



KAPP'MON

Memòria

Lliurament Final

Ruben Fitó

ETIG

Programació d'aplicacions mòbils Android

Consultors

Joan Vicent Orenge Serisuelo

Jordi Almirall López

Professor

Robert Clarisó Viladrosa

Data Lliurament

15 de juny de 2016

Índex

Introducció.....	4
1.1 Context i justificació del Treball.....	4
1.2 Objectius del Treball.....	7
1.3 Enfocament i mètode seguit.....	9
1.4 Planificació del Treball.....	11
1.5 Breu sumari de productes obtinguts.....	14
1.6 Breu descripció dels altres capítols de la memòria.....	14
Disseny.....	15
Arquitectura.....	18
Aplicació nativa.....	18
Servidor.....	19
Arquitectura General.....	19
Diagrama Arquitectura.....	21
MVC.....	21
SOA – Arquitectura orientada a serveis.....	22
EDA - Arquitectura dirigida per events.....	23
Desenvolupament.....	24
Servidor.....	24
Aplicació.....	27
Estat del projecte.....	29
Problemàtiques.....	30
Servidor.....	30
Aplicació.....	31
Testing.....	32
Servidor.....	32
Aplicació.....	34
Proves unitàries.....	37
Registre.....	37
Nou Recordatori.....	38
Canals.....	39
Subscripcions.....	40
Alarmes.....	41
Llicències.....	42
Servidor.....	42
Aplicació.....	44
Conclusions.....	46
Noves oportunitats.....	46
Recomanacions.....	47
Kapp'Mon.....	48
Bibliografia.....	49
Documentació.....	49
Fonts d'Informació / comunitats.....	49
Guies d'usuaris / Wikis.....	49
Eines ONLINE.....	50

Llicències.....	50
-----------------	----



Introducció

1.1 Context i justificació del Treball

Hem de ser conscients que estem una societat on el nivell de vida pot ser frenètic, la feina, els fills, tasques de la llar, etc. I si no inverteixes molt de temps en organitzar-te cadascuna de les tasques el temps s'esgota i sempre deixem per demà el que no hem pogut fer avui.

És clar, es pot fer un planning de feina per a que tot quadri, però, què passa amb els imprevistos? Què passa amb els nostres hobbies? Quantes vegades ens hem oblidat que hem quedat per sopar i arribem tard? Quantes vegades ens hem perdut un partit del Barça, una bona bona pel·lícula que donen per televisió? Quantes vegades...?

KAPP'MON vol fer la vida més fàcil a les persones. **KAPP'MON** és una aplicació que té com a base els recordatoris. El primer que es podria pensar és «Això ja està inventat!!» i efectivament, si fem una ullada podem trobar desenes d'aplicacions que realitzen recordatoris.

Per exemple:

- Google Calendar. <https://www.google.com/calendar>
- Begin. <http://beginapp.co>
- Any.DO. <http://www.any.do/>
- Listastic. <http://mcleanmobile.com/Listastic/>
- I molts més...

Però hi ha una cosa que cap d'ells té, poder compartir aquests recordatoris sense necessitat de disposar d'un e-mail o telèfon de contacte.

Com hem dit KAPP'MON té com a base el recordatori, i la seva potència resideix en que el propi recordatori és un objecte que es podrà compartir amb gent coneguda o desconeguda, això no importa, el que importa és que no hi haurà limitacions. Es podrà compartir des de un xat, un fòrum, un anunci publicitari, facebook, una web, dóna igual, l'important és qualsevol tingui accés a aquest objecte.

El recordatori com a tal, pot estar compost per tanta informació com se li vulgui afegir. Un títol, un text informatiu, una imatge, un àudio un vídeo i , evidentment, una hora concreta per a que ens avisi. Amb tot això tenim una aplicació realment atractiva.

La realitat ens diu que amb les eines actuals, és una feineda crear un recordatori per cada tasca que hem de fer, sobretot si es troba limitada dins un horari. Què millor que algú la faci per tu i en treguis partit, i no només això, segurament la faran millor que un mateix.





Sembla que no pugui ser, però una gran immensitat de gent comparteix per internet qualsevol informació que sigui d'utilitat pels demés. És una manera de fer-se conèixer i sentir-te bé amb un mateix. Per tant, podem dir que el mercat demogràfic és molt gran a nivell personal. Però no ens podem oblidar de les empreses, de segur tindran ganes d'emprar aquesta plataforma per tal de donar valor afegit al seu servei. Per exemple el diari MARCA podria crear recordatoris dels events que facin i retransmetin.

Per a empreses, els hi seria realment molt interessant integrar-se amb el servidor de la aplicació per tal d'automatitzar la gestió de recordatoris. Gràcies a això el ventall de possibilitats s'incrementa exponencialment.

Tot i la informació del conjunt de paràgrafs anteriors, només podem fer-nos una idea del potencial d'aquesta eina amb unes quantes demostracions pràctiques:

Entorn	Missatge del Recordatori
Empresa de transport	<i>En 20 minuts li arribarà el paquet a casa</i>
Canals de TV	<i>En 5 minuts començarà la seva pel·lícula favorita. «Gladiator».</i>
Events esportius	<i>En 15 minuts jugarà el teu equip favorit. F.C.Barcelona vs Reial madrid</i>
Events festius	<i>Posa't guapo que les mosses t'esperen a la festa de la UOC en una hora.</i>
Sanitat	<i>Has de prendre la pastilla Melainventomina en 15 minuts.</i>
Dietes	<i>En 5 minuts has de menjar una POMA.</i>
Educació Canina	<i>En 10 minuts has de treure el gos.</i>
Laboral	<i>Més mal que espavilis que tens reunió amb el JEFE.</i>
Bons costums	<i>Hora de rentar-te les dents!!</i>
ITV	<i>Tens que passar la ITV en 10 minuts.</i>
Programes d'exercicis	<i>En mitja hora posat les piles que et toca fer 20 km de bicicleta.</i>
Personal	<i>En 30 minuts el teu fill ha d'anar a Anglès. Espavila!!!</i>



Mètodes d'estudi	<i>Ara són les 16:00 hores i has de descansar.</i>
Reserves	<i>Estimat client, li recordem que té taula les 21:00 a «El Bulli». L'estarem esperant en 10 minuts. Gràcies.</i>
Mini recordatoris prefixats	<i>Nou Correu!! Has de no donar resposta en 10 min!!</i>
Gimcanes	<i>Ja són les 13:00, en 10 minuts heu d'aconseguir un mapa amagat en els arbres del Passeig.</i>
Receptes de cuina	<i>En 10 minuts ja podràs posar el pollastre al forn.</i>
Coordinar events en videojocs cooperatius	<i>En 15 minuts hem de vendre tots els nostres accessoris per aconseguir al heroi "Ubiquatron".</i>

Podem veure que les possibilitats són infinites, però la gràcia de tot això és que només cal una única APP i un únic servidor. És feina dels usuaris, tant persones com empreses, de com volen utilitzar aquest sistema.

Finalment, i possiblement una de les funcionalitats més importants, és que els usuaris podran subscriure's a aquelles alarmes que més els interressi, des de horaris de partits de futbol fins a alarmes creades per a events amb els amics. Aquesta funcionalitat és la més interessant, ja que serà l'usuari qui decideixi quines alarmes rebrà i no patirà de publicitat innecessària. Només rebrà el que desitgi.

Un cop arribat a aquest punt, hem d'esmentar que una de les funcionalitats més rellevants serà l'alta capacitat d'integració amb KAPP'MON per tal de gestionar recordatoris de forma automàtica. Aquí recau el factor més important del model de negoci, on sense entrar en detalls es podria obtenir beneficis per cada X recordatoris descarregats. No obstant, no s'espera tenir uns beneficis a curt termini, si no que el primer que es desitja és que s'estengui l'ús de la aplicació.



1.2 Objectius del Treball

KAPP'MON pretén ser una aplicació multi-plataforma, ja sigui emprada des de un terminal Andorid, IOS o WEB. Per a ser realistes en la entrega em centraré en Adroid on empraré el Framework IONIC* per tal de crear l'APK final.

Si el temps ho permet, gràcies a IONIC, tindrè la possibilitat de compilar la aplicació per IOS.

Per començar llistaré el conjunt de funcionalitats bàsiques que hauria d'oferir la aplicació.

Funcionalitats KAPP'MON
Donar-se d'alta.
Crear un recordatori
Modificar un recordatori
Eliminar un recordatori
Subscriure's
Cancel·lar una subscripció
Crear una subscripció
Eliminar una subscripció
Fer sonar les alarmes pels recordatoris

Són funcionalitats bàsiques que garanteixen un producte complet des de la primera versió.

Per tal de complir amb aquest llistat KAPP'MON es compondrà de 3 components:

Servidor KAPP'MON

El servidor d'aplicacions serà qui contingui de manera centralitzada tota la informació dels recordatoris i subscripcions creats des dels terminals mòbils. Bàsicament oferirà una interfície REST per a que la APP realitzi peticions HTTP.

A nivell de servidor oferirà les funcionalitats esmentades anteriorment, a més a més d'un servei de login per a identificar al usuari.

Es preveu que la interfície REST pugui ser accessible des de altres entorns, ja que la integració amb altres sistemes és clau per al model de negoci.

*IONIC

<http://ionicframework.com/>

Tecnología híbrida que utilitza html5, Angular JS com a Model Vista Controlador i Cordova com a sistema multi-plataforma.





Interfície d'usuari

La interfície d'usuari no és més que la part mòbil que s'instal·la a cada dispositiu mòbil i dóna als usuaris la capacitat de gestionar els seus recordatoris, a través de menús, icones i missatges d'ajuda.

La interfície, a part de les funcionalitats bàsiques anteriorment esmentades, permetrà:

Funcionalitats Interfície
Visualitzar tots els recordatoris .
Visualitzar les subscripcions.
Visualitzar les alarmes que sonaran el dia d'avui
Descartar una alarma lligada a un recordatori.

Servei Android

Per tal de complir amb la tasca més important, que és la de fer saltar una alarma per què un recordatori així ho exigeix, és necessari la creació d'un servei android que vagi consultant cada X temps al servei d'aplicacions si n'hi ha o no. Si n'hi ha, el servei Android la guardarà en una cua segons l'hora en que ha de sonar.

Al mateix temps, el servei Android, anirà llegint la cua d'alarmes i la farà sonar si coincideix que l'hora del dispositiu mòbil coincideix amb l'hora de l'alarma.

Funcionalitats Servei Android
Cerca de recordatoris
Encuar recordatoris
Fer sonar l'alarma
Purgar alarmes antigues.



1.3 Enfocament i mètode seguit

Tal i com es descriu en el punt anterior, KAPP'MON està integrat per tres components, la interfície, el servidor, i un servei Android.

El més important de tots 3 és el servidor d'aplicacions, ja que contindrà totes les dades que es distribuïran per tots els dispositius mòbils. Per tant, el primer que s'hauria d'implementar és un servidor que garanteixi aquestes necessitats.

Servidor KAPP'MON

En aquest punt he pensat en emprar llenguatge java, que és un dels llenguatges més fàcils d'utilitzar, amb un enorme suport en la xarxa. Per tal d'agilitzar la capa REST he pensat en emprar el Framework Jersey, segons he llegit és àgil, potent i molt utilitzat.

Per tal de publicar-ho empraré el contenidor de applets Tomcat, ja que és el que més conec.

Com a base de dades empraré Postgresql, un motor de base de dades de codi lliure i molt complet.

Com a sistema operatiu empraré ubuntu server. Sistema operatiu que comença a agafar mercat en el món dels servidors. És gratuït, segur, de fàcil instal·lació i també en paquets.

Amb el servidor d'aplicacions finalitzat, ja tindrem el model de dades ben clar, la arquitectura a emprar, etc... En aquest punt ja podrem començar amb la interfície d'usuari.

Interfície d'usuari

La interfície d'usuari es crearà a partir d'un Framework anomenat Ionic. Una tecnologia híbrida que em permetrà crear una aplicació multiplataforma de forma àgil. Ionic, utilitza les tecnologies HTML5, AngularJS i Cordova/Phonegap com a eines principals de desenvolupament i «compilació» d'aplicacions. Eines altament testejades i que obtenen bons resultats. És important tenir en compte que segons uns últims estudis, la tecnologia híbrida estarà present el 80 % de les aplicacions per mòbil en el futur.

Per altra banda, l'aspecte gràfic ja es troba bastant avançat, amb un gran conjunt de menús, pestanyes, botons, checkboxes, sense tenir que implementar nous components css/html. Tot i així, el framework permet modificar al gust de cadascú qualsevol component de les que disposa.





Ara ja un cop tinguem la interfície d'usuari amb la capacitat de gestionar recordatoris des del terminal només ens queda el «servei andorid»

Servei Andorid

Aquest Servei serà una part implícita de la aplicació. «Cordova» permet crear serveis que s'executen background, per tant, d'una manera senzilla i des de el mateix codi de la aplicació i emprant eines del Framework, es pot crear el Servei Andorid que necessito. Per tant, amb el servidor creat, la interfície finalitzada, només quedaria perfilar el resultat en aquest punt.

Certament, aquestes tecnologies no em són gaire conegudes, però tinc clar que seria impossible presentar una aplicació definitiva si tingués que programar-ho tot des de zero.



1.4 Planificació del Treball

En primer lloc llistaré el conjunt d'eines, tecnologies i/o plataformes que seran necessàries per dur a terme el projecte.

Nom	Tipus	Descripció
VirtualBox	Sistema virtualització	S'emprarà per crear una màquina virtual que contindrà el servidor d'aplicacions.
Ubuntu-Server	Sistema operatiu	Sistema Operatiu base del servidor d'aplicacions
Ubunut-Desktop	Sistema operatiu	Sistema operatiu emprat per desenvolupar.
Postgresql	Motor de base de dades	Base de dades que contindrà el model de dades i les dades de la aplicació.
PgAdmin	GUI per gestionar BBDD	Eina gràfica per gestionar el motor de base de dades potgresql.
JDBC driver	Driver de BBDD	API que permetrà realitzar accions sobre la BBDD des de JAVA.
Apache-Tomcat	Contenedor applets	Contindrà el servidor d'aplicacions JAVA.
Java-sdk	Kit de desenvolupament	Emprat en la execució i desenvolupament de codi JAVA
Eclipse	IDE	Interfície de desenvolupament altament reconeguda.
Plugin Jersey	Plugin Eclipse	Plugin d'Eclipse per a facilitar el desenvolupament de WS-REST.
Plugin Thym	Plugin Eclipse	Plugin d'Eclipse que agilitza en desenvolupament amb Apache Cordova.
Android-SDK	Kit de desenvolupament	Emprat en la edició, compilació i execució d'aplicacions Andorid.
Framework Ionic	Framework	Framework per a la creació d'aplicacions híbrides.
Apache Cordova	Framework	Framework per a la creació d'aplicacions híbrides.



Planificació

Llegenda

- 2 hores diàries
- 4 hores diàries
- Data Important

Març

dl	dm	dc	dj	dv	ds	dg
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

* Durant el març es preveu realitzar el disseny i arquitectura del projecte.

* Dedicació TFC =56 hores.

Abril

dl	dm	dc	dj	dv	ds	dg
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

* **6 d'abril:** Finalització disseny i arquitectura

* Dedicació Disseny i Arquitectura: 68 hores

* A l'abril s'inicia el desenvolupament.

* Dedicació Desenvolupament 1^a entrega = 62 hores

* Dedicació TFC acabat abril =134 hores

Maig

dl	dm	dc	dj	dv	ds	dg
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

* **18 de maig:** 1^a entrega APP

* Dedicació 1^a entrega: 136 hores

* A l'abril s'inicia el desenvolupament entrega Final

* Dedicació entrega final maig: 34 hores

* Dedicació TFC acabat maig: 212 hores.

Juny

dl	dm	dc	dj	dv	ds	dg
			1	2	3	4
	6	7	8	9	10	11
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			

* **15 de juny:** Entrega Final APP

* Dedicació Entrega final: 74 hores

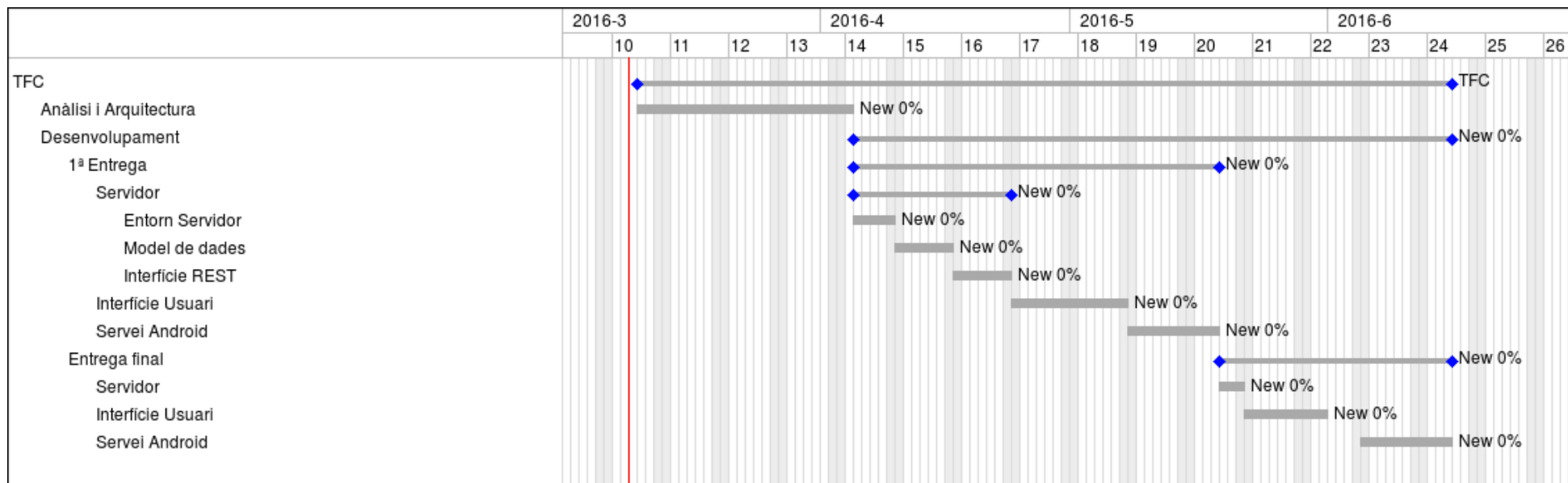
* Dedicació entrega final Juny: 36 hores

* Dedicació TFC acabat maig: 248 hores.



Per tal de gaudir d'una planificació simple, lineal i sense embussos. Assigno una determinada dedicació en cada àmbit del projecte. La creació de diferents nivells de tasques agrupades per «Entrega» em permet visualitzar clarament les dates més significatives, que són:

- Anàlisi i Arquitectura
- 1ª Entrega
- Entrega Final



Les entregues es troben agrupades en una tasca que he anomenat «Desenvolupament», una fase del cicle de vida d'un projecte. D'aquesta manera puc saber el temps invertit en Desenvolupament. Aquesta forma d'agrupar és realment interessant, per a quadrar calendaris.

Aquest gràfic que mostro l'he pogut crear gràcies a «Redmine», el gestor de tasques gratuït més populars entre els desenvolupadors.



1.5 Breu sumari de productes obtinguts

La entrega final es compondrà de diversos fitxers agrupats per entorn d'execució.

En primer lloc tenim el servidor:

- Arxiu comprimit que contindrà el codi font del servidor.
- Arxiu comprimit que contindrà el «schema» de la BBDD.
- Arxiu comprimit que contindrà dades de la BBDD.
- Arxiu comprimit que contindrà la parametrització dels diferents components del servidor.
- Instruccions per a la instal·lació i execució del servidor.

Nota: Si es vol simplificar la entrega, es proposa realitzar una còpia de la màquina virtual per fer-ne ús des de qualsevol servidor.

Després tenim la aplicació:

- Instal·lable APK que contindrà la aplicació.
- Arxiu comprimit que contindrà el codi font de la aplicació.
- Instruccions per a la instal·lació i execució de la aplicació.

1.6 Breu descripció dels altres capítols de la memòria

En els altres capítols de la memòria exposaré les diferents fases del projecte de forma gràfica i resumida. Al mateix temps mostraré el cost temporal invertit i si fa falta dedicar especial atenció en altres temes com seguretat, escalabilitat, redundància, etc. És evident que el TFC no pot cobrir totes aquelles necessitats que farien falta en el món real, però seria molt interessant tenir-les en comptes per si alguna vegada es posés en pràctica.

Bàsicament he desglossat les fases del projecte en:

- Arquitectura: Diagrames de flux de comunicacions i tecnologies.
- Disseny: Model de dades, objectes, interfície REST.
- Implementació: Lliberia, paquets, dependències.
- Proves: Joc de proves. Millores.
- Posta en producció: Servidor en el núvol, dispositius Android.

El conjunt de fases em permetrà definir tot l'entorn en la qual s'ha pogut portar a terme el projecte, incorporant els grans beneficis i les grans mancances de les que disposi.

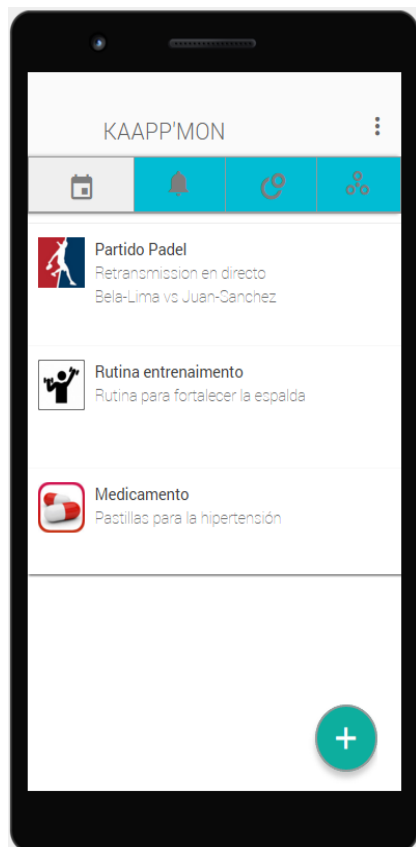


Disseny

Abans d'iniciar la programació ha sigut necessari realitzar un treball previ basat en realitzar un disseny que s'adapti als requeriments inicials.

Durant la PAC 2 es va afinar amb bastant detall la interfície de la aplicació que en el resultat final ha canviat una mica ja que el framework donava opcions més àgils i atractives.

Tot seguit mostrem els dissenys originals d'alta qualitat que es va realitzar i els presentaré de forma resumida.

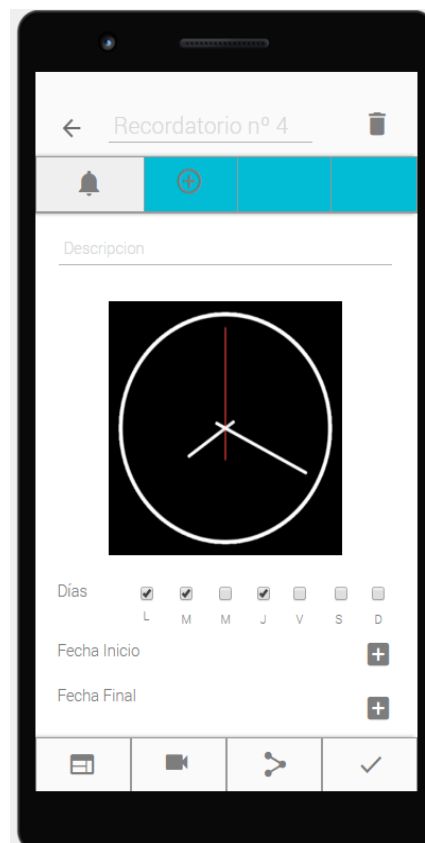


El menú principal és la pantalla inicial que s'accedeix des de la pantalla de login. Es llistaran els recordatoris creats en forma de llistat i pitjant sobre cadascun d'ells podem visualitzar la seva configuració. Podem veure que en tot moment tindrem accés a un menú en la capçalera que ens permetrà moure entre pantalles. El model de la resta de pantalles referents en el menú tenen una visió semblant.

Aquesta pantalla mostra la configuració d'un recordatori que ens permetrà canviar el títol, la descripció, l'hora d'execució de l'alarma i els dies en que es vol executar.

També tindrem la possibilitat d'escollir el rang de dates en que es vol que s'executi.

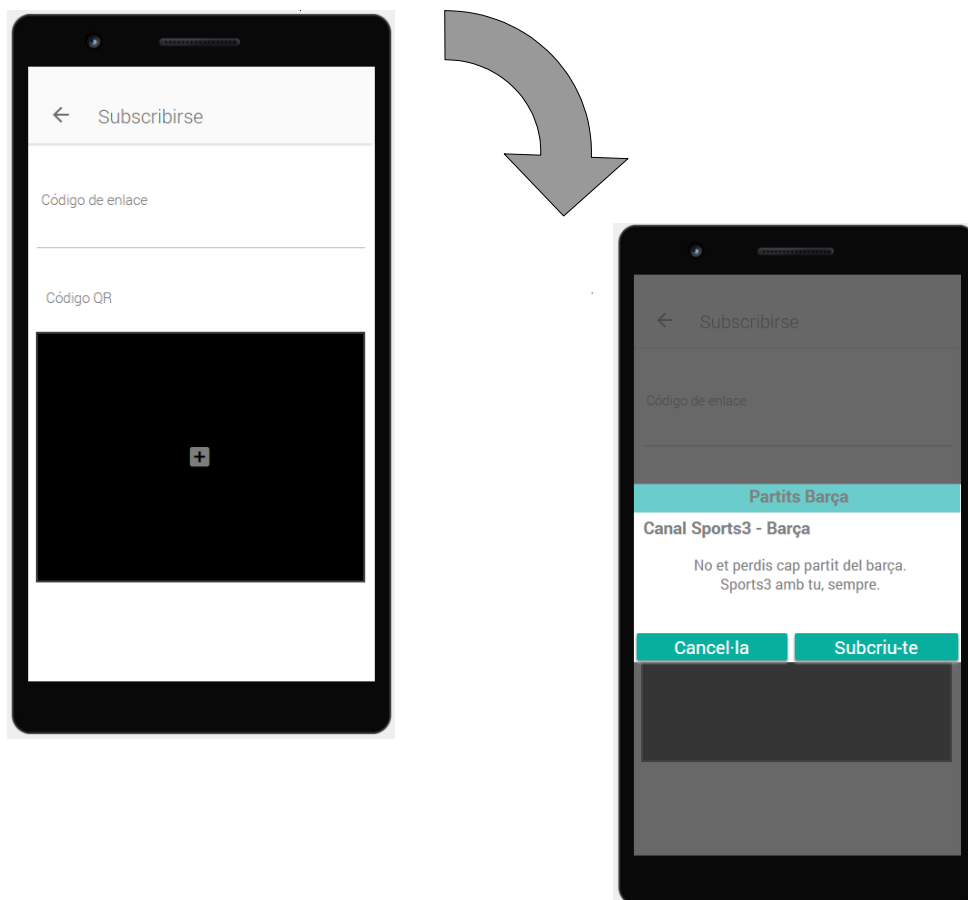
Finalment tindrem la possibilitat de compartir el recordatori amb els contactes que tinguem emmagatzemats.





Un dels grans potencials de la aplicació és la possibilitat de subscriure's a canals de recordatoris per a que tinguis un sistema d'alertes sense tenir que configurar-ho un mateix.

Kapp'mon oferirà la possibilitat de subscriure's a un canal de recordatori de diferents maneres, i com podem veure en la imatge és escanejar un codi QR que pots obtenir d'una Web, d'un diari o fins i tot de la pantalla d'un altre dispositiu.



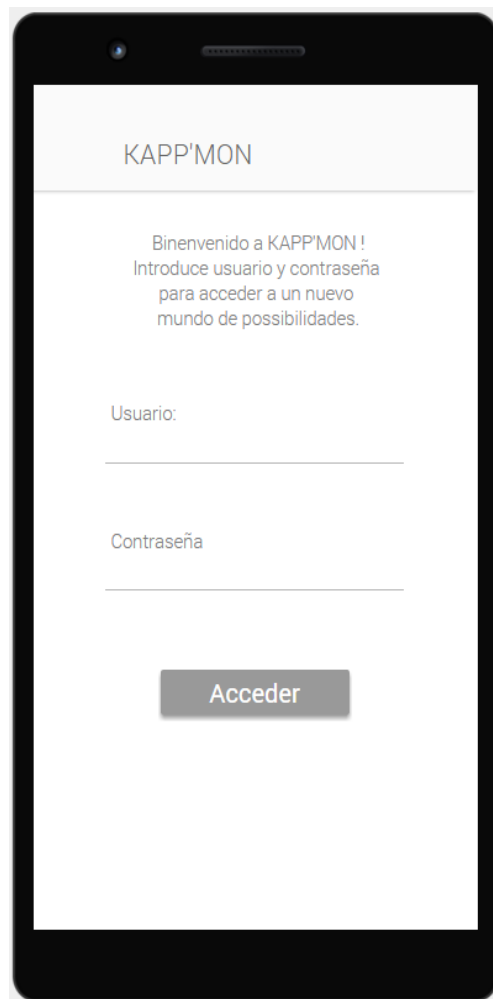
El mecanisme és ben senzill, iniciem el procés de lectura de QR, un cop el codi QR és escanejat, la aplicació mostrarà una finestra emergent amb la informació del recordatori.

Si el contingut correspon a un recordatori que desitgis pots acceptar que formi part del seu conjunt de recordatoris.



Finalment mostrem les pantalles més bàsiques, però essencials. En primer lloc tenim la pantalla de login a on en loguejarem o ens enregistrarem per poder accedir a la aplicació. A priori s'ha pensat en demanar el correu electrònic a part d'un nick, ja que lligar un compte Kapp'Mon amb un compte de correu Google permetria sincronitzar qualsevol acció referent a la aplicació a través de navegadors i fins i tot entre dispositius.

Encara que no es troba programat, un nou compte Kapp'Mon que va lligat a un correu electrònic hauria d'existir un sistema de validació de correu, a partir d'un link o partir d'un codi enviat per SMS.



Finalment la clau de tot resideix en gaudir de les notifikacions que volem rebre. Un exemple senzill veiem en la imatge de la esquerra, que mostra una finestra emergent amb el títol del recordatori, la descripció i la possibilitat d'aturar o repetir-la com si fos una alarma de mòbil.



Arquitectura

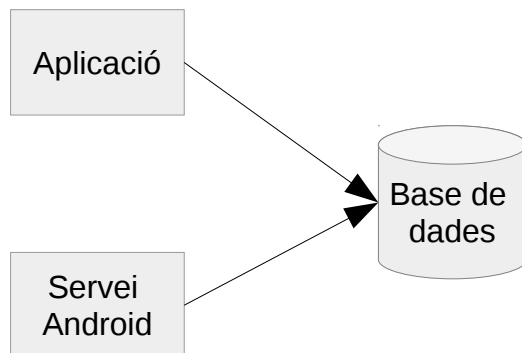
KAPP'MON és una aplicació que es basa en gestionar alarmes, i el seu objectiu és que s'executin quan toca. Per tant, el dispositiu mòbil no pot dependre d'un servidor centralitzat per completar aquesta tasca.

Per altra banda, la aplicació té amb un conjunt de funcionalitats que no serien possibles sense un servidor central, com seria la de l'enregistrament d'usuaris, recordatoris i canals de recordatoris per poder compartir-los amb altres usuaris.

Per tant, tenim 2 entorns de producció que han de compartir informació entre ells.

Aplicació nativa

En primer lloc tenim la arquitectura de la aplicació dins el dispositiu:



La aplicació enregistrarà tots els recordatoris, alarmes i canals que creï el usuari. D'aquesta manera podem oferir un servei OFFLINE, que realitzi les principals tasques sense necessitat de connectar-se a Internet.

En paral·lel la aplicació realitzarà dues tasques fonamentals en background. En primer lloc s'encarregarà d'encuar tota la nova informació emmagatzemada i enviar-la al servidor per a que sigui processada un cop tingui Internet. Així tenim un sistema de sincronització estable, sense gestionar els possibles problemes de comunicacions.

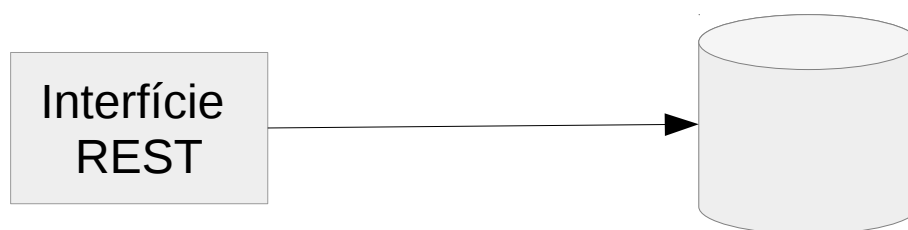
La pròpia aplicació contindrà un servei Android que s'encarregarà d'accedir a la base de dades per poder executar una alarma. Bàsicament, el servei obtindrà els recordatoris i crearà tasques



programades en background per a que s'executin encara que la aplicació estigui sense executar.

Servidor

El servidor que es trobarà en el núvol estarà compost per una interfície REST que serà accedida per peticions HTTP de la aplicació mòbil. La Interfície REST accedirà a la BBDD per Consultar, Inserir o Actualitzar les dades dels usuaris, els seus recordatoris, els canals i més important serà el responsable d'enllaçar i compatir recordatoris.

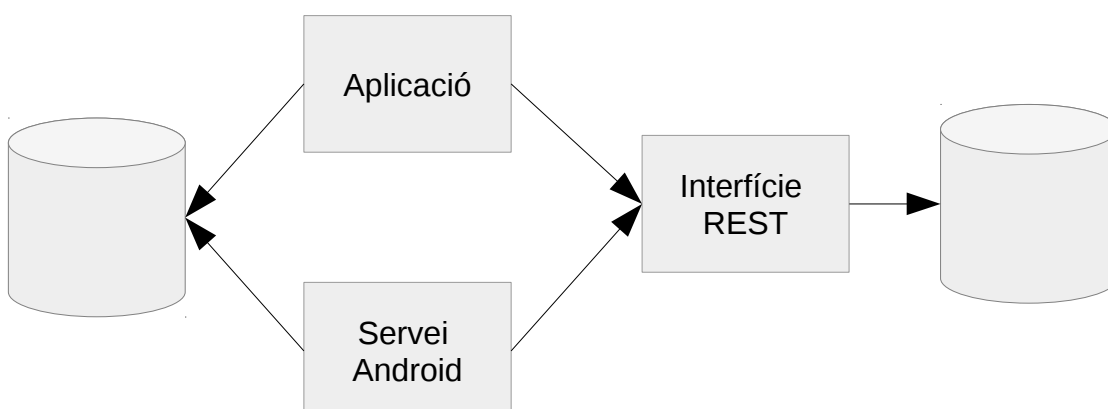


Es preveu que les dades sensibles dels usuaris s'emmagatzemin a la base dades amb encriptació DES o AES.

Arquitectura General

Tot seguit es mostra un diagrama resum de les comunicacions entre els dispositius mòbils i el servidor. Si no hi ha problemes de comunicacions, totes les accions dels usuaris passaran primer per una crida REST que el servidor rebrà per tal de processa-ho, en cas contrari el dispositiu mòbil seguirà fent la seva feina i comunicarà els canvis quan torni a tenir comunicació.

Les comunicacions seran segures emprant tecnologia SSL.





Un factor importantíssim que no apareix, és donar accés a la API de la Interfície REST del servidor per a que empreses en vulguin fer ús. És clau per a l'èxit de la aplicació que les empreses puguin accedir al servidor per automatitzar la creació de recordatoris, alarmes i canals de recordatoris. És important pensar en els diferents models de negoci que pot aportar la aplicació, i si és gratuïta la aplicació mòbil només queda l'accés a l'API REST.

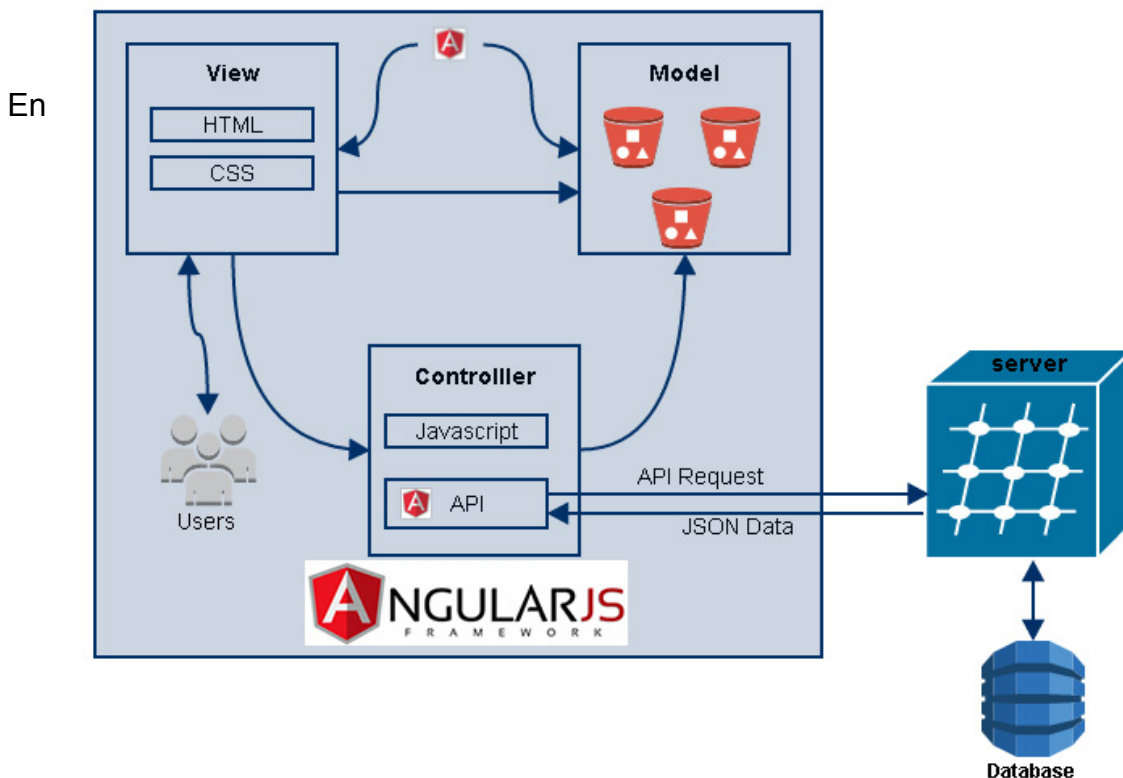
En els punts anteriors hem mostrat un resum de l'arquitectura plantejada en el projecte KAPP'MON. Però el gran repte serà en la compartició de dades. Com veurem al llarg de tot el document s'intentarà mantenir un model orientat a objectes en totes les sentits, des de la BBDD, passant per les classes implementades fins a les crides REST ja que la seva filosofia no està basada en la funcionalitat si no en la manipulació d'objectes.



Diagrama Arquitectura

MVC

El model vista controlador de la aplicació està basat en el framework AngularJS de Google. No hi ha millor tutor que el propi creador, per aquest motiu presento el diagrama que mostren en la seva documentació.



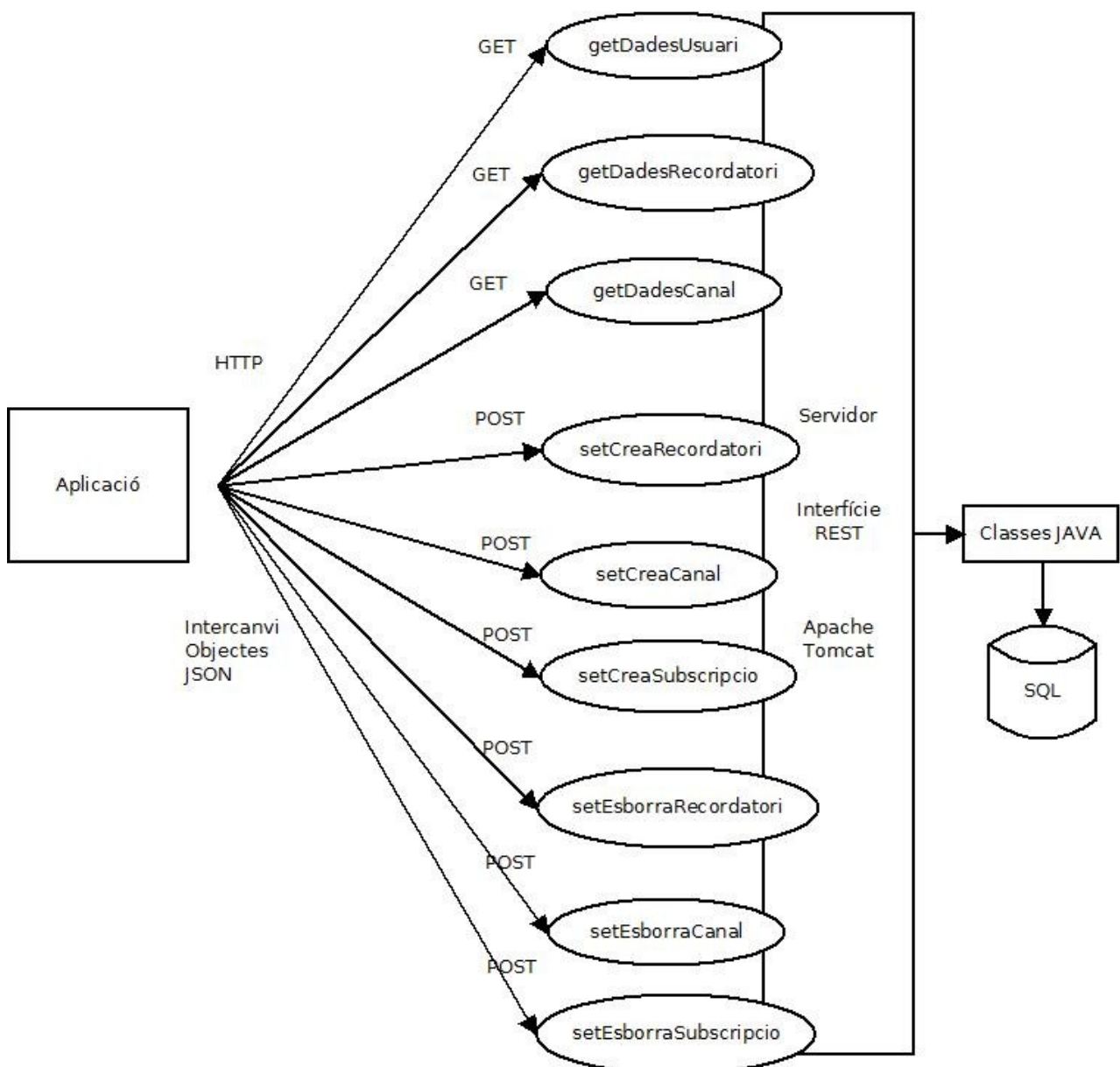
aquest diagrama podem observar que els usuaris interactuen amb la interfície de la aplicació, anomenada vista (view) corresponent a components html i css. La vista referencia el controlador apropiat per poder obtenir les dades, abreviadament és una interfície que connecta la interfície d'usuari amb el model de dades. Un cop obtingudes, es models gràcies al model(service). La vista pot reconèixer el model i presentar-lo en la interfície. Estem davant un triangle amorós que predomina en els la majoria d'aplicacions híbrides i pàgines WEB.



SOA – Arquitectura orientada a serveis

Tot seguit hem representat les comunicacions necessàries per portar a terme el projecte. Estem davant d'un servidor que ofereix una interfície REST muntada sobre la plataforma Apache Tomcat.

La aplicació disposa del Framework IONIC que gràcies a AngularJS i la llibreria REST es comunicarà fent peticions http per cadascuna de les funcionalitats que ofereix el servidor tai i com es pot veure en el diagrama.

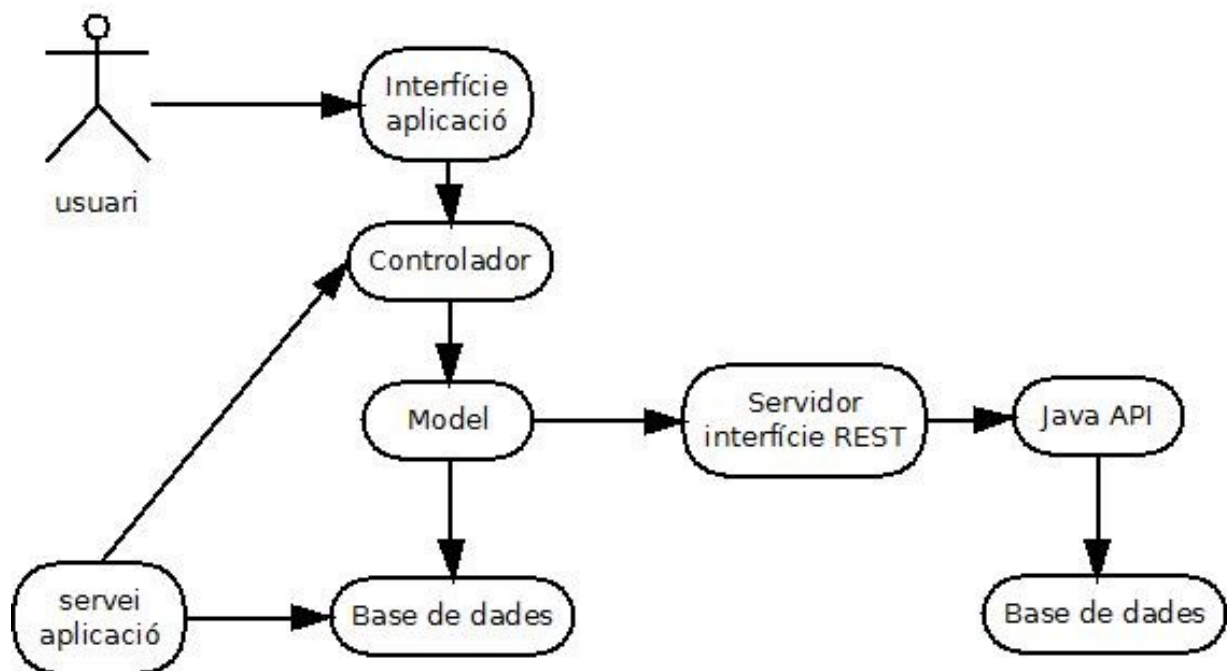




EDA - Arquitectura dirigida per events

El diagrama EDA ens mostra que tenim 2 llançadors d'eventualitats. Primerament tenim al usuari que interactua amb la interfície de la aplicació, aquest amb el controlador i finalment amb el model que connectarà amb la interfície REST del servidor i aquest amb la BBDD.

Per altra banda tenim al servei de la aplicació, que realitzarà tasques periòdiques, tant per gestionar la BBDD en local com per executar les alarmes. Com podem observar tant l'usuari com el servei comparteixen el mateix camí un cop accedeixen al controlador.





Desenvolupament

Servidor

El servidor es basa en oferir la informació necessària per a la aplicació a través d'una interfície REST a on s'intercanviaran objectes JSON si és necessari.

El **BACKEND** d'atenció de peticions REST s'ha desenvolupat amb l'IDE Eclipse i el projecte creat està compost per:

- Llenguatge [JAVA](#) , versió 8.
- Amb una interfície de treball propi de Eclipse gestionat pel plugin «[RestFull plugin](#)».
- Llibreria per a la edició de «WEB Services REST» [Jersey](#) amb Tecnologia [JAX-RS](#).
- Llibreria <http://wiki.fasterxml.com/JacksonDownload> per al tractament de dades JSON que es consumiran i es retornaran al client d'una forma transparent en el desenvolupament.
- Llibreria [JDBC](#) per al Accés a la BBDD.

Altres tecnologies a destacar:

- S'ha executat una màquina virtual amb [VirtualBOX](#). La [imatge](#) a executar s'ha descarregat d'Internet.
- [Ubuntu](#) server és el sistema operatiu escollit per a contenir el backend, al igual que s'ha instal·lat un mínim entorn gràfic per a poder desenvolupar-lo.
- S'ha emprat [PostgreSQL](#) com a motor de base de dades.
- S'ha emprat [PgAdmin](#) i [psql](#) per a gestionar la base de dades.
- S'ha emprat [Apache Tomcat 8](#) com a contenidor dels servlets, per tal de publicar el backend.

Eines de publicació i testeig:

- Eclipse ofereix la possibilitat de configurar un servidor Tomcat per a la execució de la API WS.
- Per tal de comprovar els resultats de les peticions GET s'ha emprat el Navegador [FireFox](#) i el complement [Firebug](#).
- Per altra banda s'ha emprat [SOAPUI](#) per a la manipulació de peticions GET i POST amb dades JSON incrustades en el cos del missatge HTTP.
- El possibles errors s'han pogut analitzar gràcies al sistema de logging de la consola Eclipse.



La API REST resultant en aquesta fase es pot descarregar a través d'un navegador a partir de la URL:

- <http://88.22.76.227:8080/kappmon/rest/application.wadl> .
- L'accés a aquesta especificació es pot realitzar gràcies a la configuració NAT del router que redirigeix les peticions HTTP a la màquina virtual.
- La impossibilitat d'accedir a aquesta especificació es podria donar per a un canvi en la IP pública proporcionada per el servei ISP. No obstant fa mesos que no canvia.
- De totes maneres adjuntaré com a annex el document xml resultant en la entrega. Nom del fitxer: kappmon.wadl.xml

El DLL i el DCL del «schema» de la base de dades es pot trobar en el fitxer adjunt anomenat schema_kappmon.sql.

La decisió d'emprar les tecnologies esmentades en el servidor es basen en:

- Llenguatge JAVA:
 - Multi-plataforma.
 - Fàcil aprenentatge.
 - Llenguatge orientat a Objectes.
 - Altíssima documentació.
 - Altíssima quantitat de llibreries.
 - JAVA no esdevé el llenguatge més ràpid en la execució ni en el consum de recursos, però s'ha demostrat contínuament que la seva evolució en noves versions que millora en aquest aspecte.
- Apache Tomcat:
 - Eina altament estesa.
 - Basada en JAVA.
 - Multi-plataforma.
 - Fàcil instal·lació, configuració i publicació de servlets.
 - Ofereix un gran conjunt de funcionalitats que redueix el desenvolupament i incrementa la seguretat i la escalabilitat del sistema.
- Jersey/Jackson
 - Llibreries JAVA de fàcil ús.
 - Altament esteses per donar una interfície transparent en el desenvolupament.
 - Facilitat de desenvolupament.
 - Altíssima documentació.



- PostgreSQL
 - Multi-plataforma.
 - Molt ràpid i poc consum de recursos.
 - Altament configurable.
 - Sistema de replicació natiu.
 - Gratuït.
 - Altíssima documentació.
 - Fàcil integració amb qualsevol sistema, llenguatge de programació i migració.
 - Alta compatibilitat amb qualsevol llenguatge per a generació de «Stored Procedures».
 - Constant evolució.
 - Un dels motors de base de dades més famosos per garantir la integritat de les dades.

- Linux (Ubuntu)
 - Sistema operatiu àgil per a desenvolupadors.
 - Gratuït.
 - Comunitat compromesa.
 - Configuració a mida.
 - Poc consum.

Com bé sabem, no podem afirmar que una única tecnologia és la més apropiada en tot moment, si no que segons les característiques del projecte, ja sigui per volum de peticions, seguretat, velocitat de resposta, tipus de dades etc..., s'hauria d'optar per altres eines. Per exemple si la nostra aplicació es basés en moure massivitat de dades i fitxers, un motor NOSQL seria més apropiat, o potser algun sistema híbrid. També podem dir que si estiguéssim davant un entorn transaccional de milions de peticions que necessiten resposta en milisegons, ens hauríem de plantejar emprar una alternativa a Tomcat/Java.

A nivell personal, no he volgut emprar altres sistemes com DJANGO, Firebase entre altres, ja que he pogut aquestes eines, encara que siguin ràpides, no hem permetrien programar els serveis web a mida ni poder treballar directament amb la BBDD.



Aplicació

La aplicació s'ha desenvolupat amb el pensament de que sigui una aplicació multi-plataforma, tant per IOS com per Android. Per tal d'aconseguir-ho he emprat tecnologia híbrida basada en fer ús del sistema de navegació del propi dispositiu, per tal de interpretar els llenguatges HTML, CSS i Javascript.

El framework [IONIC](#) ha sigut el pilar principal per arribar a tal propòsit. Ionic ofereix els següents components:

- Motor «[Apache Cordova](#)» com a marc de desenvolupament de tecnologia híbrida.
- Empra «[AngularJS](#)» com a sistema Model Vista Controlador desenvolupat amb JavaScript.
- Ionic empra [Node.js](#) com a entorn JavaScript d'execució.
- També ofereix un ampli conjunt de components visuals CSS que agilitzen la feina i que pretenen simular objectes visuals de dispositius mòbils.
- Com a framework basat en WEB permet desenvolupar amb HTML, CSS i Javascript.
- Finalment, IONIC ofereix eines de testeig, compilació, empaquetat i execució de les aplicació en les plataformes IOS i Android.

Llibreries complementàries:

- L'entorn «Apache Cordova» ofereix un ampli conjunt de [plugins](#) i llibreries que agilitzen el desenvolupament oferint grans funcionalitats altament testejades. En el nostre cas hem emprat «[Cordova Local-notification Plugin](#)» com a base de les notificacions generades pels recordatoris.
- Per altra banda, hem afegit llibreries JavaScript/CSS per a la gestió de calendaris i hores. Per calendaris he emprat «[ionic-datepicker](#)» i per les hores «[ionic-timepicker](#)». Dos llibreries altament recomanables, però que segurament no tenen l'èxit que té els plugins de cordova.
- Finalment, la manipulació de dates i hores esdevé un maldecap si es vol fer directament amb Javascript. Per aquest motiu vaig optar per escollir una llibreria que hem resolués qualsevol problema, anomenada [moment.js](#).

Per a portar a terme el projecte s'ha emprat l'IDE Eclipse i es compona per:

- Plugin [JBOSS](#) d'Eclipse, que agilitza la creació, edició, de projectes híbrid, al mateix temps que ofereix una interfície àgil per a la instal·lació i la resolució de dependències de llibreries.
- Al mateix temps, Eclipse afegeix automàticament els fitxers de configuració de projecte lligats amb el Framework Ionic.



Altres tecnologies a destacar:

- El sistema operatiu emprat en el desenvolupament del projecte és Xubuntu, ja que és un sistema més lleuger.
- S'han emprat altres eines per a la instal·lació de nous paquets i llibreries, com per exemple [Bower](#), el propi entorn cordova/ionic i el gestor de paquets npm de Node.js.

Eines de publicació i testeig

- El propi framework Ionic ofereix un sistema d'execució en la pròpia màquina de desenvolupament, on ofereix a través d'un navegador la interfície programada. El mateix navegador, en aquest cas FireFox, proporciona un bon sistema de debugging anomenat FireBug.
- Al mateix temps Ionic pot connectar amb els simuladors [AVD](#) i [Genymotion](#). Amb qualsevol dels gestors d'emuladors Android, Ionic ofereix la possibilitat de debugar a través de la consola amb la qual s'ha iniciat el procés d'execució. Genymotion ha sigut l'eina escollida ja que la velocitat d'execució és superior a la que ofereix AVD.
- Per altra banda, Eclipse ofereix la possibilitat de connectar amb un dispositiu físic connectat amb l'ordinador a través del port USB.
- Les proves reals s'han pogut realitzar gràcies a la generació del APK que ofereix Ionic i traslladar-lo al dispositiu mòbil a través del port USB.

Les decisions preses en l'ús d'aquestes tecnologies ja es troben explicades en els punts anteriors, però m'agradaria remarcar que estem davant una aplicació que a nivell d'interfície es basa en gestionar alarmes, recordatoris, etc, i per tant no té necessitat de grans components visuals com en un videojoc. Per tant, amb eines híbrides amb bons components visuals, obtenim bons resultats amb poc temps. Finalment, dic que he trobat a faltar una mica el desenvolupament en natiu per a guanyar experiència en aquest àmbit. No obstant, cordova et permet crear plugins que s'implementen nativament amb JAVA.



Estat del projecte

Les funcionalitats implementades fins a aquest permet que un usuari pugui gestionar els seus recordatoris, alarmes i canals des de la aplicació mòbil. Al mateix temps la pròpia aplicació ja realitza les notificacions relacionades.

A nivell d'aplicació ens trobem a un 85% finalitzat. Si més no, les característiques visuals i d'usabilitat no serien aptes per publicar-la en un «MarketPlace». El nivell de detall exigint per a una aplicació en producció no es pot portar a terme dins el període del Treball Final de Carrera, almenys per a que li agradi a la gent.

A nivell de servidor, ens trobem en un 95% finalitzat vers les funcionalitats demandades per la aplicació. Tot i així, un servidor de producció hauria de gaudir de molts més controls, com un sistema de persistència, un sistema de sessions, control de nombre de peticions, origen de les dades, millora en el tractament de les respostes, etc...

Com en tot projecte, els grans trets de la aplicació es programen relativament fàcil, però afinar cadascun dels seus aspectes pot portar un increment substancial en el cost total del desenvolupament del mateix. Per aquest motiu, moltes funcionalitats esmentades en PACs anteriors queden descartades en aquesta fase del projecte. El més important és sortir amb un producte que funcioni i que sigui atractiu, per pròximes publicacions ja s'incorporaran noves funcionalitats. De les descartades tenim:

- Afegir àudio en la alarma.
- Afegir vídeo en la alarma.
- Afegir foto en la alarma.
- Afegir docs en la alarma.
- Subscriure's a un canal a partir d'un codi QR.
- Compartir recordatoris a partir del llistat telefònic.
- Múltiples alarmes per recordatori.

Nota: Totes aquestes funcionalitats s'esmentaven com a extres en la PAC2, que haguessin pogut implementar-se si el temps ho permetia.



Problemàtiques

He volgut afegir aquest apartat per esmentar els obstacles que m'he trobat durant la fase de desenvolupament, molts d'ells per falta de coneixement de les tecnologies emprades, però que tot i així podrien millorar.

Servidor

- Tot i que podem trobar molta informació respecte les llibreries Jersey/Jackson, ha sigut difícil resoldre les dependències de les mateixes dins el projecte d'Eclipse, ja que en un principi volia emprar [Maven](#), com a gestor de paquets, però vaig acabar per afegir manualment les llibreries que necessitava, ja que portava incompatibilitats amb la versió de JAVA i altres llibreries, a més de que Eclipse tenia problemes de path per publicar-me el servei amb Tomcat.
- Per tal d'agilitzar feina, vaig instal·lar un servidor X a la màquina virtual per poder programar directament amb l'eclipse des de la mateixa. L'accés a la màquina la feina amb un escriptori remot RDP, però que no mapejava correctament el teclat, també tenia problemes amb LOCALES, etc. Finalment vaig trobar la solució emprant l'eina «[ThinLinc](#)», bàsicament una eina d'escriptori remot amb el seu propi protocol.
- L'accés a la base de dades es gestiona amb JDBC i aquest s'incorporava al projecte amb la interfície JDNI que permet la cerca de llibreries dins el directori. El problema concurrent que hem trobava és que les connexions amb la BBDD configurades es creaven directament sense tenir que programar, però no eren emprades correctament i cada cop que realitzava una crida amb la BBDD no hem tancava la sessió i al cap de poques crides la aplicació es bloquejava per la espera de poder accedir a la BBDD. Finalment, per falta d'experiència, i crec que per poca documentació vers a una eina tan transparent, he tingut que crear el meu propi gestor de base de dades(DAO per connexions) que obre i tanca connexió a cada petició. Completament ineficient, a més de no disposar del POOL de connexions que en teoria m'oferia la pròpia interfície.



Aplicació

- Per a la instal·lació de Ionic i la creació d'un projecte, és necessari instal·lar Node.js. Tenim varies maneres de fer-ho com per exemple des dels propis repositoris Ubuntu, descarregant-se un fitxer, etc. el problema és que la instal·lació de node.js per si sol no ha fet que es pogués crear una projecte Ionic, tal i com diuen des de la seva pàgina WEB, si no he tingut que descarregar altres paquets, fer «downgrades» i «upgrades», instal·lar-los manualment, fins que tot ha funcionat al uníson.
- Eclipse no ha sigut la millor elecció per desenvolupar un projecte IONIC. En primer lloc funcionalitats com la auto-completació de codi, navegació entre fitxers, i d'altres ha sigut un fiasco. Recomanaria emprar altres eines com, Sublime, Visual Studio Code o Webstorm.
- A nivell de testing, és un plaer provar la interfície des d'un navegador, però si vols provar-ho amb un emulador o directament des d'un dispositiu físic, han sorgit molts problemes. En primer lloc, AVD no pot executar un emulador al mateix temps que s'està executant virtualbox ja que comparteixen el mateix mòdul «kvm» i per a que funcioni tens la possibilitat d'eliminar aquest mòdul, però llavors tant virtualbox com AVD funcionen extremadament lent.
- Les primeres proves fetes amb el navegador varen ser tedioses ja que el navegador es queixava donat problemes CORS(Control Accés HTTP), on no hem permetia fer cridar dins la mateixa xarxa. Vaig tenir que crear un proxy dins l'aplicatiu que només funciona pel navegador i no des d'un dispositiu.
- Les vistes HTML no acaben de ser suficientment acurades per dispositius Android i els components de vegades no es mostren centrats.
- AngularJS disposa de la funcionalitat \$localStorage que el que fa és guardar informació en el propi dispositiu, el problema esdevé quan per versions d'Andorid superiors a la 5.0 aquesta informació no és permanent si es tanca la aplicació. Més endavant s'hauran d'emprar altres eines com SQL Lite per tenir dades persistents.
- Finalment, encara que no ha sigut un gran problema, alguns dels plugins cordova no es poden testejar des del navegador si no que s'han de testejar directament amb el dispositiu.

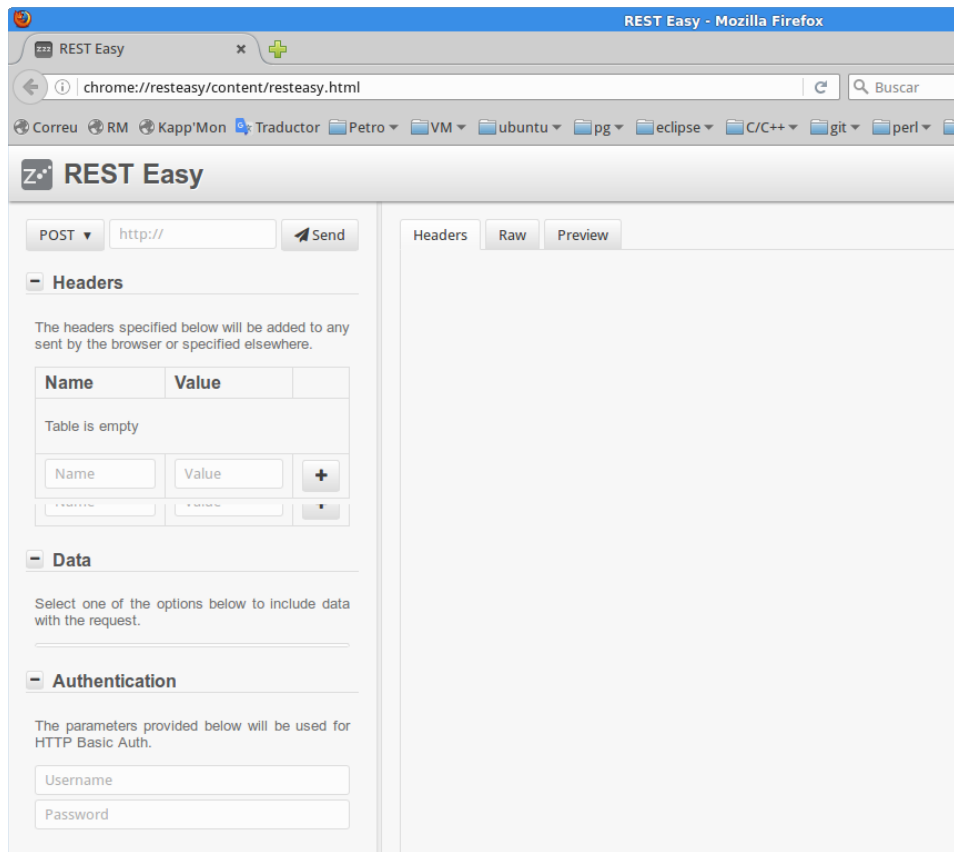


Testing

En aquest apartat mostrarem el conjunt d'eines que es poden emprar per al testinng de la aplicació i del servidor.

Servidor

El servidor ofereix una API REST que pot ser testejada amb diferents eines. De les més emprades, el navegador amb sistema de debug com el Firefox i Firebug, permeten realitzar probes unitàries per a cada Web Service. També existeixen altres plugins que permeten manipular el missatge HTTP per realitzar POST amb contingut JSON entre d'altres. Un exemple és el plugin «REST Easy» de Firefox. Aquesta pràctica és molt habitual, però limitada.





Per solventar les limitacions dels navegadors si els seus pluguins tenim una eina molt recomanada, SOAPUI. Aquesta aplicació permet testear interfícies SOAP i REST a partir del fitxer descriptor corresponent. WSDL(SOAP), WADL(REST). En el nostre cas tenim una API REST. SOAPUI també ofereix eines de test d'integració i de càrrega, encara que algunes d'aquestes característiques són de pagament.

The screenshot displays the SoapUI 5.2.1 interface. On the left, the 'Navigator' pane shows a project tree with the following structure:

- application
 - /login [/kappmon/rest/login]
 - POST - postLogin
 - Request 1
 - /register [/kappmon/rest/login/register]
 - /canals [/kappmon/rest/canals]
 - /alarmes [/kappmon/rest/alarmes]
 - /recordatoris [/kappmon/rest/recordatoris]
 - /subscripcions [/kappmon/rest/subscripcior]
 - /accessToken [/kappmon/rest/accessToken]
 - /requestToken [/kappmon/rest/requestTok]

The main workspace shows 'Request 1' configuration:

- Method: POST
- Endpoint: http://88.22.76.227:8080
- Resource: /kappmon/rest/login/login
- Parameters: (empty)

The 'Raw' tab shows the request body as a JSON object:

```
{
  "username": "ruben",
  "password": "ruben"
}
```

The 'Raw' tab also shows the response body as a JSON array of objects:

```
[
  {
    "id": 134,
    "token": "7e910aa8fb",
    "titol": "Prova",
    "descripcio": "V ZXCVZXCVZXCVoC VZ",
    "usuari_id": 1,
    "alarmes": [
      {
        "id": 116,
        "recordatori_id": 134,
        "hora": "20:12:00",
        "dia_inici": "2016-05-20",
        "dia_fi": "2016-05-30",
        "dies": [
          true,
          true,
          true,
          true,
          true,
          true
        ]
      }
    ]
  },
  {
    "id": 139,
    "token": "d43cedd739",
    "titol": "Jaume",
    "descripcio": "Nova Descripcio",
    "usuari_id": 1,
    "alarmes": [
      {
        "id": 121,
        "recordatori_id": 139,
        "hora": "20:26:00",
        "dia_inici": "2016-05-14",
        "dia_fi": "2016-05-16",
        "dies": [

```

At the bottom, the status bar indicates 'response time: 412ms (1622 bytes)' and '1 : 1'.



Aplicació

En la fase de desenvolupament la forma més àgil d'anar provant la interfície de la aplicació és emprar el provi Framework Ionic. Ionic ofereix una capa de servei WEB que serveix la interfície des del propi navegador. Bàsicament és el mateix que es fa quan es desenvolupa un Web però sense la necessitat de muntar un Tomcat o un altre contenidor d'aplicacions.

The screenshot displays a web browser window on the left and its developer tools on the right. The browser window shows a mobile application interface with a navigation bar and a list of items. The developer tools show network requests and their JSON responses.

Browser Window (Recordatoris - Mozilla Firefox):

- Address bar: localhost:8100/#/tal
- Navigation bar: Recordatoris, Canals, Subscripcions, Alarmes, Configuracio
- List of items:
 - mas2 (Nova Descripcio)
 - Prova (V ZXCZXCZXCVoC VZ)
 - Jaume (Nova Descripcio)
 - Jaume cata (Nova Descripcio)
 - Nou Titol (Nova Descripcio)
 - Nou Titol 12345 (Nova Descripcio)

Developer Tools (Firebug - Recordatoris):

- Console: GET http://localhost:8100/templates/sub-detail.html 200 OK 15ms
- Console: GET http://localhost:8100/templates/tab-sub.html 200 OK 1ms
- Console: GET http://localhost:8100/templates/canal-detail.html 200 OK 14ms
- Console: GET http://localhost:8100/templates/tab-canal.html 200 OK 18ms
- Console: GET http://localhost:8100/templates/tab-dash.html 200 OK 57ms
- Console: GET http://localhost:8100/templates/tabs.html 200 OK 51ms
- Console: POST http://localhost:8100/api/kappmon/login/login 200 OK 22ms
- Console: GET http://localhost:8100/api/kappmon/recordatoris/all/1 200 OK 32ms

JSON Response for POST http://localhost:8100/api/kappmon/login/login:

```
{
  "usuari": {
    "username": "ruben",
    "password": "ruben",
    "id": 1,
    "recordatoris": [
      {
        "id": 133,
        "token": "b26bf6ecad",
        "titol": "mas2",
        "descripcio": "Nova Descripcio",
        "usuari_id": 1,
        "alarmes": [
          {
            "id": 115,
            "recordatori_id": 133,
            "hora": "11:31:00",
            "dia_inici": "2016-05-14",
            "dia_fi": "2016-05-15",
            "dies": [true, true, true, true, true, true, true]
          },
          {
            "id": 134,
            "token": "7e910aa8fb",
            "titol": "Prova",
            "descripcio": "V ZXCZXCZXCVoC VZ",
            "usuari_id": 1,
            "alarmes": [
              {
                "id": 116,
                "recordatori_id": 134,
                "hora": "20:12:00",
                "dia_inici": "2016-05-20",
                "dia_fi": "2016-05-30",
                "dies": [true, true, true, true, true, true, true]
              },
              {
                "id": 139,
                "token": "d43cedd739",
                "titol": "Jaume",
                "descripcio": "Nova Descripcio",
                "usuari_id": 1,
                "alarmes": [
                  {
                    "id": 121,
                    "recordatori_id": 139,
                    "hora": "20:26:00",
                    "dia_inici": "2016-05-14",
                    "dia_fi": "2016-05-16",
                    "dies": [true, true, true, true, true, true, true]
                  },
                  {
                    "id": 150,
                    "token": "568ad9816b",
                    "titol": "Nou Titol 12345",
                    "descripcio": "Nova Descripcio",
                    "usuari_id": 1,
                    "alarmes": [
                      {
                        "id": 132,
                        "recordatori_id": 150,
                        "hora": "17:10:11",
                        "dia_inici": "2016-05-14",
                        "dia_fi": "2016-05-14",
                        "dies": [true, true, true, true, true, false]
                      },
                      {
                        "id": 141,
                        "token": "f77646bc0c",
                        "titol": "Jaume cata",
                        "descripcio": "Nova Descripcio",
                        "usuari_id": 1,
                        "alarmes": [
                          {
                            "id": 123,
                            "recordatori_id": 141,
                            "hora": "12:55:40",
                            "dia_inici": "2016-05-09",
                            "dia_fi": "2016-05-25",
                            "dies": [true, true, true, true, false, false, false]
                          },
                          {
                            "id": 146,
                            "token": "e5d43176f9",
                            "titol": "Nou Titol",
                            "descripcio": "Nova Descripcio",
                            "usuari_id": 1,
                            "alarmes": [
                              {
                                "id": 128,
                                "recordatori_id": 146,
                                "hora": "10:33:47",
                                "dia_inici": "2016-05-14",
                                "dia_fi": "2016-05-31",
                                "dies": [true, true, true, true, true, true, true]
                              }
                            ]
                          }
                        ]
                      }
                    ]
                  }
                ]
              }
            ]
          }
        ]
      }
    ]
  },
  "result": {
    "result": "LOGIN_OK",
    "resultmsg": ""
  }
}
```

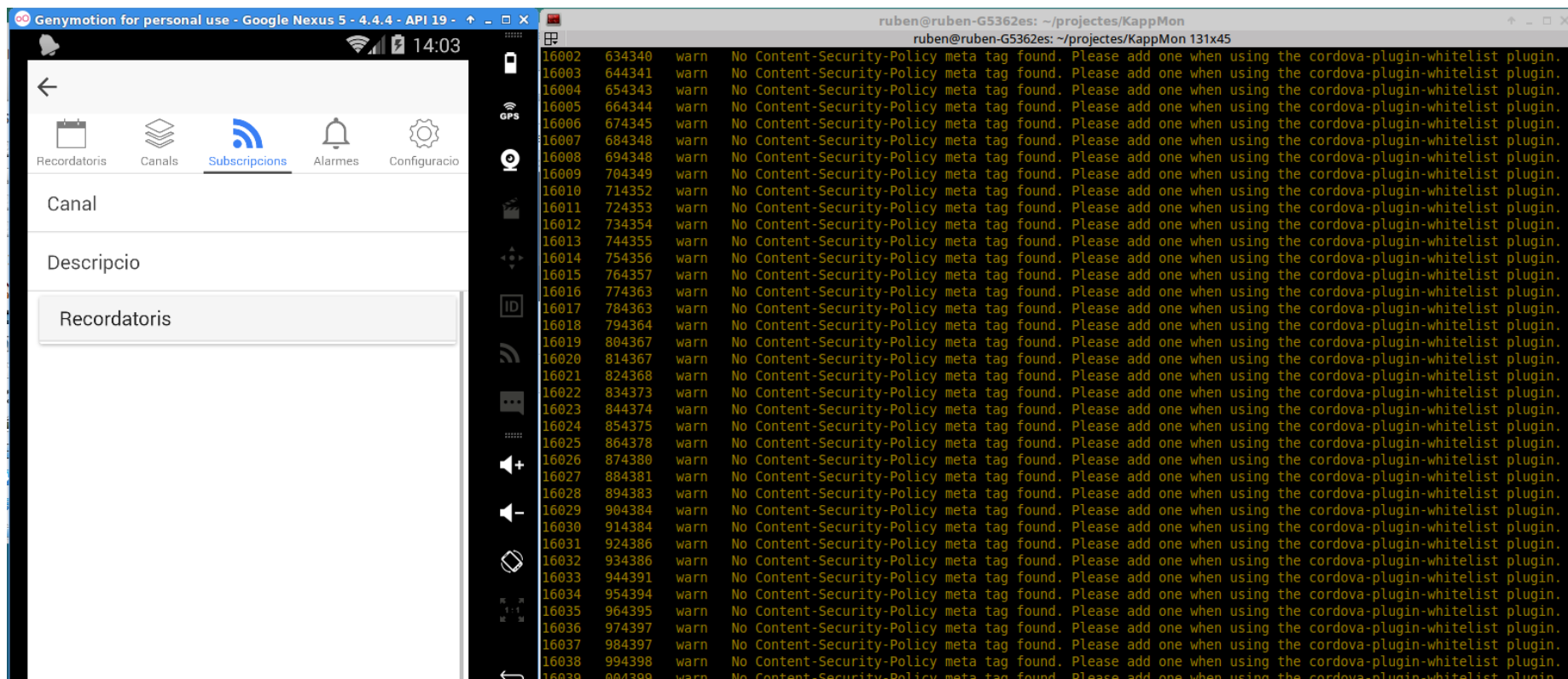
JSON Response for GET http://localhost:8100/api/kappmon/recordatoris/all/1:

```
[
  {
    "id": 133,
    "token": "b26bf6ecad",
    "titol": "mas2",
    "descripcio": "Nova Descripcio",
    "usuari_id": 1,
    "alarmes": [
      {
        "id": 115,
        "recordatori_id": 133,
        "hora": "11:31:00",
        "dia_inici": "2016-05-14",
        "dia_fi": "2016-05-15",
        "dies": [true, true, true, true, true, true, true]
      },
      {
        "id": 134,
        "token": "7e910aa8fb",
        "titol": "Prova",
        "descripcio": "V ZXCZXCZXCVoC VZ",
        "usuari_id": 1,
        "alarmes": [
          {
            "id": 116,
            "recordatori_id": 134,
            "hora": "20:12:00",
            "dia_inici": "2016-05-20",
            "dia_fi": "2016-05-30",
            "dies": [true, true, true, true, true, true, true]
          },
          {
            "id": 139,
            "token": "d43cedd739",
            "titol": "Jaume",
            "descripcio": "Nova Descripcio",
            "usuari_id": 1,
            "alarmes": [
              {
                "id": 121,
                "recordatori_id": 139,
                "hora": "20:26:00",
                "dia_inici": "2016-05-14",
                "dia_fi": "2016-05-16",
                "dies": [true, true, true, true, true, true, true]
              },
              {
                "id": 141,
                "token": "f77646bc0c",
                "titol": "Jaume cata",
                "descripcio": "Nova Descripcio",
                "usuari_id": 1,
                "alarmes": [
                  {
                    "id": 123,
                    "recordatori_id": 141,
                    "hora": "12:55:40",
                    "dia_inici": "2016-05-09",
                    "dia_fi": "2016-05-25",
                    "dies": [true, true, true, true, false, false, false]
                  },
                  {
                    "id": 146,
                    "token": "e5d43176f9",
                    "titol": "Nou Titol",
                    "descripcio": "Nova Descripcio",
                    "usuari_id": 1,
                    "alarmes": [
                      {
                        "id": 128,
                        "recordatori_id": 146,
                        "hora": "10:33:47",
                        "dia_inici": "2016-05-14",
                        "dia_fi": "2016-05-31",
                        "dies": [true, true, true, true, true, true, true]
                      }
                    ]
                  }
                ]
              }
            ]
          }
        ]
      }
    ]
  }
]
```



Com es pot observar en la imatge el servei és accessible des de <http://localhost:8100/> . Per tal de engegar el servei ens hem de trobar dins la carpeta del projecte i executem pel terminal `$ ionic serve`.

Amb el primer sistema podem testejar el 80% de la aplicació, la resta no es pot provar perquè són funcionalitats que interactuen amb un dispositiu mòbil. Per aquest motiu emprarem genymotion com a gestor d'emuladors Andorid i el Framework Ionic per a iniciar el procés de testeig i debug. Un cop tenim un emulador executant-se, escriurem `$ ionic run -l -c` en un terminal per iniciar el testeig.





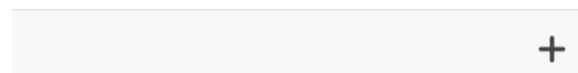
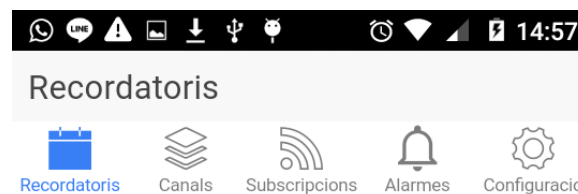
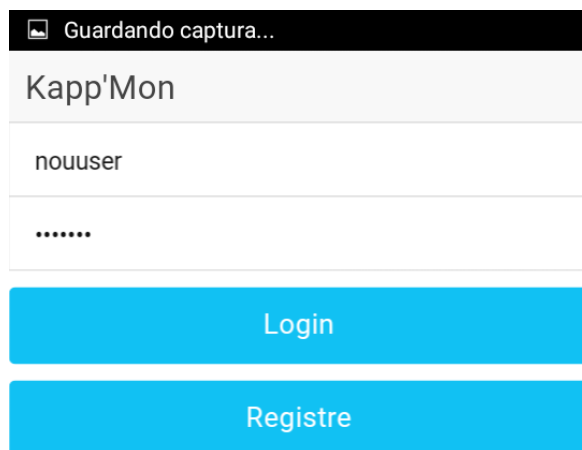
Finalment podem finalitzar el nostre testeig directament amb un dispositiu físic. El Framework ionic genera l'APK que nosaltres traslladarem directament al dispositiu a través del port USB. Un cop copiat, pitjant sobre l'APK s'inicia el procés d'instal·lació i tot seguit ja podem iniciar les nostres proves.



Proves unitàries

Registre

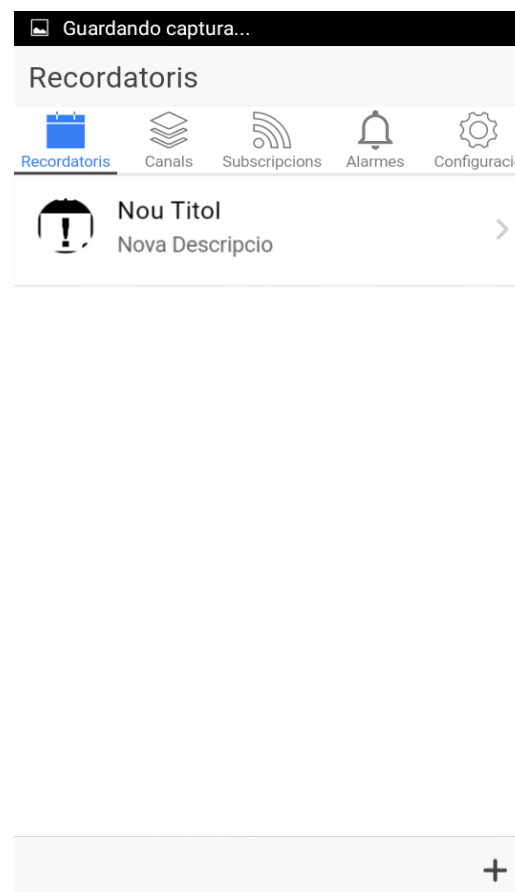
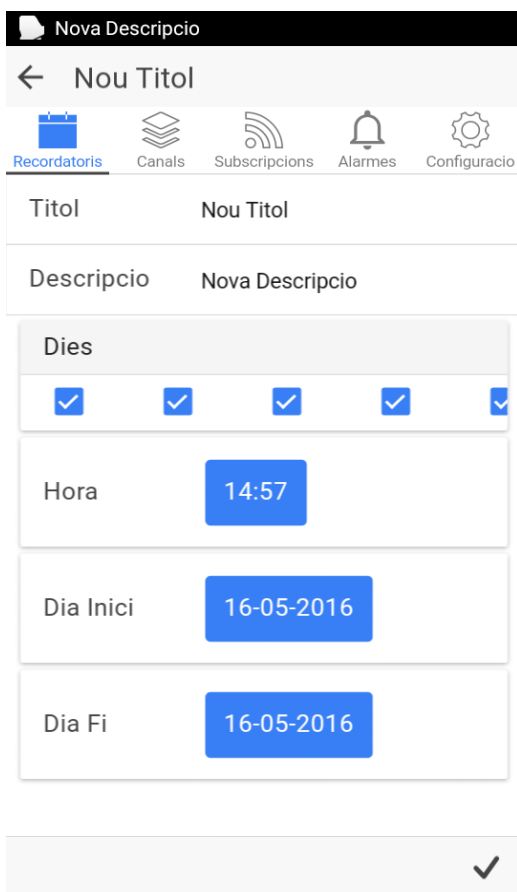
En la pantalla de login/registre es demana introduir un usuari i una contrasenya per poder enregistrar-se. Un cop introduïts i si l'usuari no es troba enregistrat en la BBDD del servidor, la aplicació accedirà al menú principal.





Nou Recordatori

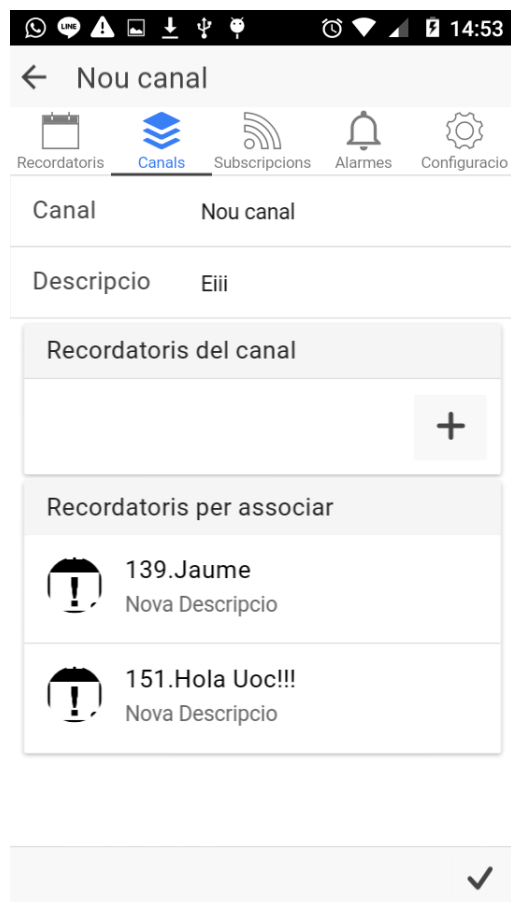
En la pantalla principal observarem que no tenim cap recordatori i procedirem a crear-ne un de nou. Hem de pitjar el botó [+] que veurem a la part més baixa de la pantalla. Un cop pitjat ens obrirà la pantalla de Nou recordatori(imatge 1) i procedirem a configurar el nou recordatori. Mantindrem el títol i la descripció per defecte. Configurarem el dies que vulguem, l'hora la data d'inici i la data final i pitjarem el botó [OK]. En aquell instant es mostrarà un pop-up dient que el recordatori s'ha creat correctament i la aplicació mostrarà el llistat de recordatoris creats en la pantalla principal(imatge 2). En aquest cas només un.





Canals

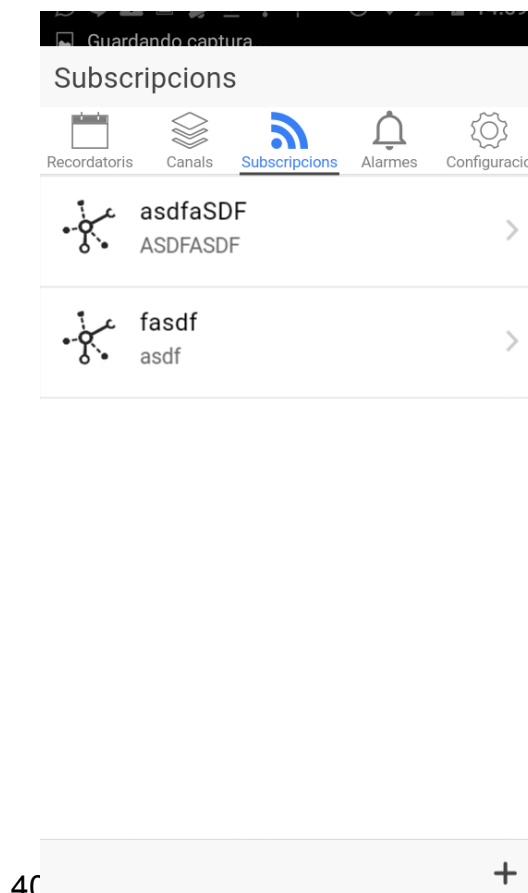
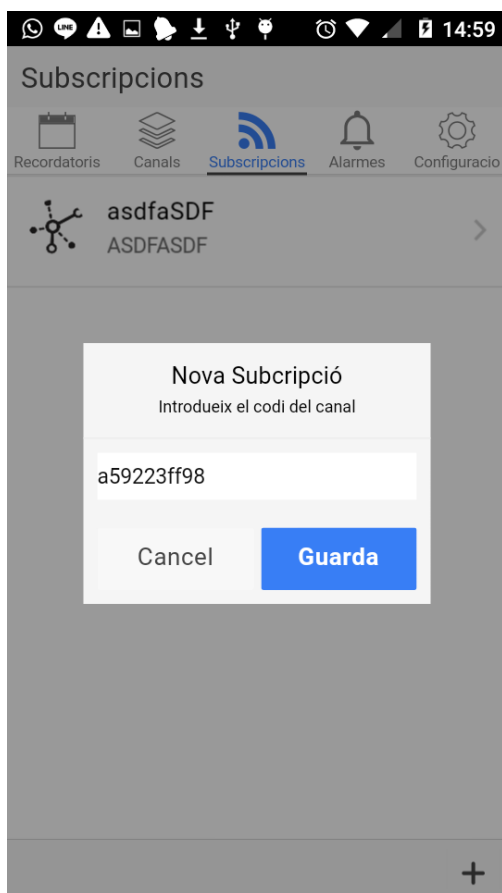
Pitjant sobre la pestanya Canals que trobarem sota la capçalera de la aplicació, podrem veure el llistat de canals que hem creat. En aquest cas ja hem creat uns quants canals i podem veure el llistat en la imatge 1. Per tal de crear-ne un de nou, pitjarem sobre la icona [+] i se'ns obrirà una nova finestra per crear un nou canal (imatge 2). en aquesta pantalla podrem posar el nom del canal la descripció, i afegir recordatoris que encara no es troben incorporats en un altre canal. Si pitgem [OK], podrem veure que el canal s'ha creat i veurem el veure el llistats de canals.





Subscripcions

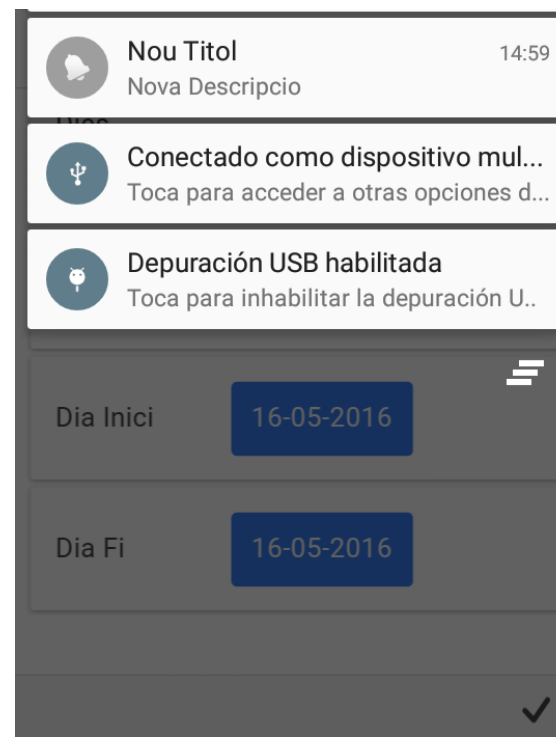
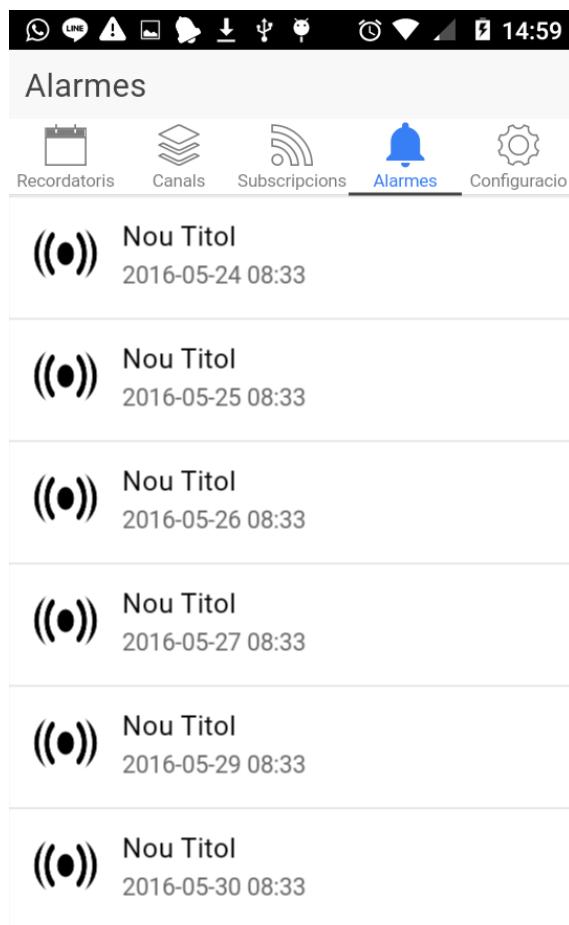
Les subscripcions bàsicament són canals d'altres usuaris que s'han pogut descarregar en el propi dispositiu per a que les notificacions relacionades s'executin sense tenir que configurar res. Per tal de subscriure's a un altre canal, hem de pitjar la icona [+] que es troba en el llistat de subscripcions. Un cop pitjat apareixerà un pop-up demanant que s'introdueixi el codi del canal. Un cop s'introdueix es pitja el botó [Guarda] i tot seguit, si el codi és correcte, el canal ja formarà part de les nostres subscripcions.





Alarmes

Les alarmes són notificacions que es generaran localment un cop s'hagin creat els recordatoris corresponents i/o canviat la configuració. Les notificacions ja realitzades no es mostraran en aquest llistat(imatge 1). En la imatge 2 podem veure com la notificació ens apareix per pantalla un cop hagi arribat l'hora d'executar-se. Si pitgem sobre la notificació se'ns obrirà automàticament la aplicació.





Llicències

Servidor

Les llicències es troben en fitxers separats dins la carpeta arrel del projecte KappMon del servidor.

POSTGRESQL

- *«Liberal Open Source license, similar to the BSD or MIT licenses.»*.
- **Llicència completa:** Fitxer LICENSE-POSTGRES.TXT

JAVA

- *«Subject to the terms and conditions of this Agreement including, but not limited to, the Java Technology Restrictions of the Supplemental License Terms, Oracle grants you a non-exclusive, non-transferable, limited license without license fees to reproduce and use internally the Software complete and unmodified for the sole purpose of running Programs»*
- **Llicència completa:** Fitxer LICENSE-JAVA.TXT

JERSEY

- *«Jersey is dual licensed under 2 OSI approved licenses :*
 - *COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL - Version 1.1)*
 - *GNU General Public License (GPL - Version 2, June 1991)»*
- **Llicència completa:** Fitxer LICENSE-JERSEY.TXT

JACKSON

- *«Jackson 1.x is offered under one of two OSS licenses:*
 - *Apache License (AL) 2.0.*
 - *LGPL 2.1»*
- **Llicència completa:** Fitxer LICENSE-JACKSON.TXT



KAPPMON

«

SUMMARY

- License does not expire.
- Can be used for creating unlimited applications
- Can be distributed in binary or object form only
- Non-commercial use only
- Can modify source-code and distribute modifications (derivative works)
- Parts of the software are provided under separate licenses, as follows:
 - AngularJS 1.0 is under the MIT license.
 - IONIC 1.0 is under the MIT license.
 - Apache Cordova under the Apache License.
 - PostgreSQL 9.3 under the MTI and BSD License
 - JAVA 1.7 under the Oracle LICENSE
 - JERSEY 1.19 under the GPL License
 - JACKSON 2.7.0 under the Apache 2.0 and LGPL 2.1 License

«

- **Llicència completa:** Fitxer LICENSE-KAPPMON.TXT



Aplicació

Les llicències es troben en fitxers separats dins la carpeta arrel del projecte KappMon de la aplicació.

APACHE CORDOVA

- «*Apache License (AL) 2.0.*»
- **Llicència completa:** Fitxer LICENSE-CORDOVA.TXT.

IONIC

- «The MIT License»
- **Llicència completa:** Fitxer LICENSE-IONIC.TXT.

ANGULARJS

- «The MIT License»
- **Llicència completa:** Fitxer LICENSE-ANGULARJS.TXT.

CORDOVA LOCAL NOTIFICATIONS

- «*Apache License (AL) 2.0.*»
- **Llicència completa:** Fitxer LICENSE-CORDOVA-LOCAL-NOTIFICATIONS.TXT.

IONIC DATEPICKER

- «The MIT License»
- **Llicència completa:** Fitxer LICENSE-IONIC-DATEPICKER.TXT.

IONIC TIMEPICKER

- «The MIT License»



- **Llicència completa:** Fitxer LICENSE-IONIC-TIMEPICKER.TXT.

KAPPMON

«

SUMMARY

- License does not expire.
- Can be used for creating unlimited applications
- Can be distributed in binary or object form only
- Non-commercial use only
- Can modify source-code and distribute modifications (derivative works)
- Parts of the software are provided under separate licenses, as follows:
 - AngularJS 1.0 is under the MIT license.
 - IONIC 1.0 is under the MIT license.
 - Apache Cordova under the Apache License.
 - PostgreSQL 9.3 under the MTI and BSD License
 - JAVA 1.7 under the Oracle LICENSE
 - JERSEY 1.19 under the GPL License
 - JACKSON 2.7.0 under the Apache 2.0 and LGPL 2.1 License

«

- **Llicència completa:** Fitxer LICENSE-KAPPMON.TXT



Conclusions

Noves oportunitats

En primer lloc, puc assegurar que la decisió de crear una aplicació mòbil ha estat completament encertada. Ha sigut la meva primera experiència en aquest àmbit i els coneixements adquirits sobre tecnologies híbrides per dispositius mòbils és una porta oberta a grans oportunitats.

Referent a les tecnologies híbrides, clarament, són eines que garanteixen un bon resultat en diferents plataformes. Segons alguns articles que he llegit, aquestes tecnologies seran emprades per a crear el 80% del mercat de les aplicacions en els marketplace d'Android i IOS.

Les avantatges són que desenvolupant un sol codi obtens el mateix resultat, o almenys molt semblant en diferents plataformes. No podem oblidar que una aplicació nativa es pot afinar fins a que el seu rendiment sigui òptim amb un temps de resposta increïblement curt i per aquest motiu la majoria, per no dir totes, les aplicacions híbrides són més aviat per gestionar, i no pas per videojocs, edició de vídeo o àudio. Les tecnologies híbrides encara no arriben a aquest nivell d'eficiència ja que operen sobre els WebView dels dispositius, però durant el temps transcorregut des de que vaig iniciar el TFC, he vist com dia a dia apareixen noves eines, plugins, frameworks, etc... que permeten interactuar amb els components físics dels mòbils d'una forma més àgil.

En els pròxims anys tindrem un enorme ventall d'eines híbrides que ens permetran crear aplicacions cada cop més elaborades i que finalment retallaran distància amb les aplicacions natives.

Per altra banda, deixant de banda la interfície de la aplicació, abordar el fet de crear el meu propi servidor afavoreix el model SOA (Arquitectura Orientada a Serveis). És evident que SOA ofereix una gran varietat de models de negoci, gràcies a oferir serveis accessibles per tothom a un cost realment baix.

Ja fa uns quants anys que SOA s'expandeix en tots els àmbits i no podem negar que és fruit de grans èxits. Infraestructures tancades, a mida, concretes per cada entorn, només són novament creades per necessitats molt específiques, que es justifiquen per obtenir una millora competitiva, que no es vol oferir com a servei i, evidentment, per seguretat.



En cada fase del projecte he pogut reafirmar la idea que aquest sistema aportarà solucions pràctiques a tots els nivells, cosa que el mercat incrementa gràcies a oferir un aplicatiu mòbil per als usuaris i una interfície REST de comunicació per qualsevol que hi vulgui connectar-s'hi.

Definitivament, estem davant un sistema que permet créixer exponencialment amb una millora constant de funcionalitats, altament escalable i que no necessita de gaire infraestructura per iniciar-se.

Recomanacions

Com se sol dir «No es oro todo lo que reluce». Tot té el seus avantatges i inconvenients. Pel que fa al aplicatiu, hem de ser conscients que que la constant evolució dels dispositius fa que la nostra aplicació necessiti evolucionar constantment. Això implica un cost constant en seguiment, manteniment i desenvolupament. És evident que absolutament tots els programes necessiten evolucionar, però concretament per les APP's aquest implica més esforç. Podríem dir que el que funciona avui, demà ja no. També haurem de limitar l'abast tecnològic, ja que no tots els dispositius podran executar la aplicació, no només per la plataforma, si no per la versió del SO, per la mida de les pantalles, pels seus components tecnològics, i finalment i possiblement el més important, que es visualitzi correctament. Per no podem obsessionar-nos en fer que la aplicació funcioni perfecte en tot el mercat de dispositius, si no que hem de limitar el nostre producte al màxim volum de tecnologies emprades.

Referent al servidor, essent un peça «invisible», probablement és la que hem de dedicar-hi més atenció, ja que és el responsable de gestionar la informació. Com tots sabem, la informació és extremadament valuosa, i la informàtica pot ser extremadament cruel vers aquest fet, ja que un bug o caiguda de sistema pot patir unes conseqüències catastròfiques. Per tant, el servidor no només ha d'oferir la gestió d'informació si no que ha de gaudir d'un bon conjunt de requeriments NO funcionals importants:

- Seguretat en les comunicacions.
- Alta disponibilitat.
- Respostes ràpides.
- Infraestructura protegida amb firewalls.
- Còpies de seguretat, redundàncies.
- Alta capacitat de resposta vers caigudes de sistema.
- Control d'accés.
- Protocols de seguretat vers atacs informàtics.
-



En conclusió, afirmem que el servidor i la infraestructura en el que està muntat necessiten de molta dedicació i inversió. La gràcia d'això és que ampliar aquests requeriments són estructurats en el temps i no cal dedicar-hi molt en un principi, si no que de mica en mica i amb possibles beneficis concurrent s'aniran resolent necessitats.

Kapp'Mon

No puc oblidar-me del meu «nen». Kapp'mon ha arribat a ser un concepte, un paradigma que té com a efectes pràctics facilitar la vida a la gent. Es poden oferir serveis individuals, cooperatius, per empreses, i en qualsevol àmbit que es desitgi.

Cal destacar que el resultat final presentat en el TFC no engloba totes les idees establertes des d'un bon principi, ja que requereix de molt més de temps i probablement de mans més expertes que les meves, però puc assegurar que he obtingut un resultat molt significatiu que té una base sòlida, coherent i que funciona.

Partint d'això la meva intenció és continuar amb el projecte ja que crec que en un futur no gaire llunyà podrà sortir al mercat amb totes les garanties d'un producte estable, i sobretot, la possibilitat de crear un negoci que vagi creixent de mica en mica.

En diverses ocasions he tingut algunes reunions amb persones que treballen per aquest sector i algunes d'elles han coincidit que el més important és que el sistema es conegui el més ràpid possible, i com a solució a aquesta premissa es proposa crear una aplicació que es centri en un àmbit, com per exemple la organització cooperativa en els videojocs. Dit d'un altra manera, Kapp'mon generalitza els recordatoris com a conjunt d'activitats a realitzar, però potser no és tant atractiu com tenir una aplicació específica per a un videojocs. El problema que veig vers això és que destrueix el principal objectiu, la centralització de totes les activitats en un sol punt. Personalment, prefereixo que la aplicació hem gestioni tot tipus d'activitats i no pas de concretes, amb la consegüent necessitat d'instal·lar múltiples aplicacions. Tot i així preval el màrqueting a les funcionalitats si es vol tenir èxit. Bàsicament cal estudiar-ho en profunditat per arribar a una presa de decisions encertada.

Per acabar les conclusions, m'agradaria comentar que he obtingut una gran satisfacció personal al crear una aplicació des de zero, que m'il·lusiona, i amb una consegüent absorció de coneixements que m'ajudaran personal i professionalment.



Bibliografia

Documentació

- <http://ionicframework.com/docs/>
- <https://cordova.apache.org/docs/en/latest/>
- <https://cordova.apache.org/docs/en/latest/guide/platforms/android/index.html>
- <https://cordova.apache.org/plugins/>
- https://wiki.postgresql.org/wiki/Main_Page
- <https://jersey.java.net/documentation/latest/user-guide.html>
- <https://jersey.java.net/documentation/latest/media.html>
- <https://github.com/rajeshwarpatlolla/TimePickerForIonicFramework>
- <http://ngcordova.com/docs/>
- <http://help.eclipse.org/luna/rtopic/org.eclipse.stardust.docs.wst/html/wst-integration/dynamic-web-proj.html>
- <https://github.com/katzer/cordova-plugin-local-notifications>
- <https://docs.genymotion.com/Content/Home.htm>

Fonts d'Informació / comunitats

- <https://forum.ionicframework.com/>
- <http://stackoverflow.com/>
- <https://www.postgresql.org/list/pgsql-es-ayuda/>
- <http://www.eclipse.org/forums/index.php?S=7f944f5e08c7c01afd50f10e902b39b3>

Guies d'usuaris / Wikis

- <https://www.nabisoft.com/tutorials/java-ee/producing-and-consuming-json-or-xml-in-java-rest-services-with-jersey-and-jackson>
- <http://www.mkyong.com/webservices/jax-rs/json-example-with-jersey-jackson/>
- <http://wiki.fasterxml.com/JacksonDownload>
- <http://coenraets.org/blog/2014/10/ionic-tutorial-and-sample-application/>
- <https://devdactic.com/simple-login-example-with-ionic-and-angularjs/>
- <https://devdactic.com/local-notifications-ionic/>
- <https://www.npmjs.com/package/datepicker-for-ionic>
- <https://www.youtube.com/watch?v=5g0iHrxOIAw>



Eines ONLINE

<https://creator.ionic.io/>

<http://codepen.io/#>

Llicències

<https://github.com/rajeshwarpatlolla/ionic-datepicker>

<https://github.com/driftyco/ionic>

<http://www.apache.org>

<http://ngcordova.com/>

<http://estuinge.galeon.com/legalidad.htm>

<https://github.com/angular/angular.js/blob/master/LICENSE>

<https://www.binpress.com/license/generator>

<http://www.oracle.com/technetwork/java/javase/terms/license/index.html>

<https://jersey.java.net/license.html>

<http://www.gnu.org/licenses/old-licenses/lgpl-2.1.html>