant anomenarem *bar* al grup *bar/restaurant*)

- Aragó: 1 bar per cada 353 habitants
- Astúries: 1 bar par cada 354 habitants
- Balears: 1 bar per cada 358 habitants

I també podríem ressaltar que, en números absoluts, les comunitats amb més bars de tota Espanya son:

- Catalunya: amb 15.859 bars
- Andalusia: amb 15.741 bars
- Madrid: amb 14.658 bars

En concret, i filant més prim, segons aquest anuari, la distribució general per comunitats, que es pot observar en la següent taula, implica que hi ha una mitjana global de 1 bar cada 452 habitants.

Comunitat	Habitants	nº.bars	bar/habitant
Andalucía	8.059.461	15.741	512
Aragón	1.296.655	3.667	354
Asturias	1.074.862	3.033	354
Baleares	1.030.650	2.880	358
Canarias	2.025.851	3.905	519
Cantabria	572.824	1.360	421
C.yLeón	2.528.417	6.189	409
C-LaMancha	1.977.304	3.579	552
Cataluña	7.210.508	15.859	455
C.Valenc.	4.885.029	11.217	436
Extremadura	1.089.990	2.241	486
Galicia	2.772.533	6.334	438
Madrid	6.081.689	14.658	415
Murcia	1.392.117	2.977	468
Navarra	605.876	1.375	441
PaísVaco	2.141.860	4.040	530
LaRioja	308.968	768	402
Ceuta	76.603	71	1.079
Melilla	69.440	105	661

Hi ha alguns teòrics de l'antropologia que afirmen que, segurament, aquesta distribució està fortament relacionada amb el clima del país. Però veient la diversitat de temperatures que tenim entre comunitats autònomes, sembla difícil unir un fet climàtic a un fet marcadament social i relacional, doncs en un bar sovint es donen serveis des de les 6 del matí fins a les 3 de la matinada.

Però de fet, encara que no ho sembli, de totes les activitats que poden desenvolupar els bars durant el seu servei, la que més moviment diari té és sens dubte la dels menús.

Contràriament al que succeeix en altres països d'Europa, aquí a Espanya hi

(1) Anuari Econòmic de la Caixa

Aquest anuari s'ha anat fent cada any. Les dades històriques sobre aquest tema, i altres, es poden trobar a:

<http://www.anuarieco.lacaixa.com/java/X?cgi=caixa.anuari99.util.ChangeLanguage&lang=esp>



ha la costum de dinar entre una i dues hores durant la jornada laboral. Lluny de voler fer cap mena de valoració comparativa, aquest fet fa que forçosament, i dins de les possibilitats de cadascú, hi hagi un hàbit de menú molt estès i que fomenta, en conseqüència, la relació i socialització entre els treballadors.

I és en aquest fet social i relacional dels clients i els menús dels bars, en el que el nostre projecte GangMenu s'emmarca.

Suposo que més d'una vegada s'han trobat amb que un company/a els ha demanat, quan anaven cap al restaurant de torn, que l'hi diguessin el menú que hi havia aquell dia, tant pot ser per escollir plat o per decidir si anar-hi o no. En aquest cas, i més actualment amb la generalització de les noves tecnologies de comunicacions, la solució es tant senzilla com agafar el mòbil, fer una fotografia i enviar-la per whatsapp.

Però aquí arriba justament el quit de la qüestió, perquè si no tens a cap company/a que et faci el favor de fer-te la fotografia, o trucar-te per dir-te els plats, no hi ha cap manera de saber el menú del bar de la cantonada, ni tampoc tens la possibilitat de tenir disponibles tots el menús de tots els bars de la teva zona per poder decidir on anar.

I això es justament GangMenu. Una aplicació mòbil-web, col·laborativa, de publicació dels menús diaris dels bars de la zona, perquè puguin estar disponibles per ser consultats.

Una aplicació multiplataforma<sup>2</sup> que permet a l'usuari interessat en trobar els bars de la seva zona on hi ha menú, saber el preu, els plats de que compta, a quina distància estan i quin es el millor camí per arribar-hi. A més al ser una aplicació col·laborativa, tant els bars com els usuaris, poden actualitzar diàriament el menú i fer-se visibles a altres possibles clients.

El funcionament es tan simple com això: un usuari fa una fotografia del menú del bar on està geolocalitzat, posa el preu del menú i la puja a un servidor web. I a partir d'aquell moment ja està disponible per ser consultada per la resta d'usuaris de GangMenu.

Bé. Tan simple, tan simple no és. Hi ha un pas intermig que és escollir d'una llista el bar on ets geolocalitzat i si no existeix, crear-lo, però només caldrà posar el nom del bar, i les coordenades ja ens les donarà la geolocalització del mòbil.

I tot i que a simple vista sembla justament això, simple, com es podrà observar en el desenvolupament del document, hi ha moltes oportunitats de negoci i grans beneficis potencials per les parts implicades en el mateix, des de els usuaris, als restauradors i, evidentment, al desenvolupador.

(2)Multiplataforma

Aquesta aplicació està concebuda per poder funcionar sobre sistemes IOS, Android, Blackberry i Windowsphone. Com es veura més endavant, això ha estat possible gracies a utilitzar Phonegap.

## 2. Estat de l'Art

Actualment vivim en l'inici de la era de les aplicacions mòbils per telefonia inhalàmbrica, i des de fa un any, amb la sortida al mercat de Android, s'han popularitzat més les aplicacions en terminals tàctils.

I dic Android, perquè si bé es cert que els Iphone són terminals que ja feia més de dos anys que existien, no ha estat fins la generalització de Android en que realment hi ha hagut un increment substancial de les aplicacions per mòbils.

La raó es bàsicament perquè mentres que Apple segueix amb la seva política de tenir-ho tot absolutament tancat i de cobrar absolutament per tot, Android ha obert el mercat de les comunicacions 2.0 a la majoria de gent introduint el terme pagar per serveis "premium". Això vol dir que tens la opció d'utilitzar l'aplicació amb funcionalitats reduïdes i si es volen més funcionalitats s'ha de pagar. (Iphone finalment va introduir aquesta formalitat de pagament per evitar la fugida de usuaris a Android).

Aquesta política ha provocat que actualment es parla de que en el market d'Android hi ha aproximadament unes 450.000 aplicacions i en el de Iphone unes 650.000. I tot i que n'hi ha més en el market de Iphone, també fa més temps que existeix, però lo important es que els estudis diuen que per culpa del tancament als desenvolupadors, aquest market s'estancarà.

Són números molt grans i les aplicacions abasten una gran quantitat de opcions englobades en grups com els jocs, les eines d'ofimàtica i qualsevol cosa que un es pugui imaginar. Una de les eines que més genèricament s'ha introduït dins del món de les telecomunicacions, ha estat la geolocalització.

L'inici d'aquest fenomen prové dels antics enrutadors GPS que es tenien els cotxes. Quan van sortir els primers terminals tàctils amb connexió a internet, Iphone ja va intentar crear una mena de navegador GPS amb SatNav, però va acabar ràpidament superat per l'aparició de GoogleMaps.

Google Maps, inicialment només en versió web, es una aplicació de Google que mitjançant la posició en longitud i latitud podia ensenyar-te el carrer on estaves. Després va venir la evolució amb la introducció de Google Earth que era una aplicació en la que podies veure qualsevol punt del món començant desde la visió del satèl·lit fins arribar a un alçada que et permetia veure els carrers.

La següent evolució de la geolocalització va venir amb Google Streetview que combinant les dues anteriors aplicacions et permetia veure el carrer exactament com era en realitat, de manera que podies fer un zoom desde el satèl·lit i arribar a veure el teu cotxe aparcats (això em va sobtar molt).

Finalment Google va donar el cop de gràcia amb la introducció de Navigator als sistemes Android. Google Navigator es un GPS que incorpora totes les funcionalitats dels GPS de cotxe, amb la geolocalització dinàmica, la predicció de ruta, la marcació de punts d'interés i les instruccions de viatge, i a més va incorporar el que havien après a Streetview i a les altres aplicacions, de manera que ara, el sistema et permet crear rutes a peu, en

cotxe o en bici amb la visió real dels carrers.

Tota aquesta tecnologia està basada en GPS (Global Positioning System) o el que es el mateix, sistema global de posicionament (per satèl·lit). Actualment tots els terminals de qualsevol plataforma té incorporat un sistema de posicionament que es pot activar per rebre certs serveis. Això dona resposta un paradigma en la evolució de les aplicacions web. Inicialment les aplicacions web es centraven en el "què?", després es van centrar en el "cóm?", després amb la irrupció de Facebook i les xarxes socials el paradigma va canviar a "amb qui?" i actualment estem en el punt de "on?". I es per això que es tant important la geolocalització en els pròxims anys. (No en va Google ha presentat el seu prototip de cotxe conduït per cecs en el que la tecnologia de conducció està basada en geolocalització GPS i sensors)

Tornant al punt de les aplicacions geolocalitzades, aquesta tecnologia ha provocat que hi hagi moltes aplicacions que en facin ús per diverses fites. Una de les que més força està agafant es Fousquare.

**Foursquare**<sup>1</sup> es una xarxa social per compartir la nostra ubicació, desde el mòbil usant la tecnologia GPS, amb els teus contactes, seguidors o amics. El sistema es simple de funcionament. Si per exemple estem a un museu i volem compartir la nostra posició, agafem el terminal, obrim l'aplicació (que a més es gratis), clickem "check-in" i fet, ja estem localitzables. Si el lloc no es un lloc localitzable, el sistema ens demanarà que el donem d'alta.

Un dels estímuls que Foursquare dona es que es transforma en una mena de joc ja que cada vegada que es fa checkin el sistema ens dona punts que s'acumulen, i si ets el primer en obrir una localització et transformes en el "master" del lloc.

Una de les característiques que Foursquare comparteix amb GangMenu es que es poden posar comentaris dels llocs on estas, de manera que siguin accessibles per a la resta de usuaris de la xarxa.

Un altre de les característiques interessants de Foursquare es que et diu quin dels teus contactes està aprop de tu i això ho aconseguix accedint a la agenda del terminal i revisant tant telèfons com mails.

Per últim, un dels aspectes importants es que te una integració perfecte amb serveis de xarxes socials com Twitter o Facebook i, per tant, quan actualitzem algun comentaris del lloc on estem i directament compartim la nostra posició, aquestes dades son publicades a les dues xarxes socials.

Un altre dels productes que hi ha actualment que també tocaria de forma tangencial a GangMenu seria **Google Places**<sup>2</sup>. Aquesta aplicació creada per Google està encarada a geoposicionar empreses. El seu funcionament es simple. L'usuari que vol donar d'alta una empresa ha de ser el propietari, o la persona autoritzada, de la mateixa. Es dona d'alta i automàticament queda registrada en el sistema.

Abans, quan volies buscar quelcom, per exemple peixateries al barri del Guinardó de Barcelona posaves això directament i amb sort trobaves el que volies. Ara, amb Google places, simplement posant un servei o un producte, l'aplicació et dona un llistat de empreses a prop de la

(1) Logo Foursquare



(2) Logo Google Places



teva posició on hi hagi el que busques.

Tot i que té molt bona pinta, es una d'aquells projectes que no acaba d'arrencar ja que significa un canvi de tendència en el ús que fa la gent dels exploradors web. Encara que segons el Sr. Enrique Dans, es una eina que facilitarà molt les compres en aplicacions posteriors.

I es veritat que es realment útil perquè, si com a l'exemple, estem buscant una peixateria a la zona del Guinardó, la nostra primera opció seria buscar-ho per google, i després geolocalitzar-la per Google Maps. Però ara amb el terminals mòbils i amb Places simplement posant peixateria a l'aplicació mòbil, aquesta ens donarà les peixateries més properes a la nostra posició.

Aquesta aplicació ve preinstalada per definició a totes les versions de Android superiors a la 1.6, i ja està optimitzada per obrir els serveis que calguin automàticament per funcionar correctament.

Seguint amb el tema de la geolocalització, hi ha altres aplicacions que en fan ús i que potser estan més encarades a les xarxes socials, que és un altre dels aspectes importants de GangMenu.

Aprofitant aquesta tecnologia han aparegut serveis com **Scene Tap**<sup>3</sup> que es una aplicació que utilitza càmeres col·locades en locals d'oci nocturn i amb les que pots veure la quantitat de gent que hi ha, el percentatge d'homes i dones i la mitjana de les edats gairebé a temps real.

El sistema funciona amb càmeres col·locades a la porta que compta la quantitat de gent que entra i surt del bar. Un altre càmera analitza les faccions de la cara per veure si son homes o dones i la seva edat aproximada, i d'aquesta manera es crea la informació que es pot veure a la pàgina inicial de l'aplicatiu.

Té integració amb Facebook i twitter de manera que quan decidim anar a algun lloc automàticament queda publicat en aquestes xarxes socials.

Aquesta es una de les moltes xarxes socials temàtiques que han aparegut actualment amb l'aparició de les grans xarxes com Twitter i Facebook, però no es la única ja que actualment han sortit competidors a SceneTap com son **WheretheLadies.at**<sup>4</sup> que es una aplicació on les noies registren i diuen la seva localització i es mostren geolocalitzats els llocs on hi ha més noies de festa per la nit.

Un altre de les aplicacions que utilitzen la geolocalització i les càmeres es **SpotFav**<sup>5</sup>. SpotFav es una aplicació col·laborativa també que està encarada als freaks del kitesurf o qualsevol esport de platja.

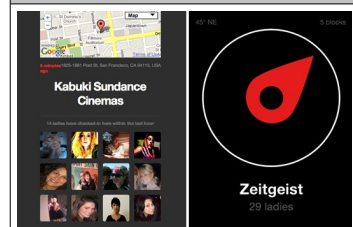
SpotFav ha agafat lo millor de les aplicacions de geolocalització i les xarxes socials i ha creat un producte en el que els usuaris poden postejar les condicions de les platjes on són perquè siguin accessibles als altres usuaris. D'aquesta manera, la gent que vulgui anar a fer kitesurf, només té que obrir l'aplicació i veure, de les platges que hi ha al seu voltant, els comentaris de condicions dels altres usuaris.

També tenen estacions meteorològiques amb càmeres de vídeo que emeten 24 hores al dia i que donen la informació real de les

(3) Logo ScenTap



(4) Logo WheretheLadies.at



(5) Logo SpotFav



condicions del lloc on estan disposades. Aquesta infraestructura dona múltiples oportunitats tant als esportistes com als xiringuitos on hi han les càmeres posades.

Donen oportunitats al restauradors perquè poden promocionar-se dins del món del kitesurf pel fet de tenir una infraestructura que dona servei als esportistes i, per tant, s'estan creant una publicitat que els hi costa ben poc.

Un altre de les aplicacions tangencials del projecte MEUNYU, son les encarades a la restauració i que utilitzen també la geolocalització.

Aquestes aplicacions poden donar una gran quantitat de serveis dels més diversos com per exemple **AquiSi**<sup>6</sup>. AquiSi es una aplicació que va sorgir a Madrid arrel de la llei anti-tabac que el govern va proposar. Com que els restauradors estaven perdent clientela, van decidir obrir un portal en el que, aprofitant la geolocalització dels terminals, es pogués rebre un llistat de bars al voltant en els que es pogués fumar.

En realitat va ser una aplicació que va tenir molt d'èxit fins que la llei es va endurir, i llavors va evolucionar a un llistat de bars amb terrassa exterior on es podia fumar.

Un altre de les aplicacions que es basen en geolocalització i en restauració es **GarrafON**<sup>7</sup>, que es una aplicació que es basa en Google Places i que dona informació sobre els bars que tenim al costat en funció de la nostra geolocalització.

Aquesta aplicació t'ajuda a identificar quin dels bars al teu voltant serveixen Garrafón per poder evitar anar-hi. Està basat en les opinions dels usuaris que classifiquen per bars i per marques de beguda que els usuaris creuen que no son begudes originals. A més ofereix informació sobre els preus del que costen les copes amb el que es creen un llistat dels bars més econòmics de la ciutat.

També té una component social en la que es poden votar els bars en els que estas, de manera que els altres usuaris poden veure quins son els millors. També incorpora una opció premium en la que es pot pujar fotografies del bar.

També hi ha aplicacions en les que s'utilitzen els mateixos recursos però que estan encarats a altres objectius. Es el cas de **MadridMola**<sup>8</sup> que es una aplicació desenvolupada per Mahou en la que es pretén descobrir la ciutat de Madrid fent una ruta per certs bars. L'incentiu es que en els bars hi ha amagats regals i es crea una mena de mega-gincama en la que vas ruta a ruta.

Però no només fa això, sino que MadridMola s'ha transformat en una aplicació que centralitza totes les activitats de lleure de la ciutat de Madrid, i el que va començar com una aplicació publicitària s'ha transformat en una guia de ocio nocturna i diurna que es consultada diàriament per uns 3500 usuaris.

**Kekanto**<sup>9</sup> es un altre aplicació que es basa amb geolocalització, restauració i entreteniment ja que es una guia de bars i hotels accessibles desde el terminal mòbil. Dona una llista prou completa dels hotels que tens al voltant de la teva ubicació i et dona la possibilitat de puntuar i fer comentaris sobre els llocs que has visitat.

#### (6) Logo AquiSi



#### (7) Logo GarrafON



#### (8) Logo MadridMola



#### (9) Logo Kekanto



La aplicació ofereix un llistat dels bars o restaurants amb més puntuació, es poden fer cerques i ofereix la possibilitat de recomanacions per part de l'aplicatiu (que m'imagino que deuen estar subvencionades). I al igual que la majoria també té el servei de publicació de la posició i els comentaris a Twitter o Facebook.

Revisant les aplicacions purament de restaurants, ens trobem amb una que es diu **ElTenedor** que es una aplicació en la que es poden fer reserves a molts dels restaurants de la ciutat.

Tenen una gran quantitat de de bars adscrits i presència a gairebé tota Espanya i França i les xifres d'aquesta aplicació son molt importants, ja que tenen uns 8000 restaurants adscrits entre Espanya i França, han gestionat prop de 5 milions de reserves en un any, donant un negoci brut de 60 milions d'euros als restaurants i actualment estan en fase d'expansió per altres països.

Aquesta aplicació et dona una immensa quantitat de informació i es poden crear filtres en les cerques per preu, per tipus de menjar, per ambient o per les notes que els usuaris posen a cada restaurant.

I a més ha aplicat algunes de les polítiques de pàgines tipus Groupon o Bydeface en les que els restaurants publiquen ofertes en la pàgina principal. De fet, es pot aconseguir descomptes fins al 50% del preu en els restaurants recomanats per la pagina (que també imagino que deuen ser subvencionats).

Actualment el mercat de aplicacions per fer reserves a restaurants està molt dinàmic i hi ha grans grups de inversors que hi estan apostant. Mostra d'això son es aplicacions Restalo o Restorando que tenen el mateix funcionament de ElTenedor i que, com ell, han rebut fa pocs mesos una gran quantitat de inversió per part de fons de capital risc.

Segons Blair MacLaren, uns dels socis del fons de inversió Active Ventures Partners, que han invertit en Restalo: "hi ha una tendència clara cap a la reserva online i aquesta pot generar molt moviment de capital". Active Ventures Partners ha invertit en Restalo uns 2 milions d'euros tot i que els números del portal no són igual de impressionants que en el cas de ElTenedor. Restalo ha generat unes 500.000 reserves realitzades i gairebé 14 milions d'Euros de facturació total generada per els restaurants, amb una expectativa per aquest any d'arribar als 5.000 restaurants i triplicar la facturació.

Està clar que l'usuari el que busca son descomptes i facilitat per trobar la informació, i es per això que es proactiu en les cerques i tendeix a compartir el seu coneixement amb la resta de usuaris perquè d'aquesta manera sap que pot rebre un profit en el cas de que ell també actuï.

S'està creant, com deia el Sr. Punset, una intel·ligència col·lectiva que està canviant la manera en que les persones ens relacionem entre nosaltres. Aquesta intel·ligència col·lectiva, està fonamentada en la facilitat d'accés a la informació i en la facilitat de generar-la. Eines com wikipedia o les aplicacions que hem presentat son una petita mostra de que els sistemes col·laboratius son el present de les telecomunicacions de la mateixa manera que el telefon ho va ser en el ser moment.

Es per això que GangMenu té una perspectiva molt bona perquè incorpora les eines i el funcionament de les millors aplicacions. En concret GangMenu utilitza:

- Geolocalització
- Xarxa social temàtica
- Sistema col·laboratiu
- Està encara a un sector en clar creixement

### 3. Disseny

GangMenu es un aplicatiu mòbil que, com hem dit en la introducció, crea una sèrie d'informació que cal emmagatzemar i que cal gestionar per poder ser accessible per qualsevol persona que la necessiti.

Com que això, per si sol, no es pot fer amb un sol mòbil calen altres mecanismes, o agents, per tal de solucionar-ho.

En particular, GangMenu, consta de tres parts ben diferenciades:

- Una aplicació mòbil, allotjada en terminals mòbils, que centralitza la interacció amb l'usuari.
- Una aplicació web, allotjada en un servidor físic (o virtual), que reculli i gestioni la informació que els usuaris vagin generant en els seus dispositius, i que a més a més, tingui certs serveis online que després detallarem.
- Una API que s'encarrega de fer interactuar ambdós plataformes.

L'estructura d'interconnexió<sup>3</sup> dels diferents agents implicats (que es pot observar al costat d'aquestes línies) implica que al haver-hi plataformes físiques diferents i funcionalitats diferents, també hi hagin solucions de software diferents.

Per tant tindrem que desenvolupar una aplicació web, amb el seu llenguatge corresponent, una aplicació mòbil amb el seu llenguatge corresponent i una API que, en aquest cas, estarà encarada a la comunicació de les dues plataformes.

En els següents capítols es desgrana les necessitats específiques de cadascuna d'aquestes aplicacions i quina es, al meu entendre, la millor solució en quant a software que es pot implementar.

I seguint amb la filosofia d'aquesta memòria, els apartats estan encarats eminentment pràctics de manera que, en primer lloc es descriuran les funcionalitats de les aplicacions, i per tant les seves necessitats; seguidament es descriuran els mockups que resulten d'aquesta informació i finalment el llenguatge que es creu més adient per cobrir totes les necessitats demandades.

Comencem per l'aplicació mòbil perquè és el objectiu final d'aquest projecte i, a mesura que anem aprofundint en el funcionament de l'aplicació, anirem resolent l'aplicació web i la API.

#### 3.1. Aplicació mòbil

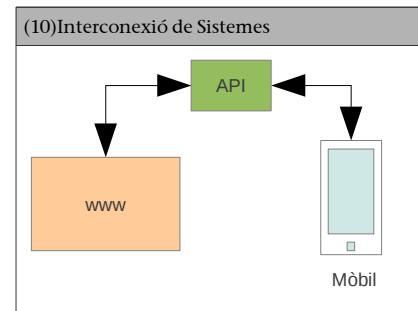
La part mòbil GangMenu es la que interactua amb l'usuari final de forma més propera als objectius inicials de l'aplicació, i es en concret, la part que més ús crec que pot tenir a curt/mig termini.

De fet en la essència del projecte està la immediatesa de "pujar" els menús o la de consultar-los, per tant la immensa majoria dels usuaris ho farà a través dels seus dispositius mòbils.

En aquest cas desenvoluparem una aplicació que pugui ser descarregada i utilitzada per els diferents sistemes operatius mòbils actuals com Android, IOS, windowsphone i blackberry i que doni accés a l'usuari a les diferents funcionalitats del programa.

Hi ha un seguit de serveis, o recursos, que els telèfons mòbils han de donar per tal que GangMenu pugui funcionar correctament. En concret la llista necessària seria:

- Connexió a Internet
- Càmera de fotos
- Accés GPS a geolocalització





Més endavant explicarem la casuística sobre què succeeix si algun d'aquest servis falla, però de moment suposarem que el terminal proveeix de tots i cadascun d'aquests serveis.

Per tant, suposant que els serveis estan actiu i que ja tenim clara la funcionalitat del sistema, quan un usuari es descarrega GangMenu, i l'inicia per primera vegada, l'aplicatiu farà una sèrie de comprovacions que seran generals a tots els casos d'ús.

### 3.1.1. Casos d'ús

Com hem comentat en la introducció, el cas més típic de utilització del programa es per consultar els menús dels bars que hi han a la zona més propera, però no serà la única que podrem trobar. He dividit els casos en els següent:

#### Inici de l'aplicació:

Quan l'usuari s'ha descarregat l'aplicació a través de la seva plataforma, al iniciar-la l'aplicació farà les següents passes:

- Obrir aplicació
- Comprovar si l'usuari està registrat
- En cas negatiu iniciarà el registre de l'usuari que tindrà que introduir un mail i una contrasenya.
- En cas que estigui ja registrat l'aplicatiu mirarà si el servei de GPS està obert.
- En cas negatiu, informará a l'usuari que cal utilitzar GPS i que s'ha de activar, per lo que obrirà la pantalla de connexions inalàmbriques del SO per connectar el GPS.
- En cas afirmatiu, l'aplicació procedirà a la geolocalització del terminal.
- Una vegada geolocalitzat, GangMenu enviarà la informació via internet al servidor de dades web mitjançant un JSON.
- El servidor web, analitza la geolocalització i serveix al aplicatiu unaltre JSON amb el llistat dels bars per ordre de proximitat.
- Aquest JSON es interpretat per GangMenu i serveix per imprimir la primera pantalla de l'aplicació.
- Es serveix la pantalla d'inici.

El diagrama de blocs d'aquesta part de l'aplicatiu es el que es pot veure a la següent pàgina (diagrama de blocs 1)

Una vegada ja hem entrat a la pàgina d'inici<sup>4</sup> l'usuari veu en un llistat dels bars més propers, la distància a la que estan i el preu del menú. També podrà trobar un barra de tasques en la que podrà afegir un bar/menu o sortir de l'aplicació. En concret el que trobarà serà la imatge que es troba la dreta d'aquest text.

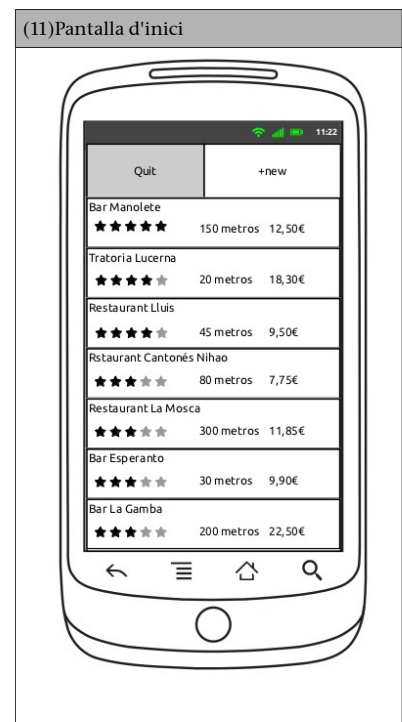
#### Afegir Restaurant:

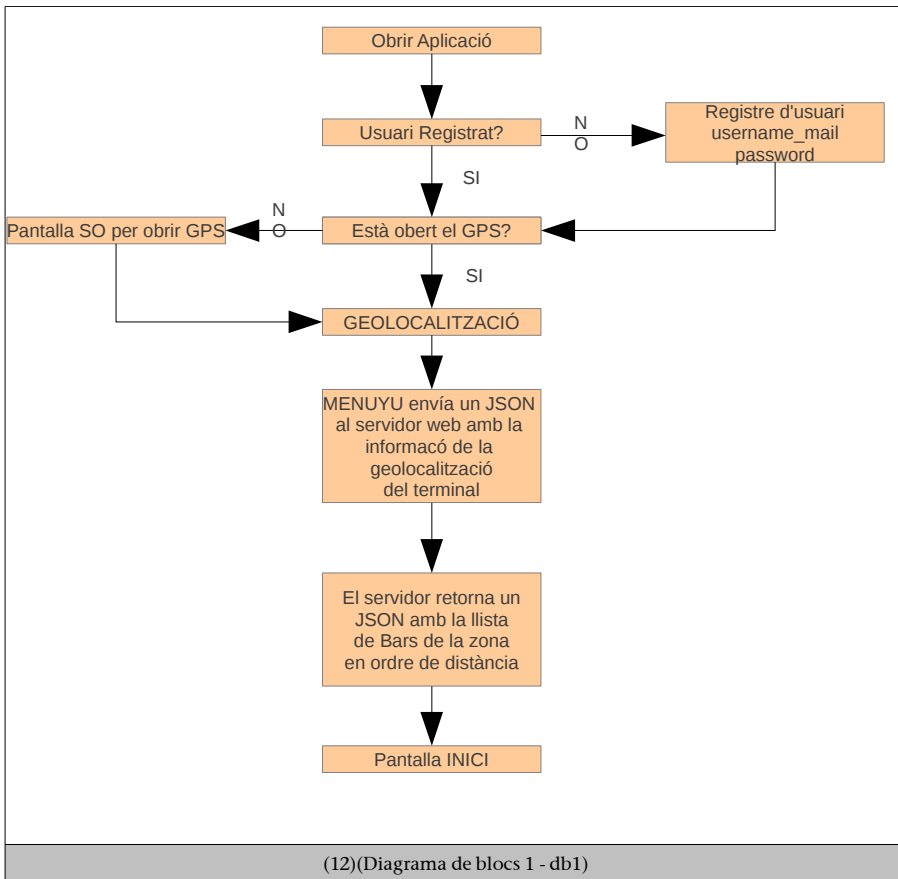
Suposem que anem a un bar i volem compartir el menú. Com que al obrir l'aplicació ja hem dit que el servidor envia un llistat de bars pot haver-hi dues casuístiques:

- que el Bar ja existeixi al llistat
- que el Bar no existeixi.

Suposarem inicialment que no existeix. En aquest cas la seqüència de utilització seria:

- Apretar el botó "+new"
- Automàticament s'accedirà a la pantalla "new Bar" en la que apareixerà un mapa de posició geolocalitzat del bar i un camp on es podrà posar el nom del bar. També hi ha una barra de tasques amb les opcions de sortir o enviar.
- En cas de sortir es retorna a la pantalla d'inici.
- En cas d'enviar l'aplicació ens redirigeix a la pantalla "Review/Add"





- A la pantalla "Review/Add" hi ha una barra de tasques amb les opcions de "Quit", "Back", "GO" (IR) i "COMEBACK" (VOLVER) que s'explicaran en el següent cas d'ús. També hi ha un desplegable en el que especifica què pots afegir: "Picture" o "Comment". Un mapa ens posiciona al lloc on som i, com que es nou, no hi hauran llistat de menús (que s'explicarà mes endavant).
- Es selecciona afegir fotografia
- GangMenu inicia la càmera del terminal
- En fer la fotografia es presenta una pantalla amb la fotografia del menú presa i amb un camp on es pot omplir el preu del menú. També hi ha una barra de tasques amb les opcions "Back" o "Send"
- En cas de seleccionar "Back" es retorna a fer la fotografia
- En cas de seleccionar "Send" s'encapsula tota la informació que s'ha generat (nom restaurant, geolocalització, fotografia del menú i el preu) i s'envia al servidor web mitjançant un json.
- El servidor incorpora la informació a la seva base de dades i ja està llesta per ser consultada.

Entenem que s'ha de completar el registre de bar i menú perquè no té sentit registrar un bar sense informar del menú que hi ha, i es per això que forcem al usuari a fer la fotografia del menú a la vegada que es crea el restaurant

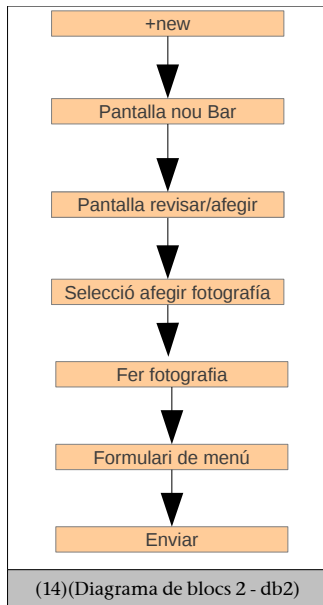
El diagrama de blocs d'aquest cas es el que es pot observar a la pàgina següent:

**Consulta de Menú:**

En aquest cas, l'usuari el que vol es consultar l'aplicació per tal de saber quins son els bars més propers i quins menus tenen.

- L'usuari vol consultar els menús que hi han per la zona
- Obre l'aplicació
- A la pantalla d'inici, on hi ha tot el llistat de bars, selecciona un bar.
- Al obrirse la pantalla de revisió d'informació del bar te varies opcions.





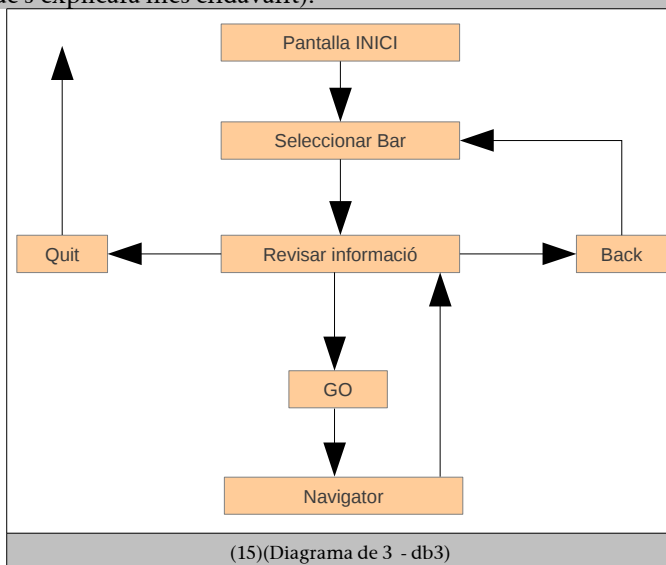
- Si vol continuar buscant, prem "Back" i retorna al llistat de bars de la pàgina principal
- Si vol ser guiat per arribar-hi te la opció de premer "GO" i s'obrirà el navegador per defecte del SO, que en cas d'Android és "Google Navigator"
- Una vegada s'ha arribat al lloc amb Navigator l'aplicació retorna a la pantalla de revisió de informació per si es vol afegir un comentari o una fotografia del menú actual.
- Per últim, en cas de voler sortir de l'aplicació, en la pantalla de revisió d'informació hi ha un botó per sortir.

El diagrama de blocs d'aquest apartat es el que es pot observar a la següent pàgina (db3).

**Afegir Menú:**

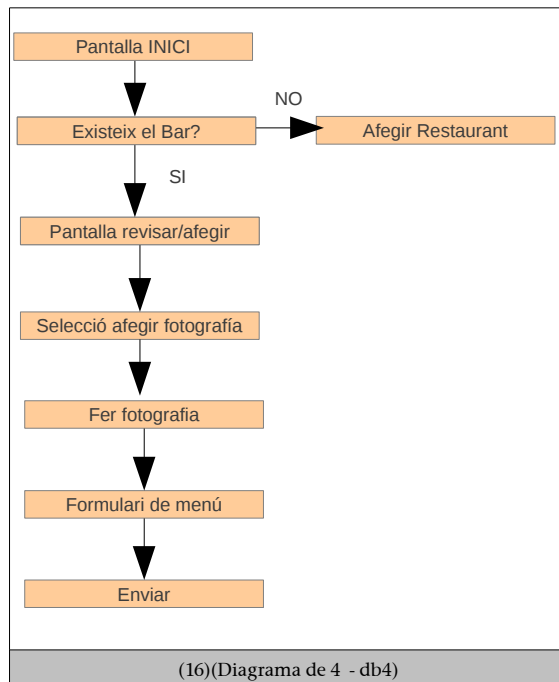
En el cas d'afegir Menú és molt semblant al cas de afegir bar, amb la única diferència de que el Bar ja està creat. I Les passes que es tindrien que seguir son:

- A la pantalla d'inici es selecciona el bar del que es vol "pujar" el menú.
- A la pantalla "Review/Add" hi ha un desplegable en el que es pot seleccionar què pots afegir: "Picture" o "Comment". Un mapa ens posiciona al lloc on som i, com que es nou, no hi hauran llistat de menús (que s'explicarà mes endavant).



- Es selecciona afegir fotografia
- GangMenu inicia la càmera del terminal
- En fer la fotografia es presenta una pantalla amb la fotografia del menú presa i amb un camp on es pot omplir el preu del menú. També hi ha una barra de tasques amb les opcions "Back" o "Send"
- En cas de seleccionar "Back" es retorna a fer la fotografia
- En cas de seleccionar "Send" s'encapsula tota la informació que s'ha generat (nom restaurant, geolocalització, fotografia del menú i el preu) i s'envia al servidor web mitjançant un json.
- El servidor incorpora la informació a la seva base de dades i ja està llesta per ser consultada.

El diagrama de blocs (db4) d'aquest procés seria el següent:



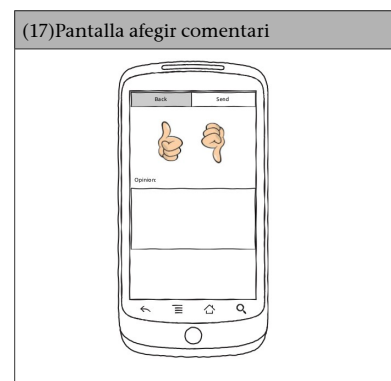
**Afegir Comentari a un Bar:**

Com que entenem que el fet de ser un sistema social ha de permetre que la gent valori el bar en el que està, GangMenu incorpora un sistema de rating que també es mostra a la informació concentrada de la pantalla d'inici, i també uns comentaris que es mostren en la pantalla de revisió/add del bar en qüestió.

A la dreta d'aquest text<sup>6</sup> es mostra el mockup de la pantalla de rating on es pot observar que es pot seleccionar "ok" o "no ok" que estan en forma de polzes, i també es pot omplir un requadre amb els comentaris que es creguin oportuns.

La intenció d'aquests comentaris es que la gent pugui anar auto-regulant la informació no correcte que es generi en la aplicació, de manera que si algú fa us fraudulent i posa alguna informació o correcte la mateixa massa social s'encarregarà de contrarestar aquesta informació.

Un dels casos més típics que voldria evitar es la competència deslleial entre bars. Amb aquesta aplicació es molt fàcil posar mal ratings a bars de la competència, però els usuaris podran contrarestar la informació amb contra-comentaris. A més la intenció es revisar els comentaris negatius amb un robot informàtic que pugui discernir entre comentari real i comentari personal per desgastar la fama del bar. Però aquesta implementació sobrepasa l'abast d'aquest projecte i es una de les millores que es volen implementar una vegada posat en marxa el sistema.



Per últim explicarem què es el que crec que ha de succeir en cas de no funcionar algun dels serveis que estan pressuposats que implementen els terminals, i quines les accions que es poden fer.

#### **No hi ha internet:**

Com a servei encarat directament a Internet el terminal ha de tenir connexió a Internet per poder utilitzar l'aplicació. Per tant informará al usuari de que cal connectar-se a Internet per poder completar l'ús de l'aplicació. Aquesta informació es farà en format alerta de Javascript en el que se l'hi permetrà a l'usuari continuar utilitzant el software o sortir de l'aplicació.

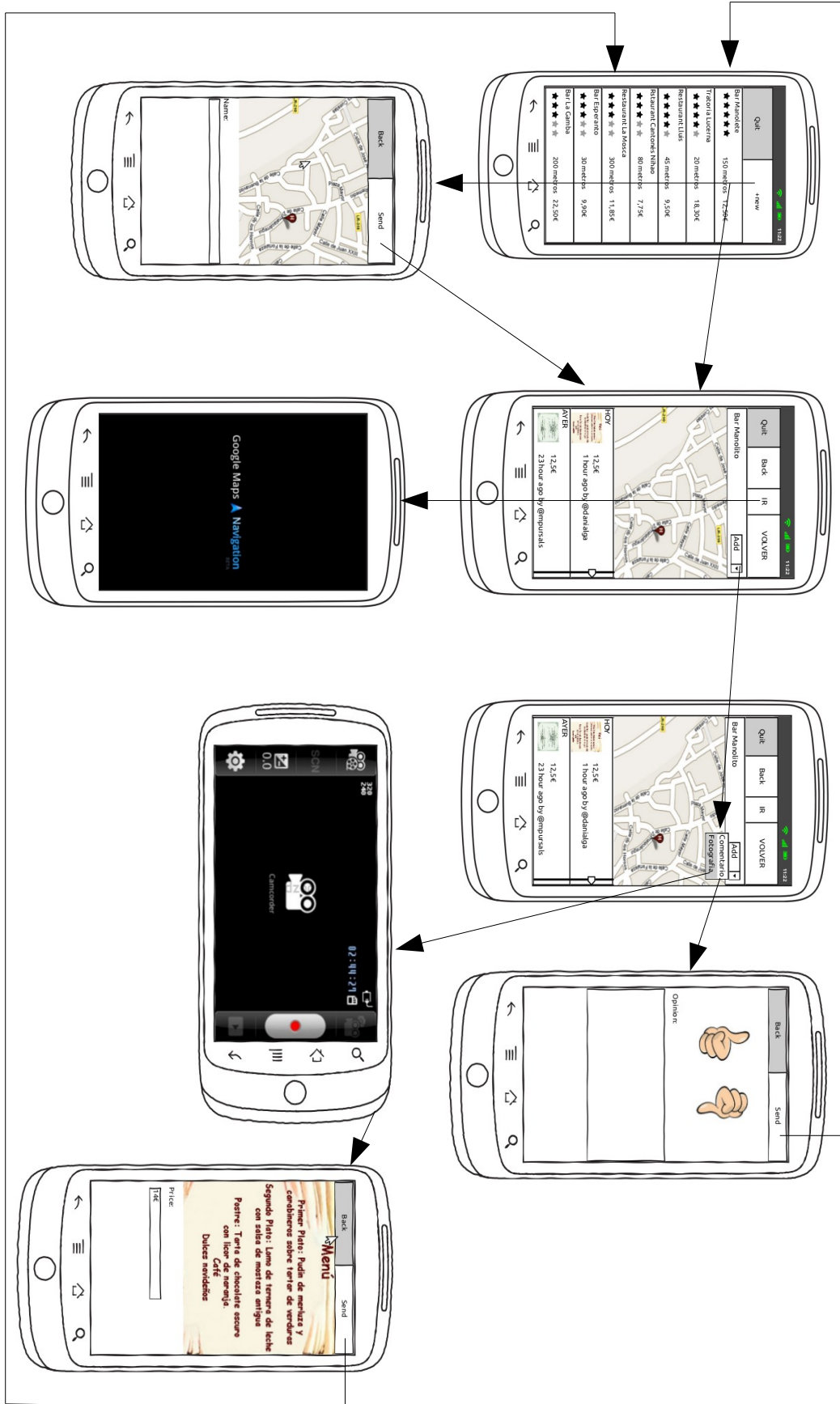
En cas de continuar utilitzant l'aplicació, aquesta ensenyará l'últim llistat de bars que es manté en memòria del telèfon, que no es res més que el JSON que ha entregat el servidor la última vegada que s'han connectat amb el terminal.

Aquest JSON amb total la informació del servidor, estarà emmagatzemat en Local i serà fàcilment accessible per l'aplicació per restaurar la informació o per modificar-la.

#### **No hi ha càmera o no funciona bé:**

Aquest cas es important en el moment en que es vol accedir a la opció de "pujar" un menú doncs, com hem vist anteriorment, es requisit necessari tenir càmera. Però no es imprescindible pel funcionament general de l'aplicació. Per tant, en cas de que a l'hora de seleccionar afegir fotografia, si no hi ha càmera, l'aplicatiu advertirà al usuari de que la operació no es pot efectuar degut a que no existeix càmera i, automàticament, retronará a la pantalla d'inici.

3.1.2. Mockups



Tal i com es pot observar, a la pàgina anterior hi ha un mockup de la funcionalitat de l'aplicació. En aquest mockup es poden observar totes les funcionalitats abans descrites això com les interaccions entre cadascú dels estats de l'aplicació.

Hem de recordar que aquesta aplicació està encarada als nous terminals amb tecnologia tàctil per lo que cada opció d'interacció està pensada per fer-se directament sobre la pantalla, amb lo que es divideix l'espai en botons que actuen al ser seleccionats.

En la primera pantalla, pantalla d'inici (adalt a l'esquerra) es veu el llistat dels bars més propers disposats per ordre de proximitat al usuari. En cadascuna de les caselles dels bars es pot observar que hi ha un conjunt de informació important que sintetitza el que l'usuari necessita: el nom del bar, la distància a la que es troba, el preu del menú d'avui i uns rating d'estrella que marquen la qualitat del lloc. Apart també s'observa una botonera amb les opcions "Quit" que tanca la aplicació, i la opció "+new" que et dona pas a crear un nou Bar, com més tard explicarem.

Si s'escull algun Bar, clicant sobre cadascun dels Bars, la següent pantalla mostra una botonera amb les opcions "Quit", "Back", "GO" (IR) i "COMMENTS" (Comentarios) que ja han estat explicats anteriorment, un plànol amb la posició exacte del Bar, i un llistat dels menús que algú ha pujat al sistema. Aquest llistat és temporal lo que significa que es mostren els últims menús per d'aquesta manera tenir actualitzada la informació que es mostra.

L'apartat Comments ens deriva a una pàgina en la que hi ha un llistat dels comentaris que la gent ha anat posant sobre aquest bar. Si s'escull IR, automàticament s'obre el navegador per defecte del sistema operatiu i ens guia fins al destí.

Hi ha un desplegable "Add" on es pot escollir afegir una de les dues opcions: Picture o Comment. En la opció Comment, ens porta a unaltre pàgina on podem posar un rating bo o dolent al Bar i, si volem, també deixar un comentari. En enviar el comentari, l'aplicació retorna directament a la pàgina inicial.

En la opció Picture, l'aplicació en obre la càmera del terminal per fer la fotografia al Menú. Una vegada feta la fotografia ens envia a un formulari que mostra la fotografia del menú que acabem de generar i un camp on podem posar-hi el preu.

Finalment, a la hora d'enviar el formulari amb la fotografia que hem fet i el preu, l'aplicació retorna automàticament a la pàgina principal.

I amb això acabem la part de com volem que es comporti la nostra aplicació segons el disseny que hem creat.

### 3.1.3. Descripció del Framework i del llenguatge de programació.

Una de les coses que primer s'ha de tenir en comte a l'hora de desenvolupar una aplicació mòbil, es el fet de que cada sistema operatiu té les seves particularitats i els seus paquets de desenvolupament, que rarament son compatibles els uns amb els altres.

De fet, normalment, els grans proveïdors de SO ja procuren que els seus sistemes estiguin "tancats" a desenvolupadors que no els utilitzin de forma habitual. D'aquesta manera, per exemple, un desenvolupador en IOS tindrà que tenir, per força, un sistema OSX per poder-ho desenvolupar. I això, com veurem mes endavant, es una limitació que ens ha portat a la elecció del sistema sobre el que desenvolupar GangMenu.

Actualment existeixen molts Frameworks<sup>7</sup> amb els que es poden fer aplicacions per telefonia mòbil, però la majoria son encarats a alguna de les plataformes imperants actuals. D'aquesta manera trobem Frameworks encarats al desenvolupament en Android, encarats a IOS, a Windowsphone o a Blackberry.

Evidentment m'interessava utilitzar un framework que fos compatible amb totes les plataformes actuals i per tant, ja directament, vaig descartar les que no m'oferien aquesta possibilitat.

També cal esmenar, que per les característiques del disseny i les especificacions de la aplicació, el que més se l'hi ajustava era un dels coneguts com MWA o "Mobile Web Application" que no es res més que una aplicació web que corre sobre un miniservidor. Això queda empaquetat en un arxiu que es fàcilment exportables a qualsevol plataforma.

El problema inicial amb aquesta mena de tecnologia era que, al ser una aplicació que s'executava en l'explorador web natiu al terminal, depenia molt de les seves capacitats. Però ara com ara, els MWA ja poden perfectament accedir al hardware i gestionar-lo sense problemes i la potència dels terminals fa que la diferència entre un sistema natiu i un encarat a web sigui mínima en quant a velocitat i accés.

De Frameworks n'hi han molts, però he fet un recull amb una petita explicació del perquè no vaig triar-los per desenvolupar GangMenu.

#### (7)Framework

En general un Framework, quan es parla de desenvolupament de software, es un conjunt de mètodes, classes i una estructura pensada per tal de poder desenvolupar una aplicació. D'aquesta manera el codi que en C++, JAVA o un altre llenguatge seria molt costós de programar, amb el Framework adequat se simplifica molt.

### Rhodes

Es un Framework desenvolupat per "Rhomobile" que es multiplataforma, o sigui, que es compatible amb qualsevol sistema mòbil actual. Té suport per GPS, geolocalització, càmera, acceleròmetre, etc... Està encarat per fer aplicacions natives reals i no "mobile web apps" per lo que el llenguatge es una mica més complex que en altres opcions. Està basat amb Ruby i és l'únic del mercat que es capaç de treballar amb models MVC per desenvolupament d'aplicacions.

### IUI

Inicialment era un framework encara bàsicament a sistemes compatibles per IOS com els Iphone o Ipad, però en els últims mesos han començat a crear mètodes que puguin ser compatibles per altres plataformes. Es un dels que creen el que s'anomenen "mobile web app", que no son més que pàgines web sobre un miniservidor web, tot empaquetat en un arxiu i que son autoexecutables en els terminals. Aquest és a més basat en javascript i CSS.

### Iwebkit

Es un framework per a la creació d'aplicacions tàctils per Iphone i Ipod touch. No té suport per altres sistemes operatius.

### XUI

Aquest es un altre MWA basat en javascript per construir simples aplicacions mòbils. No està gaire documentada i es per això que la vaig descartar automàticament.

### Titanium

La gracia de Titanium es que es tracta d'un framework open source per desenvolupament d'aplicacions natives que també es pot utilitzar per fer aplicacions d'escriptori basades en tecnologia web. Està creat per Appcelerator i si bé te una gran quantitat de funcionalitats, considero que es una mica massa gran per el tipus d'aplicació que aquest projecte es planteja. Però en qualsevol cas es una excel·lent framework molt vàlid per altra tipus d'aplicacions.

### Sencha Touch

Incialment era un sistema només encarat a Iphone, però actualment ja dona supporta a pràcticament totes les plataformes actuals. Basat en CSS3 i HTML5 desde fa un parell de mesos, es complementa amb javascript i es molt



optimitzable. També es dels que desenvolupen "MWA" i si l'hagués conegut abans possiblement l'hagués utilitzat.

## Jquery Mobile

Aquests dos últims frameworks son els que més em van agradar. Jquery està orientat a dispositius mòbils i suporta IOS, Android, Windowsphone, Blackberry, Symbian... Està basat en HTML5, CSS3 i la veritat es que ja està en funcionament després de bastant temps de desenvolupament i, ara com ara, s'està començant a menjar el mercat. El problema de Jquery es que ha arribat tard respecte del seu gran competidor, i el que s'utilitza en aquest projecte.

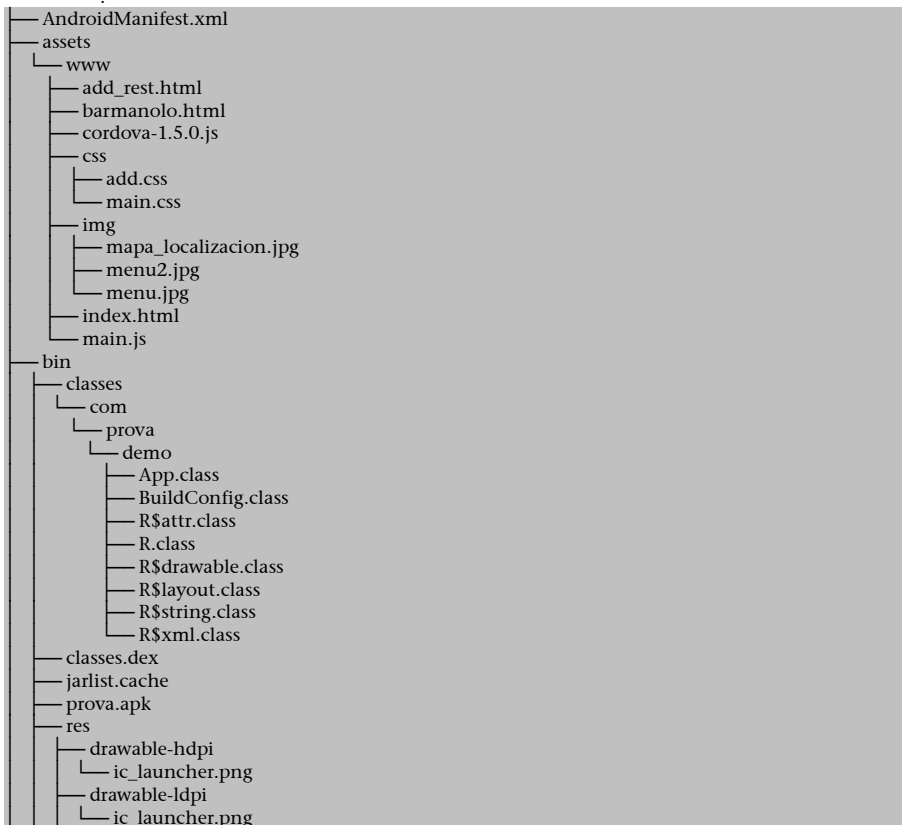
## Phonegap

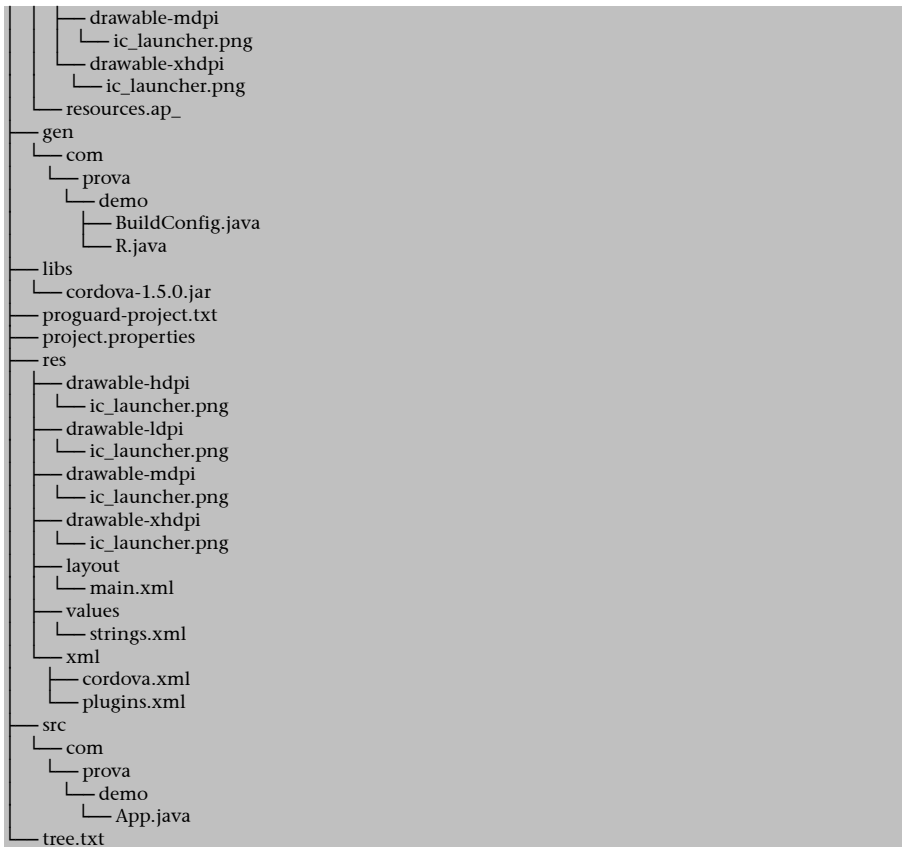
Phonegap, amb poc temps, s'ha transformat amb un stándart en el desenvolupament d'aplicacions mòbils. Està basat en HTML5, CSS3 i incorpora suport per javascript i, en la última versió, també incorpora algunes de les crides de Jquery Mobile. D'aquesta manera unifica les dues grans plataformes de desenvolupament de software per mòbils Phonegap i Jquery Mobile.

És compatible amb tots els sistemes operatius actuals com IOS, Andriod, Blackberry, Symbian... però el que el fa més interessant es que la companyia APACHE, que es la companyia més important en software per servidors web, l'ha absorbit fa poc, amb lo que Phonegap s'ha transformat, de facto, amb el standart dels desenvolupadors que a més son administradors de sistemes Linux.

Això dona garanties de que el desenvolupament del framework sigui homogeni amb les noves versions d'apache i més encara, que el miniservidor web, que aquestes MWA porten dins, sigui desenvolupat per ells.

Phonegap es un sistema amb una jerarquia de directoris que ja ve tipada desde la primera aplicació que es desenvolupar. No es estrany veure en els frameworks que els directoris i alguns arxius ja estiguin prefixats, però en el cas Phonegap potser es més restrictiu. Això es així per minimitzar les les cerques de documents i optimitzar les reaccions dels exploradors web natius. La estructura es la següent:





Quan es crea un nou projecte amb Phonegap es creen automàticament la majoria dels directoris i dels arxius que es mostren en l'anterior esquema. Per desenvolupar l'aplicació web cal crear un sub-directori *www* dins del directori *assets* i dins d'aquest crear la nostra aplicació amb HTML5, CSS3, javascript i JQuery.

La única llibertat que es dona es a l'hora de decidir quin recull de funcions de javascript podem utilitzar. Com sabem podem obtenir per la red varis reculls de funcions de javascript que poden ser cridades desde l'aplicatiu principal. Phonegap, amb la versió 1.5, que es amb la que estem actualment desenvolupant l'aplicació, incorpora una sèrie de arxius .js anomenats Córdoba que inclouen la majoria de les funcions necessàries per controlar el terminal i desenvolupar l'aplicació, però a més es possible incorporar altres paquets que puguin complir les nostres expectatives.

Com a bon framework per al desenvolupament de web es programa amb HTML5, CSS3 i javascript i es poden utilitzar totes les tècniques heretades en el desenvolupament de pàgines web. A part, també incorpora una sèrie de funcionalitats que permeten, entre altres:

- accedir a geolocalitzacions a través del GPS del terminal
- utilitzar la càmera
- Utilitzar l'acceleròmetre
- Utilitzar els sensor de presió
- etc...

### 3.2. Aplicació web

L'aplicació web es la part del projecte que dona suport a la majoria de les funcionalitats de GangMenu, doncs es la que gestiona directament totes les dades que es van incorporant des de els terminals, la que serveix la informació demanada i la que gestiona tots els usuaris. Es per això que podem dir que l'aplicació web es el cor del projecte.

El funcionament de l'aplicació des de el punt de vista del servidor, comprén 3 aspectes bàsics:

- El primer es com interfície web per als usuaris de PC
- El segon es un servidor que emmagatzema tota la informació que s'ha generat en els diferents terminals mòbils.
- El tercer es un backend per poder gestionar l'aplicació i les dades que genera.

### 3.2.1. Cas d'ús

#### Interfície/aplicació web:

Es possible, que un usuari no tingui l'aplicació instal·lada al mòbil o que, directament, no tingui un smartphone per poder connectar-se a l'aplicació.

Però també pot passar que un usuari vulgui trobar un restaurant a prop del lloc on, previsiblement, estarà demà, cosa que l'aplicació mòbil, en principi, no contempla.

Per solucionar aquests problemes, GangMenu ha de ser accessible als usuaris de PC també. I aprofitant la opció de geolocalització que té intrínseca HTML5, aspecte bàsic en el concepte de l'aplicació, això es factible.

Les funcionalitats que han d'estar operatives en ella son:

- Poder registrar-se i accedir com usuari
- Veure les últimes actualitzacions de menús
- Buscar un restaurant
- Enviar Menú

L'usuari que es connecti a GangMenu.com es trobarà amb la pantalla P1 la dreta d'aquestes línies.

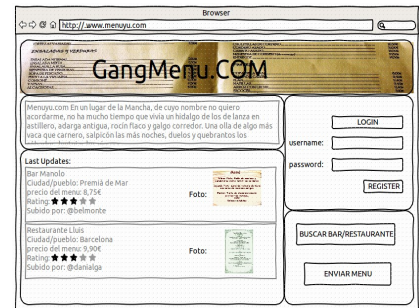
**Registrar-se i accedir com usuari:** Per registrar-se, al igual que amb l'aplicació mòbil, l'usuari tindrà que posar el seu username, que serà un mail i escollir un password.

La seguretat es un tema molt important quan es tracta de dades personals i e per això, que GangMenu, implementarà un sistema de encriptació per tal de mantenir les dades segures. Aquesta encriptació es farà a nivell de software amb Ruby on Rails, que ja té un opció per tal de escollir el sistema d'encriptació de les dades.

Aquesta informació anirà encriptada a una base de dades d'usuaris, creada amb MongoDB (de la que parlarem més endavant) i que es guardarà en un servidor de bases de dades (també més endavant parlarem de la infraestructura del projecte).

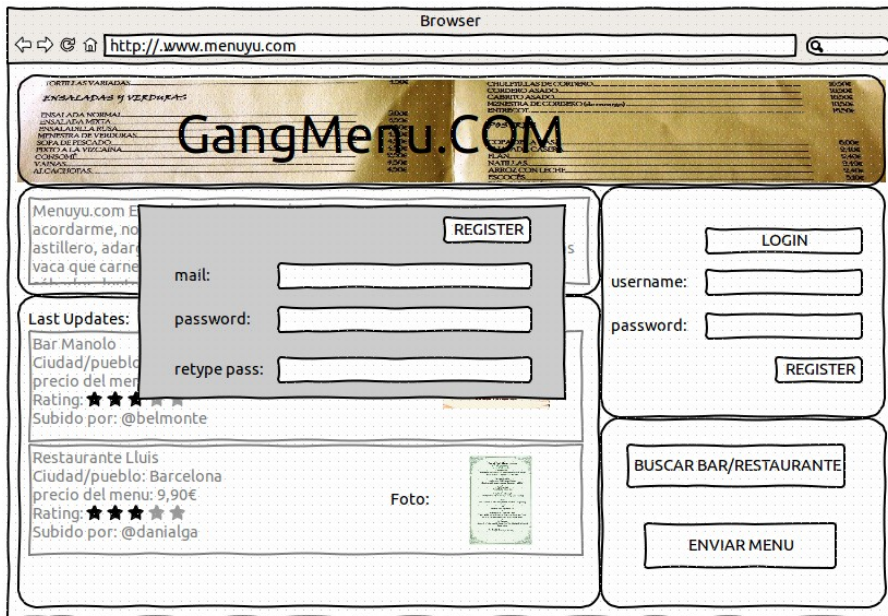
La política per la qual es registrarà l'aplicació en quant a tipus de dades son les següents:

- Tipus d'usuaris: Els usuaris tindran que ser mails verificables amb el format user@domain.com/es/... En cas de que el usuari no introdueixi un mail correcte saltarà una alerta en la que se l'hi demanarà que torni a introduir el mail.
- Tipus de passwords: Els passwords, per donar més seguretat, es forçarà a que siguin, per defecte, paraules que continguin:
  - al menys una lletra majúscula
  - al menys 2 xifres numèriques
  - mínim de 8 dígit fins a un màxim de 16
 Per exemple: Mes37ada



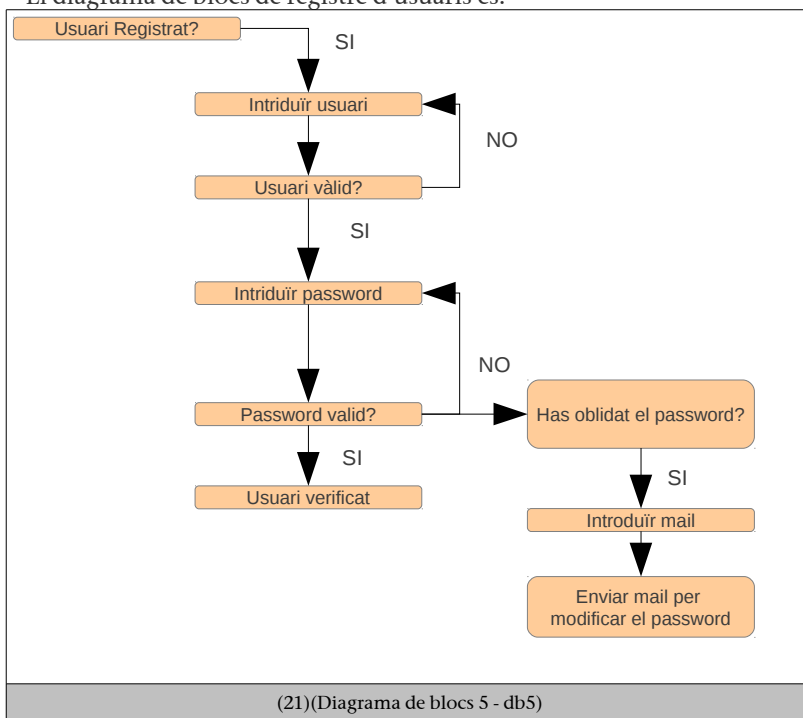
(19)

P1 – Pantalla d'inici de l'aplicació web  
P2 – Pantalla de registre



Una vegada registrat l'usuari, se l'hi enviarà un mail en el que tindrà que verificar que les dades que ha introduït són correctes i accedir a un link pel qual, l'aplicació, podrà verificar que el mail aportat es realment vàlid.

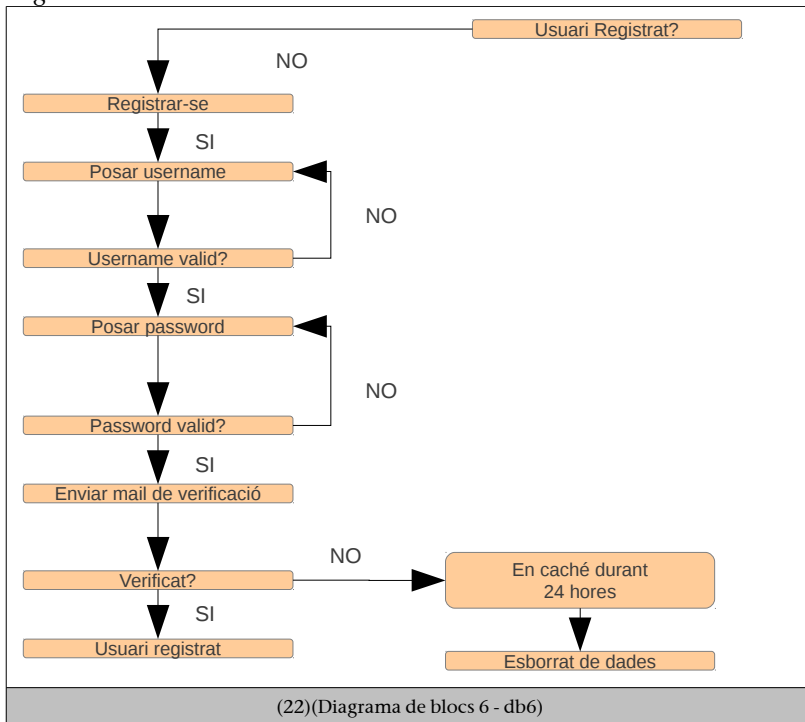
El diagrama de blocs de registre d'usuaris es:



Per accedir com a usuari, l'aplicació demanarà, a través d'un script en Ajax, les dades de usuari i password i les verificarà amb les que té emmagatzemades a la base de dades. En aquest aspecte, i per augmentar també la seguretat i minimitzar el risc de intrusions malignes provocades per aranyes robotitzades, l'aplicatiu només permetrà 7 intents fallits abans de bloquejar l'accés a login a aquella IP.

En cadascuna d'aquestes proves se l'hi demanarà a l'usuari si ha oblidat la contrasenya. En cas afirmatiu, l'aplicació demanarà el mail registrat i enviarà al usuari un link amb la direcció URL de la pàgina per modificar el password.

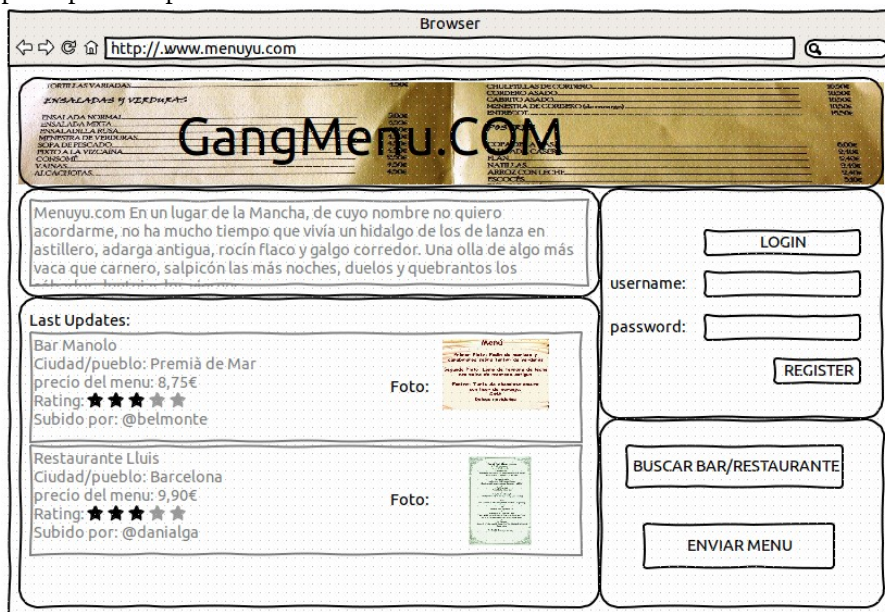
El diagrama de blocs de accés d'usuaris es:



**Veure les últimes actualitzacions:** Com es pot observar per el mock up de la pàgina anterior, la primera pàgina es compona de varies àrees. Una d'elles es un streaming a temps real de les actualitzacions que els usuaris van incorporant a l'aplicació.

Això significa que, quan un usuari fa una fotografia amb el seu mòbil, posa el preu i ho puja al servidor, aquesta informació queda automàticament registrada a la base de dades del sistema i, automàticament, s'imprimeix en el timeline de l'aplicació web.

Per tant, un usuari, encara que no estigui registrat, que vulgui observar les últimes actualitzacions les podrà observar directament desde el front-end principal de l'aplicació.



P3 - Patanalla d'inici de l'aplicació web / Time Line Streaming

**Buscar restaurant/bar:** Un usuari desde el PC que vulgui buscar un bar/restaurant tindrà varies opcions.

La primera opció serà introduir la direcció on vol buscar i la població. Aquestes dades generaran una informació amb els bars/restaurants més propers a la localització que ha introduït. Aquesta cerca serà obligatòriament amb registre i accés de l'usuari, per poder fer estadístiques d'ús en un futur.

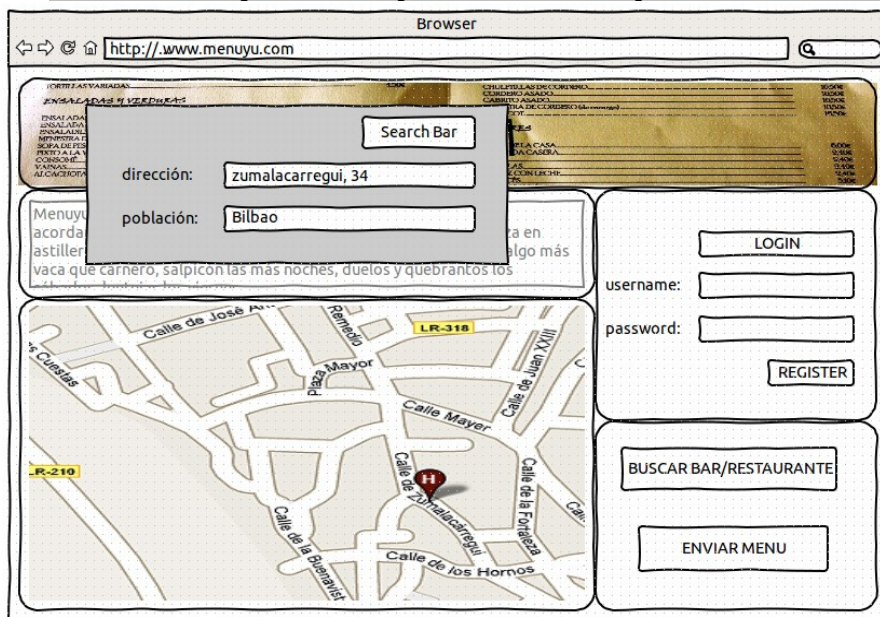
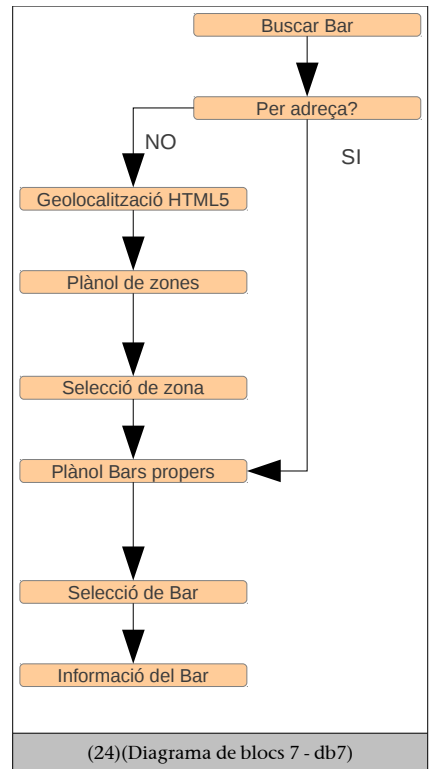
La segona opció que l'usuari tindrà, i aprofitant la opció de geolocalització de html5 dona, seria sense introduir cap direcció. En aquest cas, l'aplicació funcionaria aproximadament igual que en la versió mòbil on la geolocalització filtraria el llistat de bars més propers.

Tot i que a dia d'avui, la geolocalització per telefonia mòbil es molt precisa, en el cas de IP d'equips estacionaris no ho és tant, doncs aquesta depèn de la informació que el ISP comparteixi amb les bases de dades metropolitanas. Però en general, l'error en la aproximació no supera els 500 metres.

Per solucionar això, en futures modificacions del projecte, he pensat en imprimir el plànol per àrees de 500 metres més properes i que el usuari pugui seleccionar l'àrea exacta en la que es troba, de manera que només s'imprimeixi l'àrea que es d'interés.

Una vegada s'hagi seleccionat el Bar, es presentarà a l'usuari una pàgina amb la informació següent:

- Plànol exacte de la localització del bar
- Llistat de menús que han estat incorporats en l'ultima setmana
- Comentaris que els usuaris posin sobre el Bar en qüestió



P4 - Patanalla de cerca de Bar per direcció

**Enviar Menú:** Una de les opcions que crec que serà més interessant de cara a capitalitzar l'aplicació, es el fet que els propis restauradors puguin penjar el menú del dia.

Tot i que la aplicació ha de tenir per força el fet social i col·laboratiu perquè pugui funcionar, s'ha de compaginar amb que els propietaris puguin facilitar la informació al col·lectiu pujant-la ells. Això, encara que sembli un oximoron, pot facilitar molt l'ús de l'aplicatiu, perquè el dia que els clients no pugin el menú, ho pot fer el restaurador o viceversa i, en definitiva, els restauradors que coneguin l'aplicatiu, seran els primers en voler promocionar-se.

En un principi, aquesta funcionalitat, tindria que ser gratuïta, però a la llarga es una de les peces importants per fer que el projecte sigui viable econòmicament.

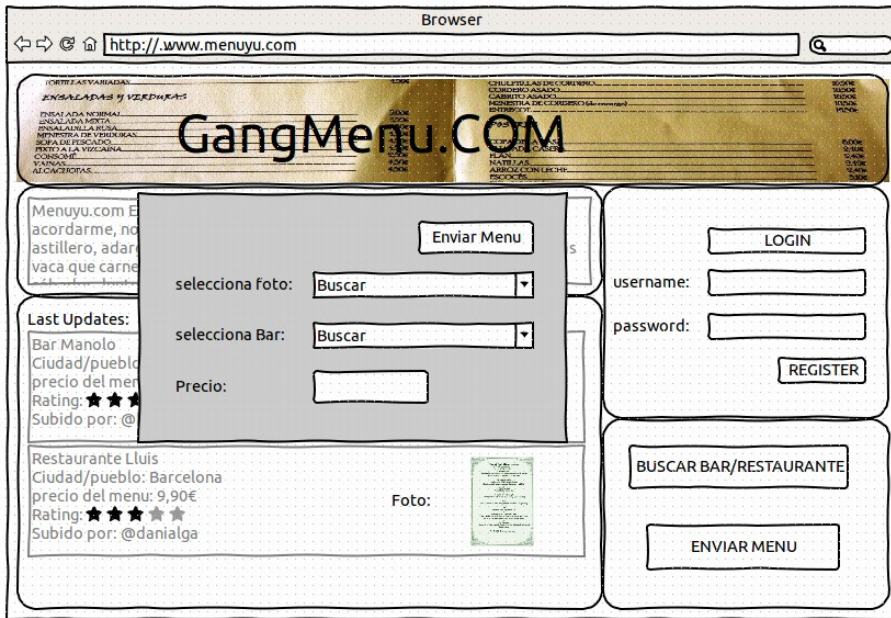
El funcionament seria molt simple.

Una vegada registrat a la pàgina web, un usuari pot demanar la propietat d'un dels restaurants, i després de les comprovacions adients, es pot assignar un "place" a un usuari.

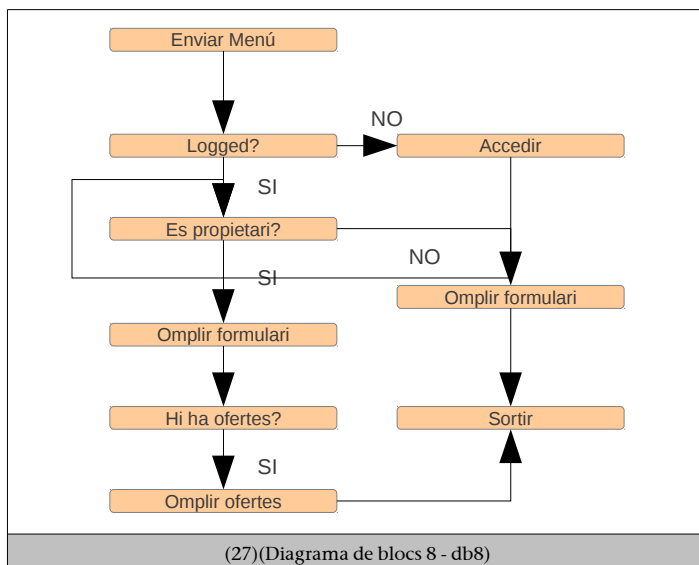
A partir d'aquest moment, el propietari del bar pot actualitzar el menú del seu bar/restaurant fent una fotografia al menú i omplint les altres dades. També podrà posar ofertes perquè els seus possibles clients puguin beneficiar-se'n, el que l'hi provocaria un augment de la popularitat en la xarxa.

En cas de que no siguis el propietari i, per qualsevol raó, no has pogut pujar la fotografia amb el mòbil, també es tindrà la possibilitat de pujar la informació des de el PC.

Cal destacar aquí, que Ruby ja té una gemma per poder redimensionar les imatges que es pugin al servidor, i homogeneïtzar-les, de manera que, independentment de la qualitat amb la que es faci la fotografia del menú, s'intentarà que totes tinguin la mateixa. Evidentment, si la fotografia no en te gens no es poden fer miracles, però com que la majoria dels terminals fan fotografies de gran qualitat, un gran percentatge de les vegades el que tindrem que fer es reduir-les.



P5 - Patanalla de enviar me



**Interfície del Back-End:** El back-end, per contraposició amb el front-end, es la part de darrera de l'aplicació. O sigui, Es una part de l'aplicació a la que tenen només accés els administradors de l'aplicació i que, en general, només es fan funcions de gestió de l'aplicació i de les dades que hi ha allotjades.

En conseqüència, el front-end d'una aplicació, es la part que tots els usuaris poden veure i interactuar. O sigui, la part que està publicada en el servidor i que te les funcions que hem descrit abans.

En el nostre cas, el back-end fa funcions clarament de gestió administrativa de l'aplicació. En concret es poden tractar dos grans grups:

- Gestió d'usuaris:
  - Donar d'alta o baixa d'usuaris
  - Donar privilegis als usuaris
  - Associar o desassociar "places" amb usuaris
  - Bloquejar usuaris
- Gestió de dades:
  - Editar/eliminar dades
  - Editar/eliminar comentaris
  - Revisar les dades
  - Crear estadístiques

### 3.2.2. Descripció del Framework i del llenguatge de programació

De la mateixa manera que hem descrit el llenguatge amb el que hem programat l'aplicació mòbil, farem el mateix amb el llenguatge de l'aplicació web.

Si bé en el cas de l'aplicació mòbil, la raó per la qual s'escull el framework es per la compatibilitat que hi ha amb les diferents plataformes de programació, en el cas de l'aplicació web, el llenguatge a escollir es un tema de matisos i de preferències.

Hi ha una infinitat de llenguatges de programació de pàgines web i fer un resum exhaustiu de tots implicaria no acabar mai aquest projecte, i com que no es l'objectiu principal fer la història de la evolució de la informàtica d'Internet, només esbossaré un petit resum dels llenguatges que crec que han estat cabdals pel desenvolupament d'Internet.

Un dia la meva avia em va preguntar que perquè hi havia tanta polèmica amb qui havia descobert Internet, pregunta que em va sobtar perquè sincerament m'esperava més que em preguntés què era i no qui la va inventar. Però de fet, les dues preguntes tenen relació i responent a la segona es respon la primera.

"Iaia, Internet es una barreja de dos conceptes: què s'envia i com s'envia. De la mateixa manera que hi ha les autopistes, les carreteres nacionals i les carreteres rurals, i cadascuna té les seves característiques que la fan diferent, Internet també té diferents camins pels quals es pot enviar informació.

Aquestes "carreteres" son les que van inventar els americans cap al 1970.

Però la informació que s'enviava a través d'aquelles carreteres, no estava gaire estructurada i va ser un senyor que es deia Tim Berners-Lee, cap al 1989, el que va inventar un llenguatge amb el que es podien estructurar la informació perquè tothom la entengués... va neixer HTML i la polèmica de si va ser primer l'ou o la gallina."

Després d'aquesta anècdota, absolutament real, que descriu el que crec que es el començament de la era de la informació, hi han hagut uns quants llenguatges que han ajudat a que la comunicació a través d'Internet fos més eficient, millorant característiques de l'anterior.



**HTML:** *Hypertext Marckup Language* Es un llenguatge encarat a les etiquetes que s'utilitza per descriure l'estructura i el contingut de pàgines web en forma de text. En aquest llenguatge es poden incorporar imatges, taules, emmagatzemar dades i crear scripts (subprogrames escrits en altre llenguatge que aporten funcionalitats al codi principal).

És la base de totes les pàgines web i, fins el dia d'avui, absolutament tots el llenguatges o frameworks encarats a web que han aparegut posteriorment, es basen en HTML per desenvolupar les seves capacitats.

**JavaScript:** Es un llenguatge de programació encarat a web i interpretat i orientat a objectes. S'utilitza bàsicament per dotar de funcionalitats extres a HTML i ha tingut una gran acceptació perquè s'executa en els navegadors dels clients, el que permet millorar molt el rendiment de certs efectes.



La sintaxi es semblant a C i incorpora certes característiques de JAVA, però encara que ho sembli, no es un a evolució ni una derivació del llenguatge encarat a objectes JAVA.

Actualment es un altre estàndard en la programa, gairebé tots els frameworks o llenguatges moderns encarats a web fan ús de les seves funcionalitats.

Va ser desenvolupat per Brendan Eich que treballava a Netscape, i va ser incorporat al navegador cap a desembre del 1995. Amb aquesta irrupció, es va crear un binomi que ja es quedaria immutable: HTML+JAVASCRIPT.

HTML i JavaScript, van estar funcionant sols durant molt de temps i a finals dels 90, altres llenguatges com JAVA, PERL o PHP, tots basats en C++, complementaven aquest binomi amb altres funcions que fins llavors no eren possibles.

Tots aquests llenguatges, que son encarats a objectes, tenen una màxima que es DRY (Don't Repeat Yourself) de manera que el codi que funciona es pot reutilitzar en un altre aplicació o per un altre usuari. Així, van aparèixer els Frameworks, que son un conjunt de mètodes o classes ja prefixades i una forma de programar amb una sèrie de normes, que tenen com a base algun d'aquests llenguatges anteriors més HTML+JavaScript, com per exemple:

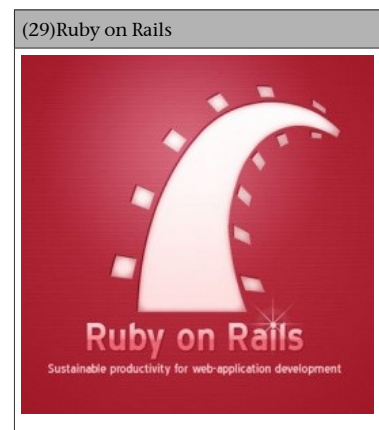
- Mojolicious: que es un framework per desenvolupar pagines web amb PERL+HTML+JavaScript, y es semblant a altres frameworks com Catalyst o Dancer
- Akelos: que es un framework per desenvolupar aplicacions amb PHP+HTML+JavaScript, i es similar a Symfoni o a PHP on TRAX

Molts dels frameworks moderns incorporen un paradigma de programació gens nou que es el MVC, Model-View-Controller, que es una lògica de programació en que es separen les dades, la interfície d'usuari i la lògica de l'aplicació en tres parts diferenciades. D'aquesta manera la vista seria la pàgina en HTML+Javascript+CSS, el model es el sistema de gestió de bases de dades i el controlador es el que rep els events des de la vista.

Hi han molts llenguatges que treballen amb aquest paradigma i tots encarats, com a mínim, a pàgines web com PHP, Phython, Java, Perl i Ruby, i es aquest ultim es que utilitzarem per fer l'aplicació web.

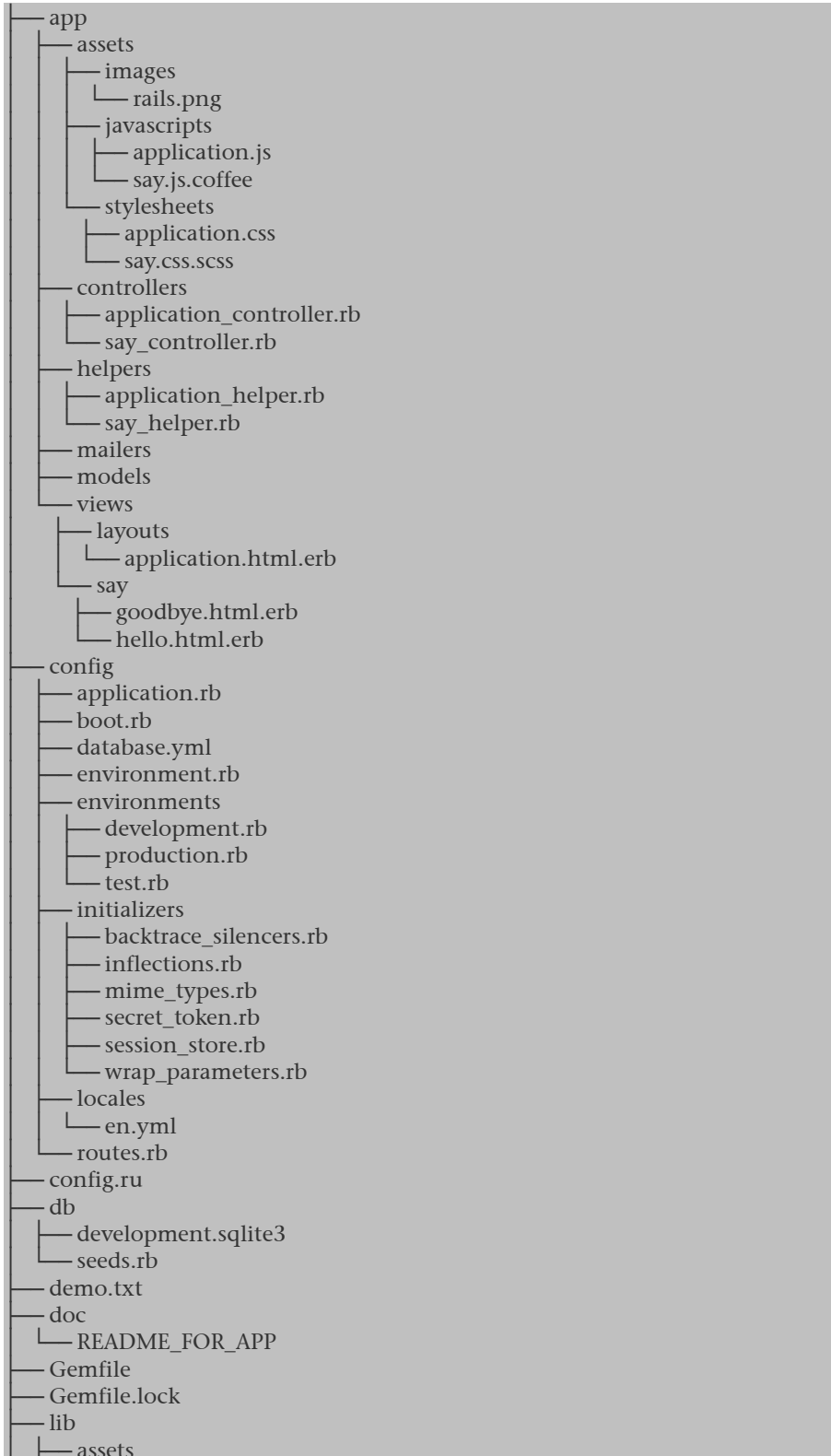
**Ruby on Rails:** Ruby es un llenguatge de programació interpretat, orientat a objectes creat per Yukihiro Matsumoto cap al 1993 i està basat en C, en Perl i Phython. Compleix amb l'ús del paradigma de programació MVC i amb la programació orientada a objectes i es distribueix sota llicència GNU/Linux.

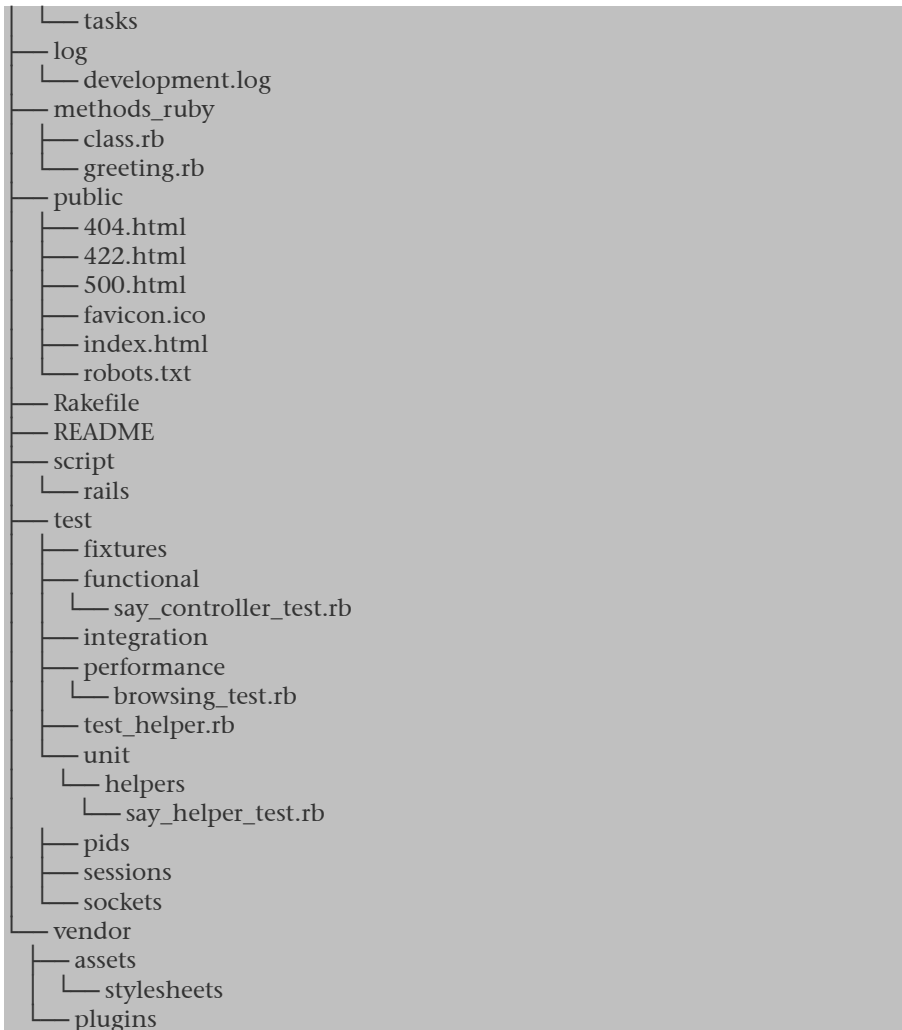
Ruby es la base del framework de programació Ruby on Rails que simplifica molt la programació de pàgines web ja que el framework fa moltes de les funcionalitats per si sol, de manera que les operacions son interpretades i desenvolupades pel mateix sistema enlloc de que el programador tingui que desenvolupar-les per ell mateix.



Això simplifica moltíssim la programació i facilita la rapidesa a la hora de desenvolupar pàgines web. Molts dels mètodes o de les funcions que en qualsevol altre llenguatge significarien hores de picar codi, amb Ruby on Rails, es simplifiquen a una única instrucció.

El preu que s'ha de pagar es la rigidesa a la hora de definir nomenclatures, variables i altres funcions, així com on s'han de col·locar físicament els arxius que s'han de generar. La estructura d'una aplicació amb RoR es la següent:





RoR permet instal·lar pluggins al framework mitjançant RubyGems, que son petites aplicacions que contenen totes les funcionalitats per executar un pluggin, com per exemple mysql2 gem es una gemma que configura automàticament la interrelació entre l'aplicació i la base de dades mysql.

Nosaltres utilitzarem RoR 3.2 amb les gemmes de configuració de MongoDB, Passenger per comunicar-se amb Apache i Capistrano per fer el volcat i posta en marxa de l'aplicació.

### 3.3. API

La API es la tercera pota del projecte GangMenu i es gairebé la part més important del sistema complet. Es així perquè es una porta a les funcionalitats que l'aplicació web i la clau per la interacció entre els terminals mòbils i els servidors.

Es molt important també perquè permet independitzar els servidors web i de dades, de l'aplicació mòbil de manera que aquesta no té accés directe al codi font.

Això es molt útil, per exemple, en cas de que volguéssim exportar tots els nostres contactes de gmail a twitter o facebook. En aquest cas Facebook o Twitter es connectarien a la API de gmail per poder accedir a les dades dels nostres contactes pero sense introduir-se en l'aplicació gmail pròpiament dit.

Es justament per aquesta funcionalitat de independència entre aplicacions que han de treballar juntes, en el que veig una gran oportunitat de incrementar la seguretat de la informació que hi han al servidor web o al de dades.

Un altre de les raons per les quals es interessant la utilització de la API, es perquè altres aplicacions que vulguin aprofitar alguna de les funcionalitats que he desenvolupat ho podran fer sense tenir que posar-se en contacte amb nosaltres doncs hi haurà tota la informació exacta dels procediments i del codi per poder fer-ho.

Actualment totes les grans aplicacions web tenen la seva pròpia API que els desenvolupadors aprofiten per incorporar certs aspectes d'aquestes en els seus desenvolupaments. Per exemple:

1. API de Youtube: permet escollir els videos i posar-los en una pàgina externa.
2. API de Facebook: permet, entre moltes altres coses, externalitzar el time-line, posar botons de me gusta o share, login amb el usuari de facebook...
3. API de Twitter: permet posar el botó de tweet, compartir articles o informació i visualitzar el time-line de l'usuari
4. OpenGL (Open Graphics Library)defineix certes funcionalitats per aplicacions que generis gràfics en 2 i 3D

N'hi moltes altres i aquesta utilitat es el que ha permés que es creïn les anomenades mashups, que no son res més que aplicacions web que fusionen serveis de altres aplicacions web, com per exemple <http://www.findthestream.com/> que es un mashup que conjuga les APIs de Spotify, Youtube, LastFm, Twitter i Facebook.

En el nostre cas, una de les moltes funcions que tindrà la API, es que s'encarregarà de gestionar totes les dades que l'hi arribin dels terminals mòbils a través de JSON, i enviar la informació de dades al servidor de base de dades i les imatges al disc dur del servidor web.

### 3.3.1. Funcionalitats de la API

Com he comentat anteriorment, la API esta basada en la interconnexió entre el terminal mòbil i el servidor web i la base de dades i que l'intercanvi de informació entre elles es farà amb JSON.

Per ser més exactes, aquest intercanvi es fa amb una variant de JSON que es diu JSONP que permet a Phonegap salvar la limitació de cross-site. D'aquesta manera Phonegap intercanvia informació entre diferents sites de forma "externa" sense tenir les limitacions d'accés a la base de dades de l'aplicació web.

Les funcionalitats que implementaré en aquesta fase del projecte son:

#### Create a user

```
Path: (informació reservada)

Parameters

email (mandatory) - Email for the new user
password (mandatory) - Password for the new user

Responses

# On success
{"status":"success","mobile_token":"d10f00ca4c1151038131c58d1bf675e1608fcde91c9586ab61"}
# On error
{"status":"error","msg":"Email is already taken, password is too short..."}
```

Es pot observar que Phonegap tindrà que enviar un JSONP al path descrit amb el mail i el password per POST, i que la resposta del servidor serà un mobile-token si no hi ha error, i un missatge d'error en cas de que hi hagi alguna anomalia en la informació enviada.

## Login

```

Path: (informació reservada)

Parameters

email (mandatory) - Email for the new user
password (mandatory) - Password for the new user

Responses
# On success
{"status":"success","mobile_token":"d10f00ca4c1151038131c58d1bf675e1608fcde91c9586ab61"}
# On error
{"status":"error","msg":"Invalid email or password"}

```

Aquest cas serveix per accedir a la aplicació en cas de haver fet logout o en cas de utilitzar un nou terminal. En condicions normals, quan s'ha creat l'usuari a partir de Create User, el mobile-token es queda emmagatzemat el memòria per tal de que, al obrir l'aplicació, el sistema es connecti automàticament sense necessitat de fer login.

## Get locations around

```

Path: (informació reservada)

Parameters
token (mandatory) - User's mobile token
lat (mandatory) - Latitude
lng (mandatory) - Longitude

Responses
# On success
{"status":"success","locations":[[{"4eb94b13185476197c1a8849","Barcelona"},
{"4eb94b12185476197c1a87f4","Mataró"}..]}
# On error
{"status":"error","msg":"Invalid latitude or longitude"}

```

En aquest cas el que fe m es enviar per POST, la informació de geolocalització que el terminal ens proporciona juntament amb el token d'usuari. Això es fa una vegada iniciada la aplicació, de manera que el que es retorna, es un llistat de pobles o zones properes a la nostra posició indexats per distància.

## Get nearby places

```

Path: (informació reservada)

Parameters
token (mandatory) - User's mobile token
lat (mandatory) - Latitude
lng (mandatory) - Longitude

Responses
# On success
{"status":"success","places":
[{"id":"4fb67ed65973744036000008","name":"barpepito","distance":"0km"},
{"id":"4fb6970b597374534e000003","name":"Suite","distance":"0 km"}]}
# On error
{"status":"error","msg":"Invalid latitude or longitude"}

```

En aquest cas el que estem fent es, amb la mateixa info que abans, rebre un JSON amb la informació dels bars del voltant per distàncies.

## Create a place

```
Path: (informació reservada)

Parameters
token (mandatory) - User's mobile token
lat (mandatory) - Latitude
lng (mandatory) - Longitude
name (mandatory) - Place name

Responses
# On success
{"status":"success","msg":"Place BAR pepe created
successfully","place_id":"4fb67fbe597374403600002b"}

# On error
{"status":"error","msg":"Name can't be blank, location can't be blank, incorrect
latitude or longitude..."}
```

Amb create place, el que farem serà enviar la informació de token, geolocalització i el nom del bar i ens retornarà un missatge de ok amb el id del nou place creat, en cas de no error.

## Show a place

```
Path: (informació reservada)

Parameters
token (mandatory) - User's mobile token
place_id (mandatory) - Place ID

Responses
# On success
{"status":"success","place_info":{"name":"Bar Pepe ","position":
{"lat":41.5393016,"lng":2.4460071},"region":"Catalunya,
Barcelona","user":"mpursals","menu_info":
{"menu_price":"9,95euros","menu_date":"12/06/2012","menu_img_url":"http://
gangmenu.com/img/2874903772993739"},"menu_info":
{"menu_price":"8,25euros","menu_date":"11/06/2012","menu_img_url":"http://
gangmenu.com/img/287490377ffjkf487n281"}}}

# On error
{"status":"error","msg":"Place not found"}
```

Aquesta iteració, ens dona tota la informació necessària per poder imprimir en pantalla els menús de un place així com tota la resta de la informació.

### 3.4. Interrelació entre els sistemes

El funcionament dels sistemes entre si serà molt simple i es pot dividir en dos grups: enviament d'informació entre servidor-API-terminal mòbil i el sentit invers. Per comunicar-se entre ells utilitzarem JSON.

JSON (JavaScript Object Notation) es un format lleuger de intercanvi de dades entre aplicacions que es molt facil de entendre per part del desenvolupador i per les màquines es simple de interpretar-ho i generar-ho. Està basat en JavaScript.

JSON es un format de text que es completament independent del llenguatge, però que conté estructures semblants a les que hi pot haver a C/C++, Java, Perl, Python i altres, i gracies a això es un llenguatge perfecte per l'intercanvi de dades.

Està constituït per dues estructures

- Una col·lecció d'objectes o hash
- Una llista ordenada de valors o matrius de dades.

I en la nostra aplicació les estructures estan dividides de la següent manera: per cada bar {geolocalització, nom, comentaris, rating}, per cada menú {fotografia, preu}.

A més d'aquestes característiques, JSON es important i bàsic en la nostra aplicació perquè està incorporat de forma nativa a JavaScript, i com que Phonegap no es res més que un framework de HTML5+JavaScript+CSS3, JSON es la forma natural de comunicar-se amb Phonegap. (Tot i que s'han de fer certs trucs per saltar-se la limitació de REST cross-domain)

Hi ha moltes maneres de fer un intercanvi de dades amb un servidor. Molts utilitzen crides Jquery.Ajax 'POST' per fer-ho, però el procediment que GangMenu fa es el següent:

1. El client (terminal mòbil) envia la geolocalització a través d'una petició GET de la següent forma:  
[http://gangmenu.com/\(informació reservada\)?Lat=123,4&Lng=456](http://gangmenu.com/(informació reservada)?Lat=123,4&Lng=456)  
De manera que introdueix la geolocalització on estem en la capçalera de la api.
2. Quan es servidor rep la informació de geolocalització, selecciona els bars per distància, des de els que estiguin més aprop fins als que estiguin més lluny, amb un màxim de 1000 metres a la rodona. Empaqueta tota la informació per ordre en un JSON que envia al client.
3. El client resp la resposta, converteix els objectes JSON en variables locals JavaScript i les imprimeix en el DOM.

Una possible implementació de la funció seria aquesta:

```
function getBarDetails() {
$.ajax({
type: 'POST',
url: <Server URL>,
data: {
'Client': 'Location'
},
dataType:'json',
success: function(data) {
var barObject = eval('(' + JSON.stringify(data) + ')');
var barObjectLen = barObject.spotDetails.length;
for (var i=0; i<barObjectLen; i++) {
// Get the data in local JS variables
var barId = barObject.barDetails[i].barId;
var barName = barObject.barDetails[i].barName;
var barLocationMap = barObject.barDetails[i].barLocationMap;
var barMenuPicture = barObject.barDetails[i].barMenuPicture;
var barMenuPrice= barObject.barDetails[i].barMenuPrice;
var barRating= barObject.barDetails[i].barRating;
}
},
error: function(xhr, type) {
alert('Internal Error Occoured!');
}
});
}
```

### 3.5. Facebook i Twitter

Una de les funcionalitats que crec que serà interessant es la integració amb altres xarxes socials com Twitter o Facebook.

La idea es, aprofitant que les APIs de totes dues aplicacions permeten postejar a través d'aplicacions externes, donaré la opció de, una vegada un usuari hagi pujat un menú, posar-ho a twitter o facebook, de manera que se sàpiga on està dinant i quin es el menú del lloc on està dinant.

Això es una opció més de les que es poden beneficiar els usuaris de restaurants que utilitzin la nostra aplicació, perquè es publicitat gratuïta exponencial i sense esforç.

Unaltre de les opcions relacionades amb les xarxes socials, seria la de importar tots els contactes que els usuaris tinguin a gmail, facebook o twitter per tal de recomanar l'aplicació, i augmentar així els usuaris potencials i la publicitat natural de la xarxa.

### 3.6. Descripció tècnica

En aquest apartat especificarem les característiques tècniques necessàries per poder implementar l'aplicació GangMenu.

Com que la gràcia del projecte es entendre el que cal per desenvolupar un projecte integral, el que farem serà gestionar tots els recursos enlloc que externalitzar-lo.

D'aquesta manera, muntarem els servidors, les bases de dades, les polítiques de seguretat, les polítiques de backup... Per tant en aquest apartat parlarem tant de software com hardware, les característiques i perquè he pres les decisions que he pres al respecte.

#### 3.6.1. Bases de dades

Les bases de dades son grups de informació que estan emmagatzemades en un servidor o en un PC. Normalment aquestes dades estan estructurades de forma coherent i tenen unes especificacions concretes, per lo que no es possible emmagatzemar en una base de dades res que no estigui predeterminat.

Cada base de dades es pot subdividir en una o N taules que a la seva vegada, estan subdividides en tuples i cada tupla correspon a un llistat concret de dades. Per fer una analogia simple, i gens acurada, seria com les taules del Calc o Excel de tota la vida.

Cadascuna de les files, o tupla, conté una serie d'informació que esta detallada en la creació de la taula, i cada columna correspon a aquesta especificació.

A part, cada taula pot tenir elements en comú amb altres taules de la mateixa base de dades, i per això es molt important fer un bon disseny de les taules i dels dominis, per tal de que la interrelació entre elles sigui el més eficient possible, i es puguin recuperar dades molt ràpidament. Vamos que, com diria la meva avia, les BBDDs son la madre del cordero.

En el nostre cas, he creat una base de dades amb 5 taules interrelacionades, i una 6a taula que està destinada a gestionar els accessos al back-end dels administradors.



Taula 1: Usuaris:

user_ID	username	password
1023	<a href="mailto:mpursals@gmail.com">mpursals@gmail.com</a>	DJjsl32o0
1024	alexei.solarin@gmail.com	dkdo32od9

En aquesta taula d'usuaris, s'emmagatzemen l'identificador d'usuari, el username i el password, que estarà encriptat amb una de les gemmes de Ruby. En aquest cas la clau seria el user\_ID.

Taula 2: Geoposició:

bar_ID	Nom	Geoposicio
b722	Bar-Eto	2455,55/789,36

A la taula de Geoposició, s'emmagatzema l'identificador del bar, el nom i la geoposició. I les claus que son accessibles desde les altres taules son la geoposició i el bar\_ID.

Taula 3: Menú

bar_ID	menu_ID	user_ID	Preu	foto_ref
b722	m435	1023	12€	file://...

A la taula de Menú s'emmagatzemaran les dades relatives a bar\_ID, menu\_ID, user\_ID, preu i foto\_ref. La foto\_ref es necessària perquè no emmagatzemarem les fotografies a la base de dades, sinó que l'emmagatzemarem en una partició del servidor web i serà referenciada. En el moment que s'hagi de enviar una fotografia, el servidor llegirà la direcció, agafarà la fotografia i l'adjuntarà al json.

Taula 4: Comments

bar_ID	user_ID	menu_ID	Comment
b722	1023	m435	Este bar es muy bueno y la relación calidad-precio es genial

A la taula Comments, s'emmagatzemen el bar\_ID, user\_ID, menu\_ID i el comentari. S'emmagatzemen totes aquestes variables relacionades amb el comentari perquè d'aquesta manera ens permet relacionar el comentari amb la resta de variables i així recuperar-les quan calgui.

Taula 5: població

nom_poble	Geoposicio
Mataró	2455,55/789,36

Aquesta taula es una relació directe entre la geoposició i el nom de la població. Està extreta de google-maps i es el que ens servirà per poder seleccionar els bars per poblacions.

Taula 6: Administradors

admin_ID	username	password	mail
1023	<a href="#">marcpursals</a>	DJjsl32o0	marcpursals@gmail.com
1024	alexei.solarin	dkdo32od9	alexei.solarin@gmail.com

Aquesta es la taula que gestiona i emmagatzema els usuaris que tenen accés al back-end de l'aplicació. Al igual que en el cas del usuari, la contrasenya està encriptada per una gemma de Ruby. En canvi, el username no serà un mail. Això es així perquè, si bé pot haver-hi diferents administradors de l'aplicació, no té perquè haver-hi diferents mails de contacte, sobretot en cas de que l'aplicació envii un log. Ens pot interessar que una serie d'administradors tinguin un mail d'incidències genèric i uns altres un mail específic, i es per això que independitzem les dades.

Tot i que l'esquema anterior es claríssimament el d'una base de dades SQL Relacional, el projecte no està implementat en aquest tipus de base de dades, i ara explicaré perquè, i perquè he posat aquest esquema si teòricament no el faig servir.

### 3.6.1.1. Descripció de les diferents alternatives

En el món de les bases de dades s'ha sempre ens havien dit que els tipus que n'hi havien eren:

- Relacionals
- Orientades a objectes
- Relacionals orientades a objectes

Però a la pràctica, s'han acabat imposant les bases de dades Relacionals i l'estandart per fer les consultes que era SQL. Com a exemple de bases SQL que son molt utilitzades son: SQLite, MySQL, PostgreSQL o Oracle.

El concepte de base de dades relacional es que es creen taules dins d'una base de dades i que, en aquestes taules hi ha una serie d'elements comuns que permeten diversificar la informació (cosa que es molt bona a l'hora d'organització), i ajuntar-la per trobar relacions entre les diferents dades.

La veritat es que el concepte sempre ha estat brillant, però tenen una serie de inconvenients que han fet que els desenvolupadors estiguin migrant a altres tipus de BBDD. Aquests inconvenients són bàsicament la duplicitat de dades en les taules (com es pot observar en el meu disseny de taules), i molt poca flexibilitat a l'hora de la escalabilitat.

Respecte a la escalabilitat es evident que, a mesura que augmentaven les dades a les taules en quant a tuples o dominis, cada vegada era més difícil trobar informació arribant a ser normal fer tres i quatre JOINS en una query. En definitiva, quan més grans son cada vegada més complicades de entendre.

Es per això, que no fa gaire, han aparegut una sèrie de bases de dades que, precisament, milloren molt l'aspecte de la escalabilitat, la velocitat d'accés a les dades i eliminen la redundància de dades, les NOSQL BBDD.

Si tinguéssim que resumir les característiques comuns d'aquests sistemes son tres: ausencia d'esquema en els registres, escalabilitat horitzontal i molta velocitat.

La ausencia d'esquema significa que les dades no tenen una definició

d'atributs fixe com en les bases de dades relacionals, o sigui que cada Document (registre) pot tenir una informació diferent cada vegada, o sigui, que tenen formes diferents. Això també facilita que es puguin emmagatzemar estructures de dades complexes en un sol document, com per exemple emmagatzemat una de les taules anteriors en una de les tuples d'un altre taula, de manera que es tenen totes les dades relacionades amb el mateix bloc en un sol registre.

Respecte a la escalabilitat horitzontal significa que per cada registre es poden afegir dinàmicament més nodes indicant al sistema quins son els nodes disponibles. De manera que es fa molt fàcil la interacció amb els registres i l'enteniment del disseny.

Per ultim, la velocitat depén del fet que molts d'aquests sistemes treballen directament amb la memòria física de l'ordinador i només volquen les dades al disc cada cer temps, el que agilitza molt l'accés a la informació i, en conseqüència, la rapidesa del sistema.

Com a inconvenient, i relacionat amb aquesta última característica, en cas de fallida del sistema entre els moment de volcat de dades al disc, com que les dades estan només emmagatzemades a la memòria física es perdrien. Per tant té moltes avantatges però alguna desavantatge que les fan plantejables però sabent el que es fa perquè impliquen cert risc.

### 3.6.1.2. Justificació de la elecció

Com es pot observar en les descripcions de les taules, hi ha taules que emmagatzemen dades que perfectament es podrien incorporar dins de altres taules seguint un patró semblant al següent:

bar_ID	Nom	Geoposicio				
b722	Bar-Eto	2455,55/789,36	menu_ID	user_ID	Preu	foto_ref
			m435	1023	12€	File...
			m457	2241	9€	File...
			....	....	....	...

D'aquesta manera es molt més natural el registre de dades que tenen relació entre si sense tenir que duplicar-les. En conseqüència, les 6 taules que hem vist al principi d'aquest apartat es transformarien en una sola base de dades amb registres dinàmics on cadascun tindria la seva estructura particular i les seves subestructures.

Es per això que he decidit utilitzar una base de dades no relacional i em decanto per NOSQL.

Dintre de les bases de dades NOSQL n'hi ha moltes que complirien la funció i, per ser absolutament realistes, CrouchDB potser es la base de dades no relacional més estesa que hi ha, però com que Ruby on Rails no té implementació directa de CrouchDB però si que en té per MongoDB, he decidit utilitzar aquesta última per implementar-la a GangMenu.



Les característiques principals de MongoDB son:

- Base de dades documental
- No te esquema prefixat
- Escalable horitzontalment de forma fàcil
- Emmagatzemament immediat en memòria
- Molt més ràpida que les altres BBDD

- Es poden omplir les col·leccions (taules) amb documents de diferents formats
- Minimitza o gairebé elimina la redundància de dades
- Ruby te una gemma que connecta directament amb MongoDB
- Te un molt molt divertit.

### 3.6.2. Sistema d'allotjament

Un bon disseny de sistemes de servidors ha de tenir en compte els següents aspectes:

- Tenir clar on hi haurà la informació de totes les aplicacions
- Fer la relació de totes les aplicacions que caldrà instal·lar
- Veure quins permisos hauran de tenir totes aquestes aplicacions.
- Esbrinar les necessitats reals de disc (particions d'usuaris, de correu electrònic...)
- Necessitat d'informació
- Numero potencial d'usuaris
- Nivell de concurrència

I en funció de tot això es prendran les següents decisions:

- Numero i capacitat de discos
- Contingut i numero de particions dels discos
- Disposició de la informació en els servidors
- Número total de servidors

De forma natural i bàsica, la distribució dels discos en el servidor contempla els següents punts:

- Partició de sistema
- Partició d'usuaris
- Partició d'aplicacions
- Partició de dades

Per tant comencem a especificar cadascuna de les dades anteriors.

#### Quines aplicacions caldria instal·lar?

- Sistema operatiu Linux
- Servei web Apache
- Servei llenguatge PHP
- Ruby on Rails
- Servei Database MongoDB
- Passenger Apache\_RoR
- Postfix mail server
- Afbakup
- OpenSSH
- Servei ftp vsftp
- Samba file sharing
- iptables

**On hi haurà la informació de totes les aplicacions?** En principi hi han aplicacions que poden conviure en un mateix terminal, per tant podríem fer els següents grups:

- Sistema 1:
  - Linux Debian 6.04
  - Apache
  - PHP
  - Ruby on Rails
  - Passenger Apache\_RoR

- Postfix mail server
- OpenSSH
- Vsftp
- Samba
- AfBackup client

- Sistema 2:
  - Linux Debian 6.04
  - MongoDB
  - AfBackup Client
- Sistema 3:
  - AfBackup Server

**Quins permisos han de tenir aquestes aplicacions?**

• Sistema operatiu Linux	: Tots el permisos
• Servei web Apache	: Lectura i escriptura de la carpeta dedicada al servidor web
• Servei llenguatge PHP	:Lectura i escriptura dedicat
• Ruby on Rails	:Lectura i escriptura dedicat
• Servei Database MongoDB	:Lectura, escriptura i execució dins el servidor de base de dades
• Passenger Apache_RoR	:Lectura i escriptura en carpeta apache, web i desenvolupament RoR
• Postfix mail server	:Lectura i escriptura en carpeta dedicada
• Afbakup	:Lectura i escriptura en el servidor de backup
• OpenSSH	:Lectura, escriptura i execució en funció de l'usuari
• Servei ftp vsftp	:Lectura i escriptura en funció de l'usuari
• Samba file sharing	:Lectura i escriptura en funció de l'usuari

**Esbrinar les necessitats reals de disc:** En principi, necessitaríem:

- Partició per allotjar la pàgina web i la API
- Partició per accés ftp
- Partició per PostFix mail
- Partició o disc dedicat per la base de dades
- Disc dedicat per el sistema de backup

**Esbrinar el numero potencial d'usuaris:** Donat que es un projecte molt nou, només podem fer aproximacions del potencial de tràfic que podem esperar. Però els grans números serien:

- Usuaris potencials de la web a un any vista: des de 30 diaris al principi a 500 diaris al cap de 365 dies
- Usuaris potencials del mail: màxim 6 contes de correu en el primer any
- Usuaris potencials ftp: màxim 6 usuaris en el primer any
- Usuaris potencials de la base de dades: màxim 2 usuaris els primers 3 anys
- Usuaris potencials del sistema de gestió dels servidors: màxim 1 usuari el primer any
- Usuaris potencials SSH: màxim 6 usuaris el primer any.

**Per últim calcular el nivell de concurrència:** Tal i com hem estimat en el punt anterior, calculo que com a màxim aconseguirem 500 usuaris diaris en el

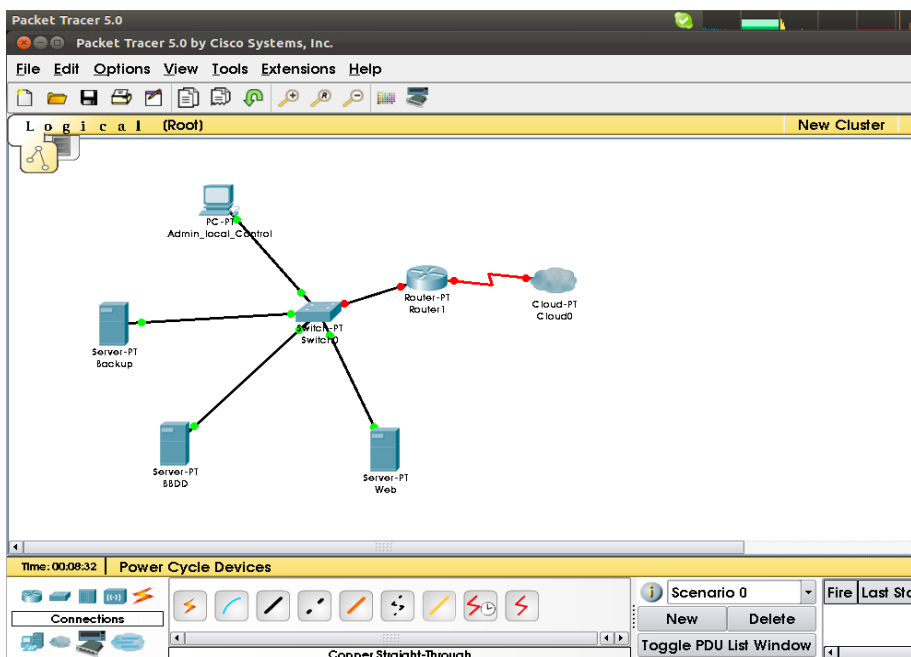
primer any, i en el segon, l'objectiu seria aconseguir triplicar aquesta xifra.

### 3.6.2.1. Sistema inicial

Per tant, després d'haver vist tot el que cal, podem dir que el sistema d'allotjament de GangMenu ha d'estar dividit en diferents subsistemes tal i com es mostra a la captura següent:

Segons les especificacions que he estat exposant fins al moment, el que necessitem de hardware per desenvolupar GangMenu es:

- Un servidor Web que contingui:
  - l'aplicació web
  - la API
  - Servei de mail
  - Servei ftp
  - Servei SSH
  - Servei Samba
  - Servei iptables
  - Tres particions de disc independents,
- Un servidor de bases de dades que emmagatzemi les dades
- Un sistema de backup pel servidor de bases de dades i pel mail del servidor web
- Un sistema de control local per accedir a cada terminal amb:
  - SO Linux Ubuntu 12.04
  - KVM
- Un switch
- Un router



### 3.6.2.2 Sistema Objectiu (virtualització)

Amb les noves dimensions i complexitats dels sistemes informàtics actuals, un dels problemes més importants que tenien els enginyers a la hora de provar noves tecnologies era que, per instal·lar nous sistemes operatius o aplicacions de propòsit específic, calia dedicar algunes màquines físiques a les proves d'aquests elements.

A la llarga això es va tornar terriblement ineficient pel cost, en temps i diners, que implicava tenir que muntar màquines específiques, buscar drivers adequats, configurar targetes, muntar discs durs específics i demés coses, per tal de

testejar aquestes novetats.

Ja fa més de 30 anys que IBM va crear un sistema per facilitar tota aquesta feina i, sense esperar-ho (o si) van crear un nou concepte en la seguretat dels sistemes i les dades i en la redundància de plataformes i servidors... va néixer la virtualització.

Segon la definició de la pròpia IBM ([www.ibm.com](http://www.ibm.com)), una màquina virtual es un programa que es capaç d'executar sistemes operatius de la mateixa manera que ho farien màquines físiques.

Aquest programa està instal·lat en el mateix sistema operatiu, anomenat host, i proveeix de certes interfícies virtuals d'entrada/sortida al sistema operatiu guest (convidat) per tal de poder funcionar.

A la pràctica, es tracta d'un emulador que, en arrencar el CD d'instal·lació del nou sistema operatiu, aporta totes les capacitats bàsiques perquè aquest sistema es pugui instal·lar correctament.

Això té com a principal característica que els recursos que el nou sistema operatiu "veu", no són els que realment tingui la màquina física, sinó que són el que s'han definit abans de la posada en marxa del mateix.

Aquesta característica permet independitzar les dues instal·lacions i assegurar que els processos que s'executin en el sistema guest, no s'escapin al control del host.

Per fer-ho encara més fàcil, diríem que una màquina virtual, es un ordinador dins d'un ordinador, i que les funcionalitats d'aquest ordinador intern estarien supervisades per un altre sistema que en limitaria les càrregues.

En particular, en una màquina virtual es poden definir totes les capacitats del ordinador que volguem implementar. Per exemple podem definir la quantitat de processadors que hi volem posar, la memòria caché, la memòria RAM, els accessos a discs externs, a les memòries usb, als CD/DVD i, fins i tot, podem definir quines són les carpetes del sistema host que el guest pot accedir de forma transparent.

Però aquesta assignació de recursos té la limitació de que, en tots els casos menys l'assignació de CPU, els recursos assignats no poden superar els físicament existents a la màquina host.

D'aquesta manera, si per exemple en el host tenim 2Gb de memòria RAM, no podem assignar a la màquina virtual més que una part, en concret  $M_{\text{guest}} = M_{\text{total}} - (M_{\text{usada\_host}}) * 1,2$ .

Un dels usos més estesos d'aquesta mena de tecnologia es la replicació de sistemes operatius, per part dels enginyers o tècnics, per provar-los. Així, en una mateixa màquina poden conviure diferents sistemes operatius sense patir a que es desconfiguri el sistema host.

Un altre dels usos que actualment s'està fent molt es amb la implantació de servidors virtuals de correu. En aquest cas, hi ha una màquina física amb molts recursos, que es "divideixen" de forma virtual amb N màquines diferents.

Cada màquina virtual es un servidor amb el seu sistema operatiu, la seva assignació de memòria, la seva assignació d'espai de disc, i la seva configuració, de manera que, en una "chatarra" física, podríem fer conviure varies màquines virtuals en que la suma dels seus recursos complissin la fórmula anterior.

Un altre de les màquines virtuals més utilitzades es la *java virtual machine JVM*. Aquest es un dels grans avantatges de JAVA com a llenguatge multiplataforma. En aquest cas, per executar qualsevol programa desenvolupat en JAVA, cal instal·lar en el sistema operatiu una petita màquina virtual que permet

l'execució del programa.

Es una interfície que està entre el codi desenvolupat i la màquina física pròpiament dit, i és la que s'encarrega de "traduir" al sistema operatiu les ordres concretes perquè el programa pugui funcionar.

En realitat JAVA, al compilar-se, crea bytecode binari que podria ser executable perfectament a nivell d'assembler, això és la gran gràcia del llenguatge, però és ineficient per funcionar sobre sistemes operatius, per lo que cal una interfície que pugui traduir el bytecode a llenguatge natiu i aquesta es JVM.

Com podem intuir per les explicacions, podríem agrupar les diferents màquines virtuals en dos grans grups, les anomenades *system virtual machines SVM* i les *process virtual machines PVM*

### 1. System Virtual Machines

Les màquines virtuals de sistema, permeten instal·lar sistemes operatius diferents directament sobre una interfície, no sistema operatiu, que aïlla cada instal·lació de l'altre, mantenint una capa on el hardware presentat no és exactament el de la màquina.

En aquest cas estaria inclòs, JAVA i la seva màquina de virtualització.

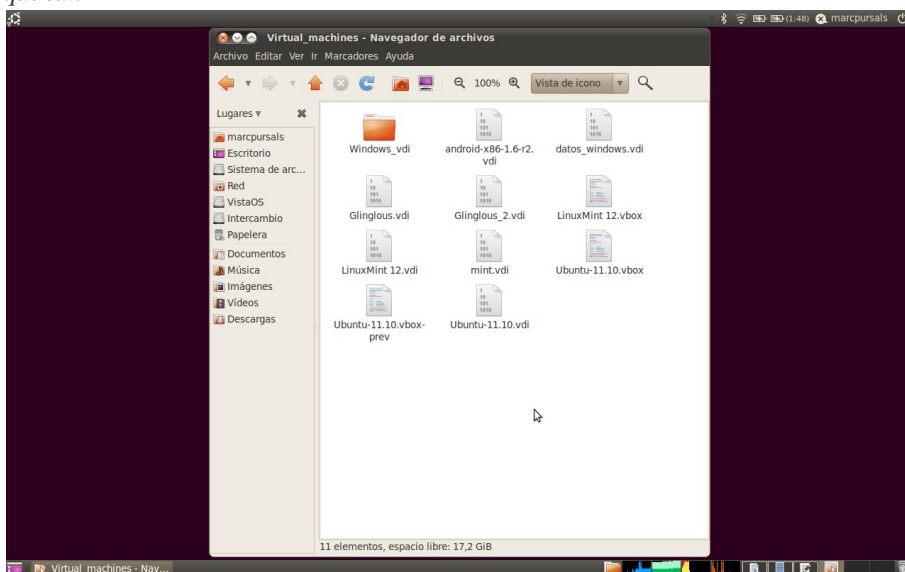
### 2. Process Virtual Machines

Les màquines virtuals de procés, en una aplicació on les màquines virtuals instal·lades ho estan sobre un sistema operatiu. D'aquesta manera es pot aconseguir que els sistemes operatius "vegin", com hem explicat anteriorment, un maquinari virtual que unifica l'execució i la controla en recursos.

*En aquest cas estaria inclosos la majoria dels productes de software de virtualització com Virtualbox (el que presentarem aquí) o Vmware, entre altres.*

Les màquines virtuals, com hem explicat, generen un arxiu on es conté tota la informació específica d'aquella instal·lació. A aquest arxiu se l'hi anomena disc dur virtual, i en el cas de virtualbox té la extensió .vdi (virtual disc interface).

*Cada vegada que es genera una nova màquina virtual, es crea, o es recupera, un d'aquest discos, de manera que en ell s'instal·la el sistema operatiu i tota la configuració que cal.*



Amb lo que, en un sol arxiu tenim tota la feina feta. I la podem recuperar de forma directa.



Per tant, les avantatges de la virtualització respecte a les màquines físiques son clares:

1. Menys procés de configuració
2. Menys cost ja que no s'ha d'invertir en maquinaria nova.

### 3.6.3. Escalabilitat

El procés d'escalabilitat és també una de les parts importants del projecte, ja que d'ella s'extrau les necessitats reals de hardware que caldrà acomplir en un futur.

Per fer els càlculs s'han tingut en compte les variables de projecció del negoci que es poden extreure de les corbes de Christian Kvalheim \*, en les que s'explicita que , per una startup com la que volem implementar el us diari a 30 dies son unes 30 persones, mentres que a 365, amb la apropiada gestió comercial i de implantació podria arribar a 500 persones diàries. D'aquesta manera podem extrapolar la nostra corba personal.

**\*Christian Kvalheim**  
 Christian Kvalheim es ex director de la oficina espanyola de XING i actualment es part del core de desenvolupament de MongoDB. A principis del 2012 va presentar unes corbes de desenvolupament teòric sobre la projecció de les startups en funció de certs paràmetres. Aquestes corbes donen una estimació de usabilitat a 365 i 720 dies amb lo que s poden fer aproximació de ús.

La corba seria exponencial amb els condicionants ja explicats. A més tindria un factor de correcció entre 365 i 720 que ara només podem aproximar però que s'ha de anar monitoritzant a mesura que passi el temps.

Corba inicial:

$$f(t) = x \cdot e^{yt}$$

Condicionants:

$$f(t=30) = x \cdot e^{y \cdot 30} = 30 \text{ visites / dia}$$

$$f(t=365) = x \cdot e^{y \cdot 365} = 500 \text{ visites / dia}$$

Resolem la equació de dues incògnites:

$$y = \frac{\ln(500) - \ln(30)}{335} = 8,4 \cdot 10^{-3}$$

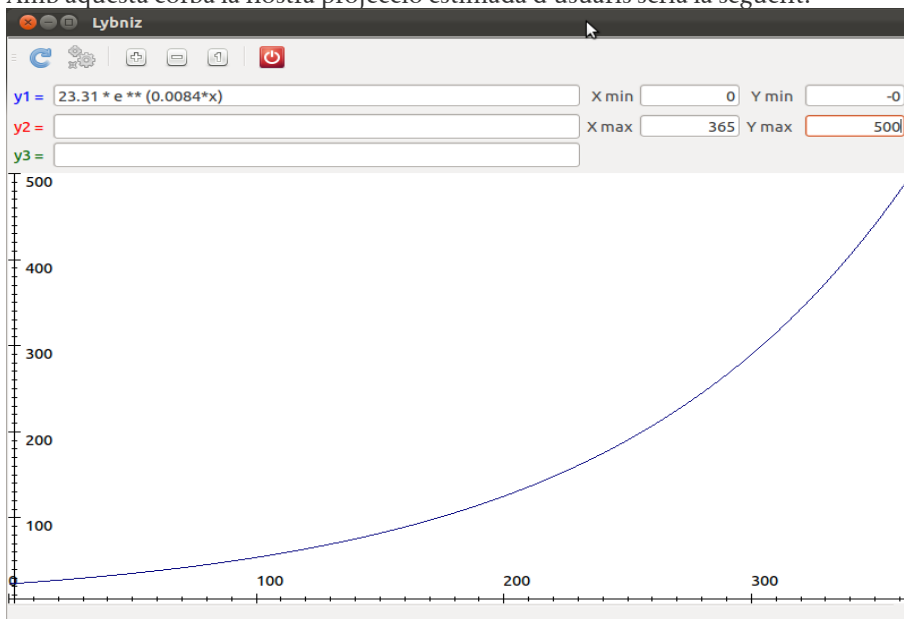
$$\ln x = \ln(30) - 8,4 \cdot 10^{-3} \cdot 30 = 3,1492$$

$$x = 23,31$$

Per tant la nostra corba de projecció d'usuaris seria:

$$f(t) = 23,31 \cdot e^{8,4 \cdot 10^{-3} t} \quad (\text{formula 1.1})$$

Amb aquesta corba la nostra projecció estimada d'usuaris seria la següent:



Una vegada sabem aquesta dada, el que farem serà condicionar l'ús de la aplicació. D'aquesta manera es previsible que el 20% dels usuaris siguin els que realment aportin continguts a la nostra aplicació. També sabem que el gruix de dades que s'emmagatzemaran seran fotografies.

Actualment, els terminals mòbils fan fotografies de pes que està al voltant dels megabytes, que amb un software adequat podem reduir a l'ordre dels kilobytes. En el cas de gangmenu utilitzo un compressor que ve amb el framework de rails que es diu Paperclip que en certs casos pot reduir el pes de les imatges de 1,5MB a 10kB. Però com que no es pot assegurar que totes les imatges es comprimeixin de la mateixa manera, he fet unes proves i amb 40 fotografies i puc concloure que, guardant-me una mica les esquenes, les imatges es comprimeixen de 1,5MB a 80kB, per tant utilitzaré aquesta xifra per calcular la dimensionalitat dels disc durs al llarg del temps. En aquesta aproximació de 80kB ja hi compto també la informació en text generada en cada iteració.

Per començar partirem de la fórmula que hem trobat abans (1.1) per calcular les necessitats d'emmagatzemament en funció del temps, sabent que només el 20% puguen fortografies i que cada fotografia pesa al voltant del 80kB.

$$c(t) = 0,2 \cdot 23,31 \cdot e^{8,4 \cdot 10^{-3} t} \cdot 80 \text{ kbytes} = 372 \cdot e^{8,4 \cdot 10^{-3} t} \text{ kbytes}$$

O el que es el mateix:

$$c(t) = 372 \cdot 10^{-5} \cdot e^{8,4 \cdot 10^{-3} t} \text{ Gbytes} \quad \text{diaris en funció del temps.}$$

En el cas que ens interessés emmagatzemar la informació continguda durant dos anys abans de esborrar-la la capacitat que ens caldria assegurar durant aquests dos anys seria:

$$C_T(t) = \int_0^{720} 372 \cdot 10^{-5} \cdot e^{8,4 \cdot 10^{-3} t} \text{ Gbytes} = 187 \text{ Gbytes}$$

Per tant, ja tenim la projecció de capacitat necessària per acomplir les exigències de l'aplicació i, en conseqüència, de hardware.

## 4. Implementació i demostració

### 4.1. Modificacions i demostració sobre l'aplicació mòvil

La veritat es que l'aplicació mòbil ha tingut moltes diferències respecte a lo que s'havia plantejat inicialment. La majoria d'aquestes modificacions han estat per culpa de una deficiència que té Phonegap com a framework de programació.

Phonegap, en realitat, es un framework que crea pàgines web amb HTML5, CSS3 i JavaScript, això ja ho havíem comentat. El tema es que cada aplicació en phonegap funciona sobre un micro servidor web (un micro Apache) que es parla amb el terminal i amb el nostre domini. I aquí està el tema.

Com es conegut, hi ha una restricció de política en la xarxa que especifica que no es poden fer peticions GET i POST amb confirmació entre diferents dominis, i com que la nostra aplicació, de totes totes, es un domini independent, no pot fer requests directes al domini web per enviar variables o per demanar-les. I tot i que sembli impossible, perquè està en la base del sistema, resulta que no pots interactuar directament amb JSON.

Aquesta problemàtica la he solventat amb la utilització de JSONP, que es una variació de JSON que et permet saltar-te la restricció del cross-domain. Però té un preu a pagar. JSON només permet fer peticions tipus GET, de manera que s'ha de configurar el servidor per:

- Acceptar l'enviament de dades per GET enlloc de POST
- Configurar el servidor per enviar dades en JSONP

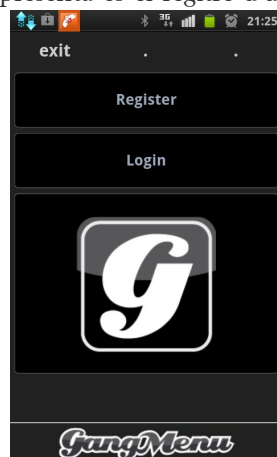
Per sort, Ruby on Rails, permet fer això amb una gemma que intercanvia el format de JSON a JSONP, de manera que quan l'hi arriba una request GET amb callback, ell entén que es tracta de JSON i respon la petició amb un JSONP.

Un altre problema que m'he trobat es que, en el cas de enviar una fotografia, cosa que ens interessa, al no poder enviar POST, es té que fer un truco que es, enviar les dades per JSONP i ens arriba una conformació de petició amb un identificador, i enviar la fotografia per transferència de fitxer sobre un socket virtual que s'ha de programar en el servidor. Per sort això està documentat en la API de Phonegap, on la versió amb la que he treballat ha estat Cordova-1.5.0.

A part d'aquesta part més tècnica hi ha hagut molta diferència en el disseny de la aplicació. Al final no he pogut comptar amb amics que m'ajudessin en el disseny i els css, però si que he aconseguit que un bon amic em faci el logo i la tipografia del nom, amb lo que he pogut fer quelcom decent, però molt lluny del que m'agradaria.

També funcionalment hi ha hagut diferències doncs, per exemple, el registre dels usuaris i el login han d'estar a la primera pàgina, tal i com es mostra a continuació:

La primera cosa que es presenta es el registre d'usuaris o login per entrar a l'aplicació:



En el cas de que no haguéssim tingut encés el GPS o el wifi, ens hagués sortit un diàleg en el que ens demanaria que encenguéssim alguna de les dues opcions per poder-se geoposicionar.

Si agafem la opció de registrar-se ens demanará:

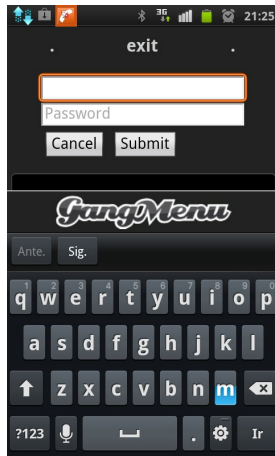
- Email
- Username
- Password
- Password\_confirmation

Una vegada registrat l'aplicació redirecciona directament a la pàgina principal.

En el cas de que ja tinguem fet el registre, podem accedir a la aplicació a través de Login, on ens demanará:

- Username
- Password

En ambdós casos, les peticions s'envien al servidor que valida el registre o el login en funció de les dades que rep per JSONP.



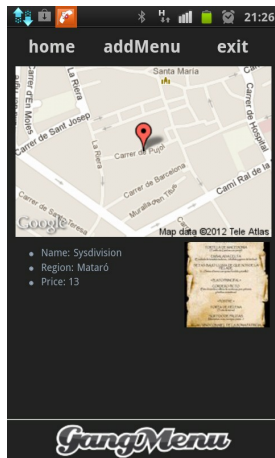
Una vegada hem arribat a la pàgina principal, l'aplicació fa una request JSONP al servidor perquè l'hi enviï les dades de tots els llocs que tenim al voltant. Per fer-ho, l'aplicació ens geoposiciona i envia la latitud i la longitud al servidor i ell ens retorna totes les posicions que tenim al voltant.



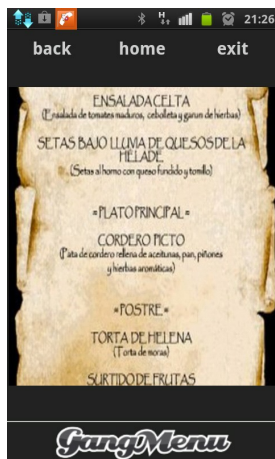
Una vegada tenim la llista complerta podem seleccionar algun dels items que ens mostra i l'aplicació ens porta a una pàgina amb tota la informació disponible

del place, que es:

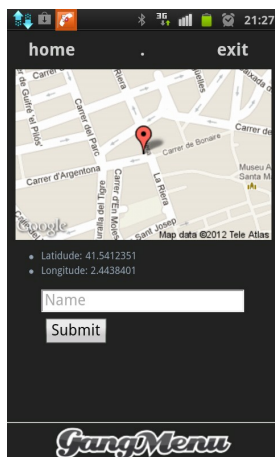
- Geoposicionament del bar en un mapa
- El nom del Bar
- El preu del últim menú que s'ha pujat
- La fotografia, en petit, del menú.



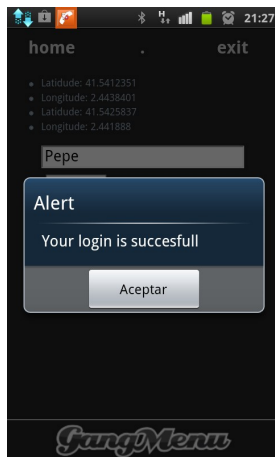
Si volguéssim veure el detall del menú, el que tenim que fer es clicar a sobre d'ell perquè ens ensenyi la fotografia més gran.



Tornant una mica enrere, si no tinguéssim places al voltant, podem crear-ne de nous amb el botó que apareix a dalt a la esquerra en el menú de la home: 'addBar'. En aquest cas, el telèfon ens geoposiciona i ens mostra el lloc exacte on estem en un mapa, i ens permet afegir el nom del bar.



Una vegada hem enviat el formulari al servidor, aquest ens retorna un ok en el que ens diu que la request ha estat correcte.

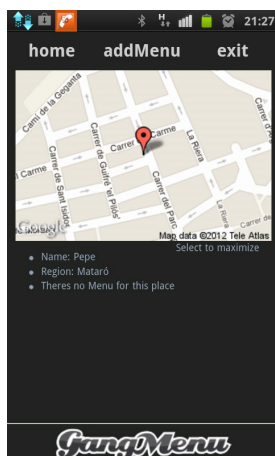


L'aplicació ens retornarà a la home, on veurem que el lloc que acabem de crear està en la primera posició, com era d'esperar, doncs es el lloc amb la distància més curta respecte a nosaltres.

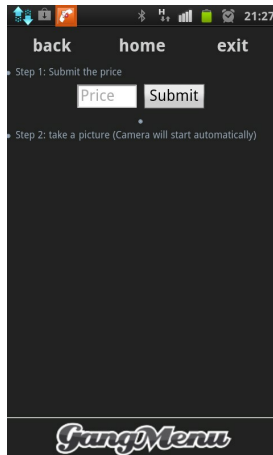


En una futura versió, en aquesta pantalla hi haurà per cada item, el preu de l'últim menú en el que s'ha penjat, i la data en la que s'ha fet.

Si entrem dins d'aquest place que acabem de crear veurem que hi ha totes les dades del lloc però que no hi ha menús postejats, i així ens ho fa saber l'aplicació amb la última línia de les dades.



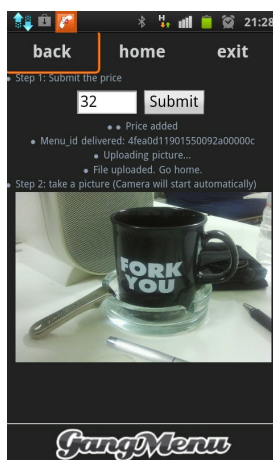
En aquesta situació ens pots interessar afegir un menú al Bar. Per fer-ho, a la part superior de la pantalla hi ha un botó 'addMenu'. Si el clickem ens porta a la següent pàgina.



Es aquesta pàgina ens demana que introduïm primer el preu del menú que volem penjar. Una vegada fet això, l'aplicació obre automàticament la càmera del telèfon i ens permet fer la fotografia. Una vegada feta la fotografia s'ens demana si volem guardar-la.



En guardar-la, l'aplicació de forma automàtica, ens retorna a la pàgina on hem partit per fer la fotografia i ens mostra el estat actual de enviament de la mateixa cap al servidor.



Quan ja està carregada la fotografia, ens permet tornar al menú principal o home, o tornar enrere o sortir. On finalment ens retornarà a la següent pàgina:





## 4.2. Modificacions i demostració sobre l'aplicació web

Després de les modificacions en mòbil, també he tingut que fer e mateix procés en l'aplicació web, doncs a mesura que anava programant algunes coses s'anaven simplificant i unes altres complicant. Però en general, aquesta part també s'ha simplificat bastant.

Per començar, i de la mateixa manera que en l'anterior apartat, hi ha hagut una sèrie de canvis que son producte del disseny. Si bé al inici de l'aplicació volia tenir una estètica concreta, al final he decidit fer un producte resultón però que no impliqués tancar-me dies per millorar l'estètica. Així es pot observar que les línies són molt més simples que en la proposta inicial.



Per començar he cregut que no tenia gaire sentit atabalar massa a la gent amb molta informació que no l'hi sigui de interès i, per tant, he cregut que la primera pantalla de la web tindria que ser molt simple, només preguntar ón ets.

El tema aquí es que, això només està accessible si no funciona en el teu navegador la opció de geolocalització. M'explico. Quan s'arrenca la primera pàgina hi ha una request d'html5 que automàticament et geoposiciona i localitza, amb 150m d'error, ón estas, amb lo que la primera pàgina, segons el navegador, no dona temps casi ni a veure-la.

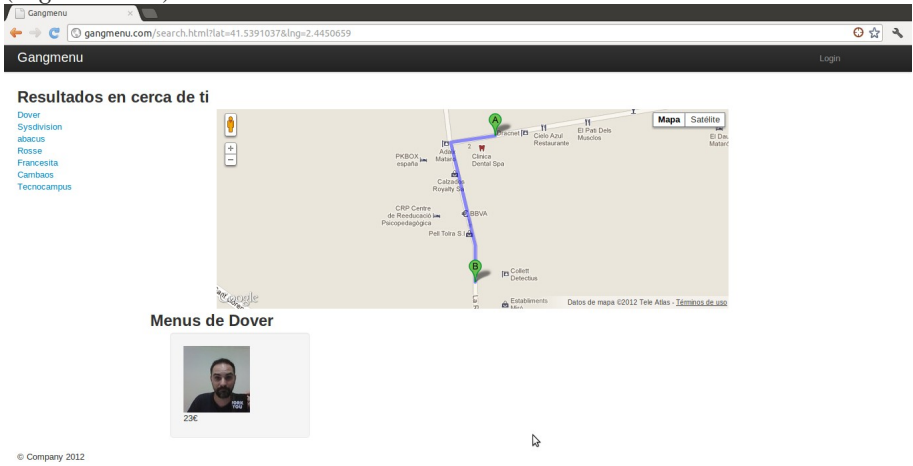
Què es el que passa? Doncs que al geoposiciona-nos llancem una request a google perquè ens ensenyi un mapa on hi ha: (veure screenshots de la següent pàgina)

- per un costat la nostra posició (punt A)
- la posició del bar més pròxim (punt B)
- per altra banda un llistat (a l'esquerra) dels llocs més propers ordenats per distància
- Per ultim el llistat dels menús que s'han pensat en aquest Bar

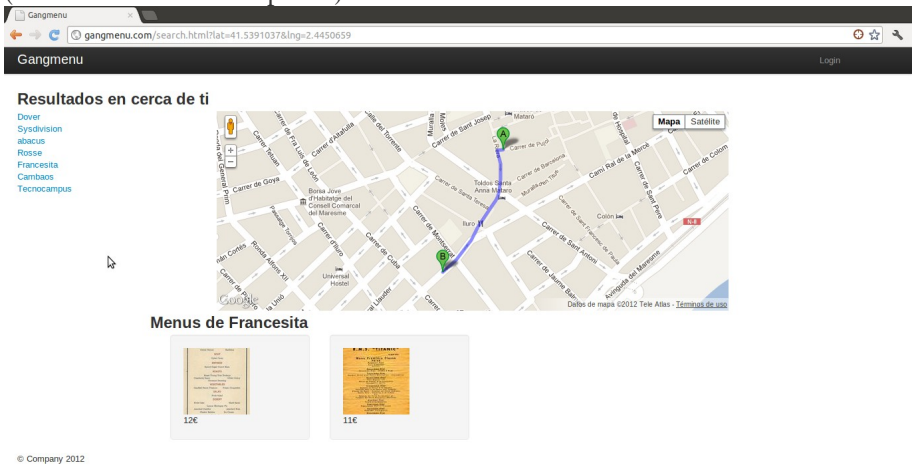
Si anem iterant sobre els bars que tenim més a prop, podem veure que la posició del mateix en el mapa canvia. Això es així perquè he aprofitat la funcionalitat de upload dinàmica de la api de Google Maps. (veure screenshots de la següent pàgina)

Com ja he comentat, la aplicació s'ha simplificat respecte a la dissenyada perquè, en realitat no crec que faci falta que la gent vegi el llistat de places que es van crear.

(Pàgina d'inici)



(Iteració entre diferents places)



Per altra es pot observar que, adalt a la dreta hi ha un botó de login. Amb aquest botó es pot accedir al back end de la aplicació on podem gestionar la aplicació en cas de que siguem usuaris o administradors.

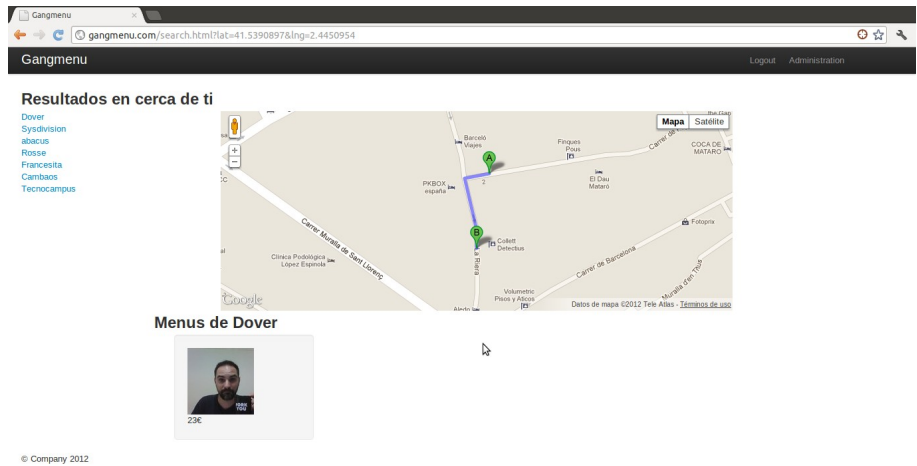


Comes pot observar hi ha les funcionalitats bàsiques de qualsevol pàgina web: Registre (sign up) o Sing In. En Sign in es pot accedir en qualsevol del dos rols que estan permesos: Usuari o administrador. Les funcionalitats de ambdós son diferents, ja que els usuaris (de moment) encara no poden crear places, però es una cosa que començaré a desenvolupar en breu.

No està implementat perquè no tenia clar com es resolrien els temes de supervisió de pujada de continguts des de la web. Això és important perquè no voldria que els amos dels bars comencessin amb una guerra de pujar fotografies desprestigiant als altres. En quant tingui la política clara en aquest aspecte ho implementaré.

En el cas dels administradors té totes les funcionalitats per poder:

- crear places (bars)
- crear i pujar menús
- editar i eliminar places
- crear usuaris
- editar i eliminar usuaris

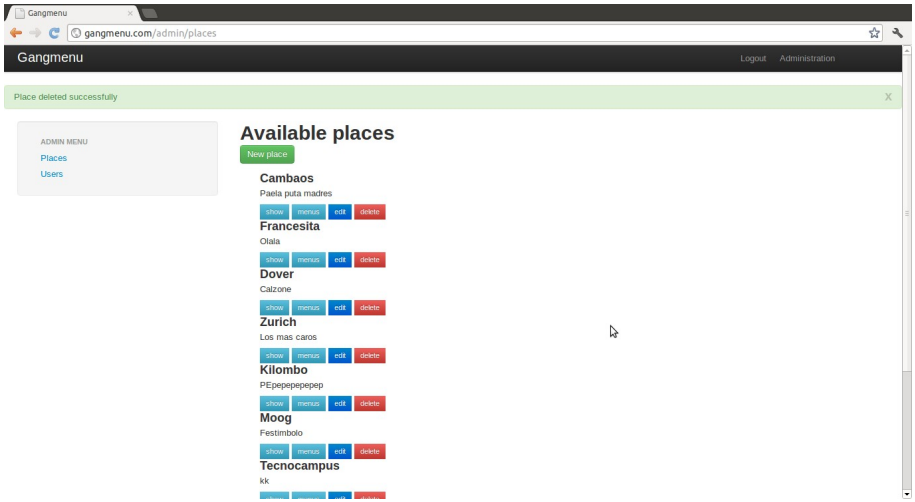


La pàgina dels administradors és una mica diferent a la dels usuaris, doncs com es pot comprovar, hi ha un botó al costat de logout que posa 'Administration' i que ens dona accés a les funcionalitats que hem comentat abans.



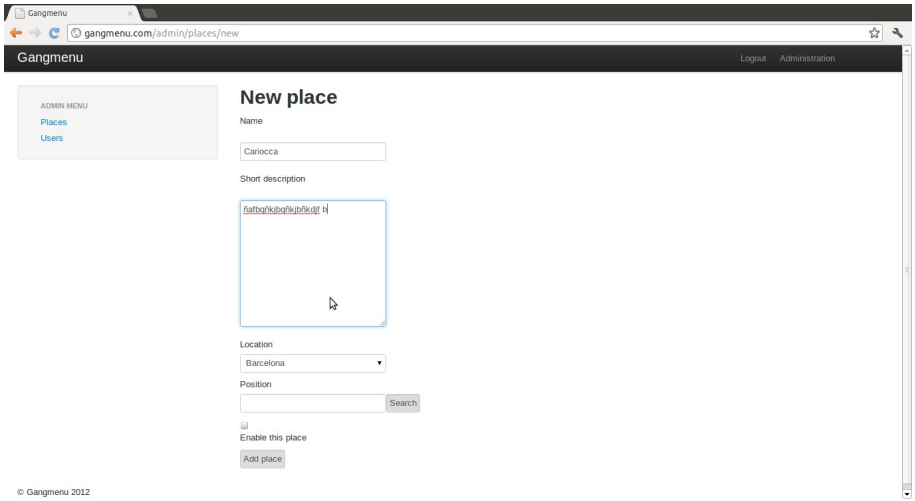
Una vegada dins de Admin, ens trobem amb dos opcions per gestionar: Places (bars) o Users (Usuaris), que tal i com hem comentat anteriorment, ens permetran administrar aquestes funcions.

En el cas de Admin Places, ens trobem amb una llista dels llocs que estan creats per ordre d'antiguitat, o sigui, els primers que es veuen són els que es van crear primer. L'objectiu és que aquesta visió estigui seccionada per localitzacions generals com ciutats o pobles. Però al igual que en el tema de política d'usuaris, com que encara no tinc clar com gestionaré aquest punt no ho he implementat. En realitat podria fer-ho per localitats físiques o per km a la rodona... he de veure quina és l'ús real per decidir respecte als bars deslocalitzats.

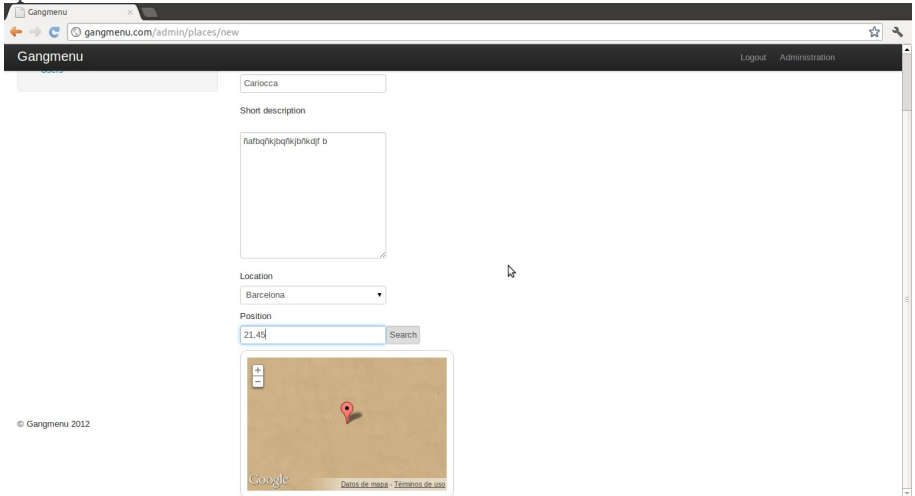


En el cas que es vulgui crear un new place, el funcionament serà el següent:

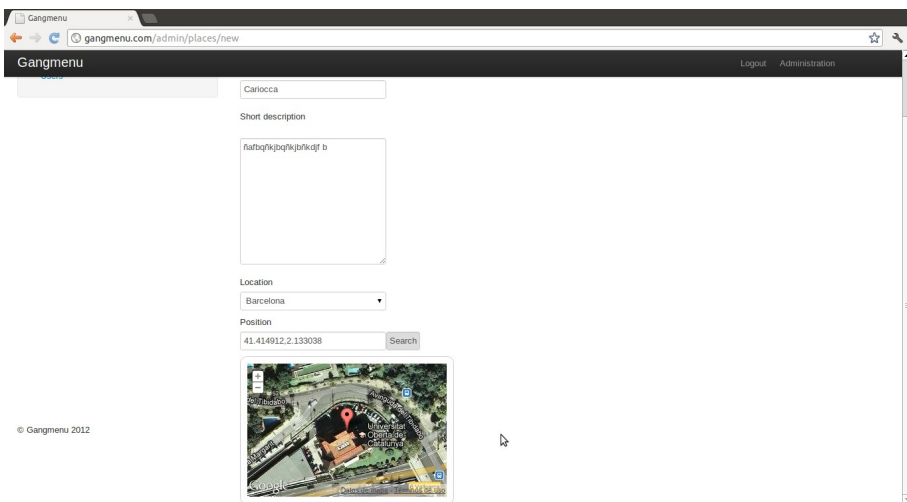
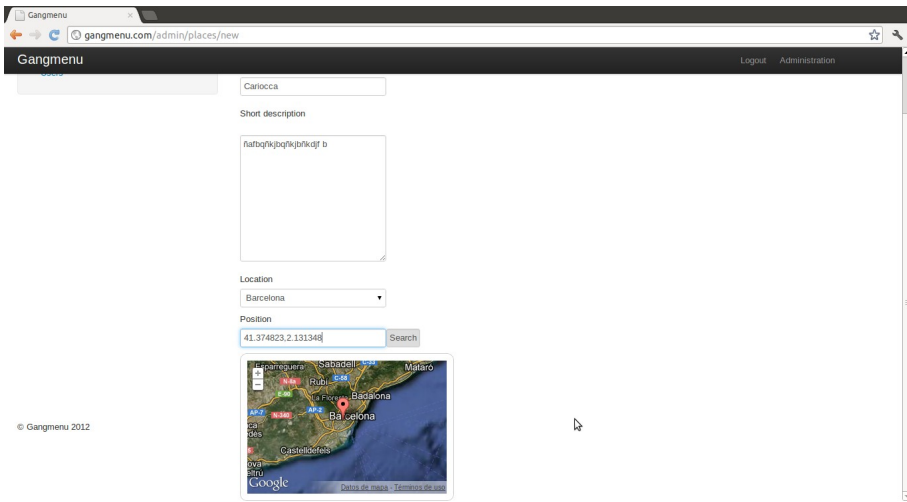
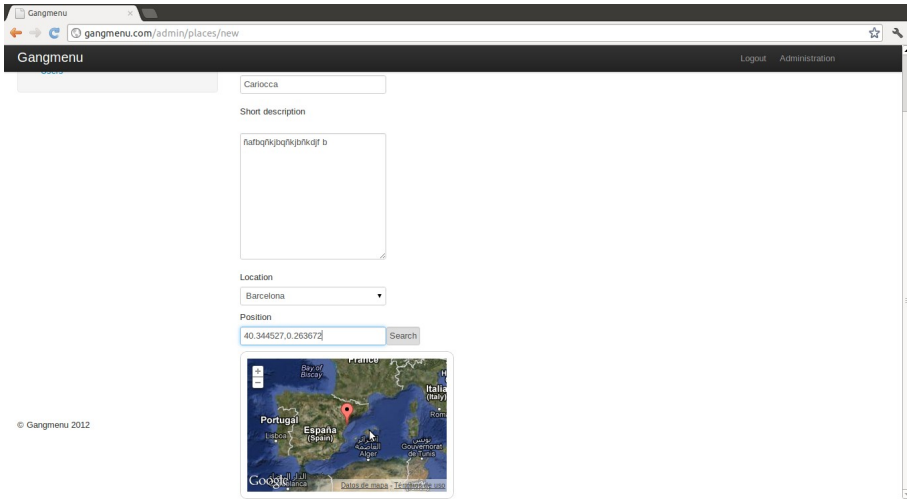
1. Omplir les dades:



2. Seleccionar la posició exacta des de google-maps posant una posició com 21,45 i apretant el botó search:



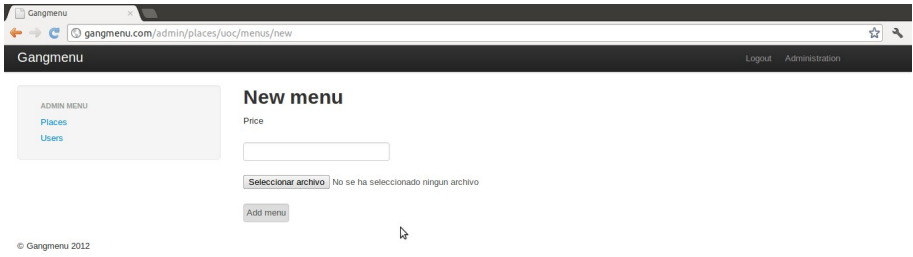
Ara fem scroll out sobre el mapa i recolquem el pointer en la posició que ens interessa:



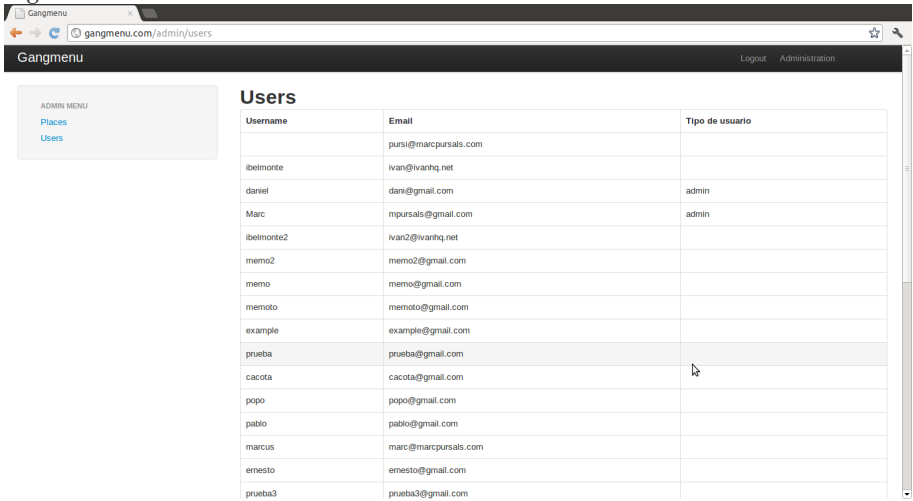
4. Ara que em seleccionem la UOC clickem a un altre posició de la pantalla i afegim el place que acabem de crear.

Una cosa important que s'ha de tenir en compte es que hi ha una relació directa amb l'aplicació mòbil, de manera que qualsevol canvi en el llista de places té afectació directa amb la corresponent llista de phonegap.

Ara, si volguéssim afegir un menú al place UOC que acabem de crear, el que tindriem que fer es seleccionar el botó menus. Una vegada a dins podem pujar una fotografia del menú i també posar el preu.



Per el cas de usuaris, en realitat la gent es dona d'alta en Sign Up, i d'allà passen a una base de dades que es pot consultar en l'apartat users de Administration. Els Administradors no té sentit que creïn usuaris a no ser que siguin altres administradors, però si que poden eliminar-los. El que trobem es el següent:



Per qualsevol revisió online: [www.gangmenu.com](http://www.gangmenu.com)

## 5. Conclusions i línies de futur

Phonogap ha demostrat ser un framework molt adequat per fer aplicacions multiplataforma que estiguin encarades a una extrapolació directe del paradigma web a terminals mòbils, i la majoria de les aplicacions que es faran es basaran en el mateix concepte, per tant es adequat. Però es queda una miqueta curt a l'hora de implementar capacitats dels propis terminals, i és impossible de executar elements nadius com botoneres i cert l'aspecte de la distribució.

Una de les coses que no m'ha agradat gens ha estat la impossibilitat de poder compilar el codi per Iphone. Això ha estat per culpa de que, com he comentat de passada, per desenvolupar aplicacions per iOS necessites X-Over que només funciona sobre OSX, que a la vegada només funciona sobre maquinari original Apple.

Això significa que tots els desenvolupadors que desenvolupem aplicacions multiplataforma sota Linux, com es el meu cas, estem condemnats a no fer-ho per Iphone o a comprar un portàtil cutre amb OSX per poder instal·lar l'SDK només per compilar el que hem desenvolupat amb Linux. Això es un despropòsit que entenc que s'ha de solventar en les pròximes versions de Phonogap o amb alguna altra eina que permeti fer-ho.

Respecte a Ruby on Rails no hi ha paraules per descriure la facilitat amb el que s'aconsegueix desenvolupar aplicacions minimalistes però de gran eficiència, i la gran interacció que té amb "l'exterior" gracies a les seves gemmes, com la de mongoDB, que et permet una interacció amb la base de dades gairebé tant ràpida com amb MySQL.

A part, el sistema de programació basat en el paradigma de MVC (model-view controller) permet estructurar molt el desenvolupament, i agilitzar la eficàcia del codi especialitzant cada part en el que realment cal. De manera que cada part conté el mínim codi per funcionar, el que també permet una major eficiència a la hora de trobar possibles errors en la programació. A

MongoDB ha estat un gran descobriment de fa uns mesos, que permet una flexibilitat a la hora de emmagatzemar dades, que amb altres bases de dades relacionals era una utopia. El fet que cada document pugui tenir documents dins, que tenen llistes relacionades amb altres documents, permet una llibertat a la hora de dissenyar la taula que no havia trobat amb MySQL o PostgreSQL. El fet de pensar "vull posar això aquí" i poder-ho fer i que sigui tractable es impresionant.

Si bé es cert que per certs dissenys en els que les transaccions son una part fonamental del disseny, MongoDB no es una bona base de dades perquè realment no compleix amb aquesta funcionalitat.

Per altra banda, inicialment m'havia plantejat la possibilitat de fer aquest treball paral·lelament amb el desenvolupament del process mangement, ja que ambdós àmbits els he treballat bastant, però em vaig trobar que hi ha una sèrie de dificultats en la implementació de RESTFUL en phonogap i que la seva implementació no es trivial, lo que em va retrassar bastant fins que vaig trobar la solució. Per tant, vaig decidir centrar-me només en la part de desenvolupament de les aplicacions i deixar la part de process management.

També vaig estar temptat de afegir un pla de negoci pel projecte, per donar la visió completa del mateix, però el vaig deixar a mitges perquè, si bé el projecte té una gran projecció comercial, l'anàlisi pormenoritzat en un camp tant nou com el de les startup, implicava una inversió de temps que estava fora dels càlculs que inicialment m'havia plantejat.

En quant a les línies de futur del projecte podem dir varies coses:

- Lo primer que s'ha de fer es contractar a algú que faci la feina de dissenyador gràfic. El que he fet ha estat molt pobre en quant a disseny ja que, amb la gent que comptava per ajudar-me en aquesta part, no ho han pogut fer, i s'ha de reconèixer que el meu gust estètic deixa molt que desitjar.
- En segon lloc l'aplicació tindria que poder accedir a Navigator perquè et dirigeixi cap el restaurant, cosa que actualment, com ja hem explicat, no està implementada
- Ampliació de les funcionalitats:
  - Ja que tenim la possibilitat de tenir una porta de connexió directe amb els restauradors, estaria bé que es poguessin **fer reserves** a través de l'aplicació.
  - També estaria bé que es poguessin fer **suggerències directes** a les bústies dels restauradors per augmentar la relació amb el client, i la seva fidelització.
  - **Creació d'un sistema de bonificacions** per client habitual o per "pujar" més menús al sistema. Això necessitaria de desenvolupament d'un sistema de codificació i certificació de les bonificacions que tindria que ser validades pels restauradors.
  - Permetre que els restauradors facin **ofertes directes** als clients habituals, a través de comunicació directe al mail. Això només seria possible si el client potencial està d'acord en acceptar les comunicacions.
  - **Publicació de carta on-line**, per augmentar les possibilitats, estaria bé que l'aplicació pogués donar el servei de veure la carta del restaurant i les suggerències del dia o de la setmana.





**BIBLIOGRAFIA:**

• Agile web development with Rails	Autor	Sam Ruby, Dave Thomas i David Heinemeier
	Any	2011 4a edició
	Font	Pragmàtic Bookshelf
	Resum	Llibre de capçalera per programar aplicacions amb Ruby on Rails
	Aspectes rellevants pel projecte	Es la base del desenvolupament de l'aplicació
• RESTful web services	Autor	Leonardo de Seta
	Any	2008
	Font	Www.dosideas.com
	Resum	Introducció al model predominant en la elaboració de serveis web
	Aspectes rellevants pel projecte	Donar una base sòlida de la interrelació entre aplicacions web i les APIs
• Introducció a JSON	Autor	Varis
	Any	Varis
	Font	Www.json.org
	Resum	Introducció a la arquitectura i usos de un model que majoritàriament s'utilitza per intercanviar hash de dades entre aplicacions
	Aspectes rellevants pel projecte	Totes les APIs, pel que conec, entreguen JSON's a les peticions eternes.
• Google Data Protocol	Autor	Google Data Apis Team
	Any	2009
	Font	code.google.com
	Resum	Documentació general sobre les APIs que Google posa a disposició dels desenvolupadors d'aplicacions
	Aspectes rellevants pel projecte	La geolocalització es la base del projecte i tots els serveis que d'ella s'en deriven.
• Phoneyap	Autor	Andreu Lunny
	Any	2011
	Font	Packt publishing
	Resum	Llibre de capçalera per programar aplicacions amb Phoneyap
	Aspectes rellevants pel projecte	S'utilitzarà per desenvolupar la aplicació multiplataforma del mòbil.

**AGRAÏMENTS:**

*En primer lloc m'agradaria donar les gràcies a la persona que m'ha permès estudiar una carrera i que sempre m'ha ajudat en els pitjors moments. Mare només n'hi ha una, però tu ets la millor que em podia tocar. Gràcies de tot cor.*

*En segon lloc agrair a la meva parella Aroa, a la meva germana Rosa, al iaio Santi i a tota la meva família, l'ajut i suport per fer aquest somni realitat.*

*En tercer lloc m'agradaria donar les gràcies al Dani i a l'Ivan per ensenyar-me a programar ja fa uns quants anys, i mostrar-me un camí que estic gaudint molt. FORK YOU!*

*Per últim agrair a tots els amics que m'han animat a acabar quelcom que tenia penjat, i que sempre m'han encoratjat. Gràcies a tots.*

*Aquest projecte està dedicat als meus dos petits grans homes, Luca i Albert. Espero que algun dia estiguen tant orgullosos d'algú, com jo ho estic de vosaltres. Us estimo. I ho faré sempre.*

*Marc Pursals i Garcia*

