

Sistema de Notificacions Android per a Aules Moodle (CVNotify)

Vidal Joven Montull

Resum — Actualment, l'ús de les noves tecnologies en l'àmbit educatiu s'ha vist incrementat gràcies a tots els avantatges que ha aportat a tots els sectors de la comunitat educativa. Una d'aquestes avantatges és el fet de poder fer arribar als usuaris finals, alumnes i professors, tota la informació que desitgen amb un moviment senzill, obrir el dispositiu mòbil. Basant-nos en aquesta idea es va proposar el disseny i implementació del projecte.

L'aplicació respon a la necessitat de tenir una eina que permeti amb molta facilitat a l'usuari rebre les notificacions que es produeixen en les Aules Moodle[1] del seu curs acadèmic. Tot i això, ens vam decantar per implementar una única funcionalitat de l'aplicació, rebre les notes de les entregues realitzades en les pràctiques.

CVNotify, el nom de la nostra aplicació, és l'aplicació fruit de la realització d'aquest treball final de Grau, la qual permetrà als alumnes rebre les notificacions, que es produeixen en les retroaccions per part dels professors en la qualificació de les entregues realitzades, en els seus dispositius Android.

Gràcies a la realització d'aquest projecte hem pogut veure quin és el funcionament d'un Sistema de Notificacions Push[2] per a Android, hem pogut veure com es realitzen tasques en segon terme des de dispositius Android o quin és el funcionament intern de la base de dades de Moodle.

Paraules Clau — Moodle, Notificacions Push, Android, Tasques.

Abstract — CurrentlyThe use of new technologies in education has been increased thanks to everyone who has contributed benefits to all sectors of the educational community. One of these advantages is being able to reach end users, students and teachers, all the information you want with a simple movement, open the mobile device. Based on this idea has been carried out this project.

The application responds to the need for a tool that allows the user to very easily find the mark obtained in the delivery of work in Moodle Classroom Systems.

AppNotify, the application is the result of the completion of this final level, which will allow students to receive notifications that occur in feedback mechanisms by teachers in grade deliveries made in its Android dispositius .

The motivation of the project was to improve the original idea aplicatiu native Moodle, so that students know firsthand what their qualifications are set in the delivery done without the need to enter the Moodle classroom

Index Term — Moodle, Notificacions Push, Android, Assignments.



1. INTRODUCCIÓ

L'Objectiu del present article és presentar la solució per a la implementació d'una aplicació per a rebre les notificacions que es produeixen en les Aules Moodle[1][3] dels alumnes com a resposta a les necessitats que aquets han mostrar en l'estudi realitzat.

En l'actualitat podem trobar al mercat diverses varietats d'aplicacions que faciliten la tasca als alumnes, en podem trobar algun exemple en l'Apartat 2 del document, però cap ofereix una solució exacte pel problema que s'ha plantejat com a base principal, oferir a l'alumne una notificació personalitzada, molt més útil que les actuals.

Per tant des de la anàlisi del problema, les necessitats i altres factors es presentarà una solució aplicada en un cas concret i específic.

La motivació d'aquest projecte era crear una aplicació alternativa a l'actual aplicació que hi ha sobre l'antic Campus Virtual i sobre l'actual aplicació nativa de Moodle, ja que l'any vinent l'antic Campus serà obsolet i es realitzarà una migració a les Aules Moodle.

En un principi es va optar en una aplicació basada en l'actual Campus Virtual. Després de reunir-nos amb els responsables del Campus Virtual, ens van assessorar que treballéssim sobre les Aules Moodle, ja que el pròxim any es realitzarà una migració cap aquest entorn.

Seguidament es va mantenir contacte amb els responsables de l'aplicació nativa de Moodle. Aquets ens van mostrar la seva negativa a donar-nos suport en la realització del projecte, al legant que en el mercat ja es pot trobar la seva aplicació que realitza la funcionalitat que volíem reproduir.

Arribats a aquest punt es va donar una volta a la idea principal, i es va decidir centrar-nos en la recepció de notificacions més detallades per part de l'alumne.

- E-mail contacte: vijom14@gmail.com
- Menció: Enginyeria Software
- Treball tutoritzat per: Daniel Ponsa (dEIC)
- Curs 2015/16

1.1. Objectius

L'objectiu principal del treball va ser poder crear un sistema propi de Notificacions Push[2] per a poder rebre totes les notificacions que es produeixen en les Aules Moodle dels alumnes.

Després d'una sèrie de reunions amb la gent responsable del Campus Virtual i de la pròpia aplicació nativa de Moodle, tal com s'ha explicat anteriorment, es va decidir acotar l'objectiu principal del projecte. Es va acordar juntament amb el tutor realitzar les notificacions de les notes publicades en les entregues de les pràctiques dels alumnes.

Dintre de l'objectiu principal hi havia una sèrie de subobjectius relacionats per tal de dur a terme amb èxit el projecte. Aquets subobjectius han sigut:

- Estudi Sistemas Moodle: Durant aquest subobjectiu vàrem realitzar la instal·lació de l'aula Moodle en el nostre servidor. A continuació es va procedir a realitzar un estudi sobre la Base de Dades, MySQL[4], i sobre el funcionament de Moodle.
- Estudi Sistema Notificacions Push Android: Es va dur a terme un estudi sobre com es realitza l'enviament i recepció de notificacions Push en dispositius Android.
- Implementació Entorn Client: Durant aquesta fase es va realitzar la implementació de l'entorn client del nostre projecte. Consisteix bàsicament en la programació de l'aplicació mòbil de l'usuari, la qual serà l'encarregada de realitzar el login contra la Base de Dades de Moodle i registrar el dispositiu en el servidor de Google Cloud Message (GCM)[5], quedant a l'espera de rebre les notificacions.
- Implementació Entorn Servidor: Aquest subobjectiu ha sigut de llarg el que més problemes ha portat. Primerament per la presa de decisió sobre quina solució seria més adient per poder notificar al client, i a continuació pel desconeixement de la creació de tasques Cron[6] en JAVA. Es va dur a terme la programació del servidor, encarregat d'obtenir la informació introduïda pel professor i notificar-ho a l'entorn Client.

En l'Apartat 4 del projecte es pot trobar un explicació més detallada de les fases que s'han dut a terme durant el projecte.

Aquest article està estructurat de la següent manera. A la Secció 2 es presenten les experiències i aplicacions prèvies que s'han considerat més rellevants en els sistemes existents, les quals s'han tingut en

compte per desenvolupar el nou sistema de notificacions. La metodologia seguida pel desenvolupament de l'aplicació s'exposa a la Secció 3. La Secció 4 recull els resultats i els elements generats al llarg del desenvolupament del projecte. Finalment, la Secció 5 presenta les conclusions i futures línies de treball.

2. ESTAT DE L'ART

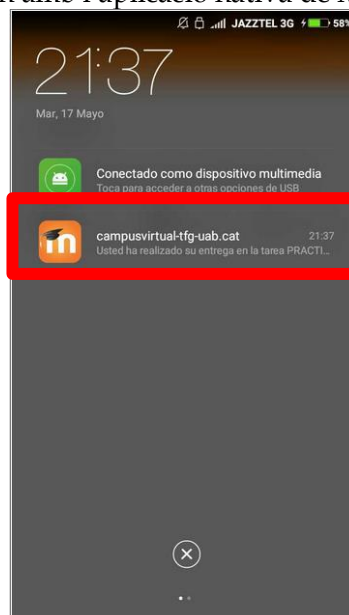
En l'actualitat les aplicacions mòbil s'estan convertint en un model de negoci molt extens. Això provoca que existeixin una gran varietat d'aplicacions en el Google Play Store, per a dispositius Android, dedicades a Aules Moodle.

Després de realitzar proves de diferents aplicacions que trobem al Google Play (MyCampus Moodle, Moodle Mobile i MDroid), s'ha pogut observar que, entre d'altres, ofereixen les següents característiques:

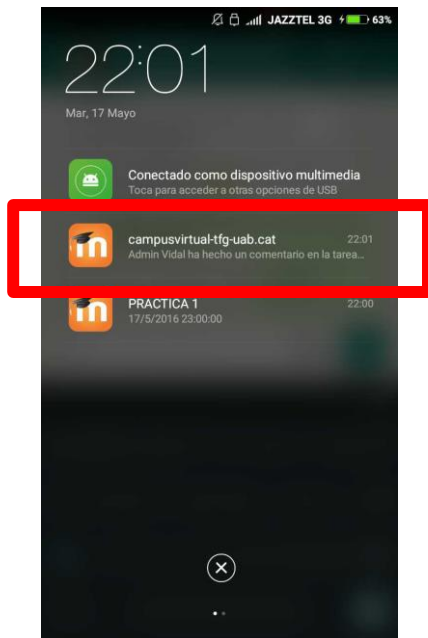
- Navegar pel contingut dels cursos de l'alumne
- Rebre notificacions instantànies de missatges i altres events.
- Buscar de manera ràpida persones i contactar amb elles.
- Veure les qualificacions dels cursos.

En el cas de les notificacions s'han comprovat quines són aquestes notificacions i que notifiquen en tot moment.

A continuació es mostren una sèrie de notificacions que es reben amb l'aplicació nativa de Moodle.



Imatge 1: Notificació que rep l'alumne quan ha realitzat l'entrega d'una pràctica.



Imatge 2: Notificació que rep l'alumne quan el professor dona retroacció a l'entrega d'una pràctica.

2.1. Elicitació de requeriments.

En aquest projecte s'ha desenvolupat una aplicació des de zero i no s'havia especificat cap restricció prèvia-ment. Per tant, observant i analitzant els sistemes existents es va elaborar un primer llistat amb els requisits necessaris per poder definir un sistema que millorés als que ja existien. A partir de reunions setmanals amb el tutor, el qual és un dels Stakeholders principals, es valoraven els requeriments proposats i es triaven els més prioritaris per ajustar la complexitat de l'aplicació amb el temps de desenvolupament disponible.

Simultàniament es va realitzar el sondeig de diferents Stakeholders, en aquest cas alumnes, usuaris de l'aplicació mòbil. Es van obtenir els requeriments necessàries per poder dur a terme el projecte i que pogués satisfer als usuaris. Aquets resultats es poden veure en el següent link:

https://github.com/vidiJM/CVNotificaciones/blob/master/doc/Informe_Brainstorming_v.1.0.docx

Entre els resultat més importants que vam aconseguir estan:

- Rebre notificacions amb les notes de les pràctiques.
- Notificació una setmana abans de l'entrega de la pràctica

En un principi vàrem decidir conjuntament amb el tutor centrar-nos en les notificacions que rebrien els alumnes i els professors. Per qüestions de temps i desconeixement de l'entorn Moodle es va decidir retallar el projecte i centrar-nos bàsicament en les notificacions que rebrien els alumnes quan es realitzes la publicació de les qualificacions per part del professor sobre les tasques que s'haguessin entregat.

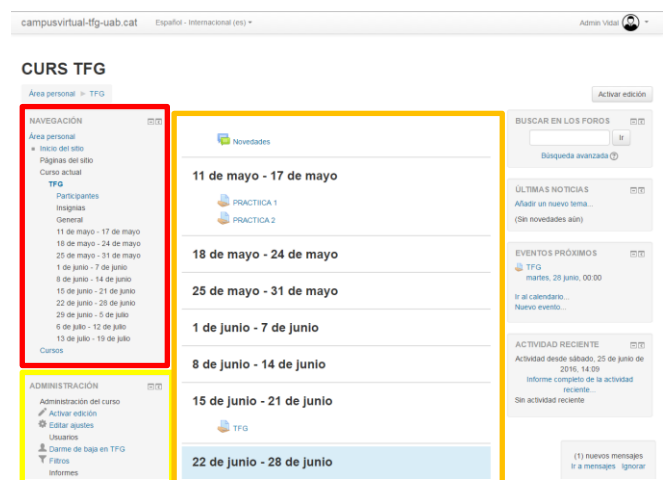
2.2. Estudi del sistema

Per poder dur a terme aquest projecte es va realitzar un estudi en viu sobre el funcionament i en que consisteixen el sistemes d'Aules Moodle.

Primerament es va realitzar la instal·lació en el servidor de l'aplicatiu Moodle 2.7. Per a la realització de la instal·lació es va haver de configurar la Base de Dades (BBDD), es va crear una nova BBDD de nom moodle, juntament amb la creació d'un usuari amb permisos sobre la BBDD.

A continuació es va crear un directori a l'arrel del servidor, de nom moodle, on pujarem la carpeta descomprimida de l'aplicatiu. Finalment es va realitzar la instal·lació del lloc Moodle seguint els passos que s'indiquen després d'accedir a la URL del nostre servidor.

Un cop realitzada la instal·lació ja podem accedir amb les dades d'administrador que s'hauran introduït durant el procés d'instal·lació. A continuació ens apareixerà una pagina semblant a la que es mostrà en la Imatge 3.

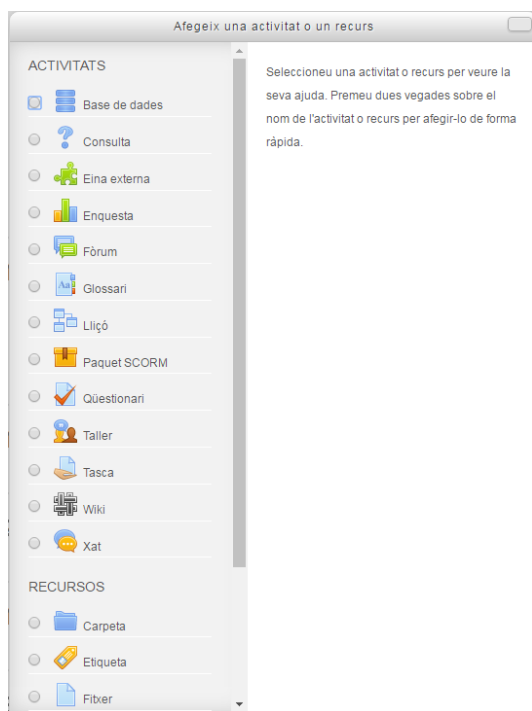


Imatge 3: Pàgina d'inici del nostre sistema d'Aules Moodle.

En aquest pagina podem trobar diferents seccions, senyalades en color en la Imatge 3:

- Color vermell: Bloc de navegació, podem trobar tots els cursos en els que estem donat d'alta, es pot accedir al perfil i àrea personal de l'usuari.
- Color groc: Bloc d'Ajust, podem trobar tot el que fa referencia a l'administració del curs, configuració del perfil.
- Color Taronja: Diagrama de Temes, normalment dividit en Seccions o Temes, on es guarda el contingut dels cursos.

Des de la secció Diagrama de Temes l'administrador de l'Aula, el professor, pot realitzar la generació d'una nova Tasca, Entrega, Documentació, Enquesta, entre d'altres activitats. En la Imatge 4 es pot observar les diferents activitats que es poden crear.



Imatge 4: Activitats que pot crear un professor.

En el nostre projecte en centrarem en les activitats anomenades Tasca, ja que en el nostre cas equival a les entregues de les pràctiques realitzades als laboratoris. Un cop la Tasca ha estat creada, apareixerà la nova activitat en el Diagrama de Temes, tant del professor com de l'alumne.

A continuació vam realitzar la creació d'un nou

usuari amb rol d'Alumne. Un cop l'alumne hagi realitzat l'entrega, el professor li donarà una retroacció, la qual serà notificada pel nostre sistema de notificacions.

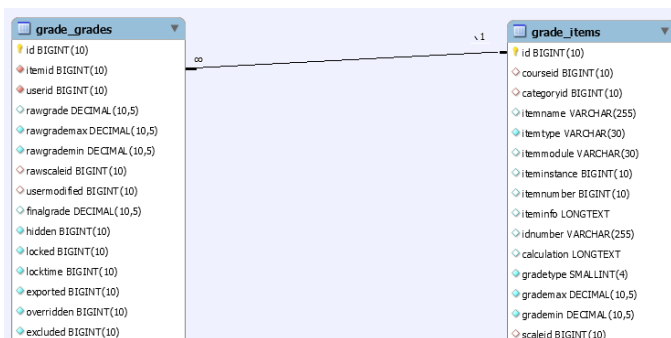
2.3. Estudi Base de Dades Moodle.

L'estudi de la base de dades és un element imprescindible per entendre el comportament de les Aules Moodle, ja que sense aquest estudi seria molt difícil saber d'on podem obtenir la informació necessària que mostrarem als usuaris. Ens aprofitarem que la base de dades es relacional per poder obtenir tots els valors necessaris.

Per a la realització del projecte ens centrarem en les taules de la base de dades mdl_grade_grades, mdl_grade_items, i mdl_user. Gràcies a la relació que hi ha entre elles podem obtenir la informació necessària per a la creació de la notificació que s'enviarà a l'alumne. A continuació realitzarem una explicació de la informació que podem obtenir de les taules:

- mdl_grade_grades: en aquesta taula podrem obtenir quina serà la nota de la pràctica avaluada pel professor (rawgrade). També podem obtenir el identificador d'usuari (userid), ens serà útil per a relacionar aquesta nota amb l'usuari de la taula mdl_user, i l'identificador de l'entrada a la taula (item_id), el qual utilitzarem però obtenir el títol de la pràctica assignat pel professor.
- mdl_grade_items: a partir d'aquesta taula i gràcies a la relació entre el camp id i el camp item_id de mdl_grade_grades, podem obtenir el nom de la practica que ha sigut avaluada, gràcies al camp itemname, el qual es mostrarà en la notificació que rebrà l'alumne.

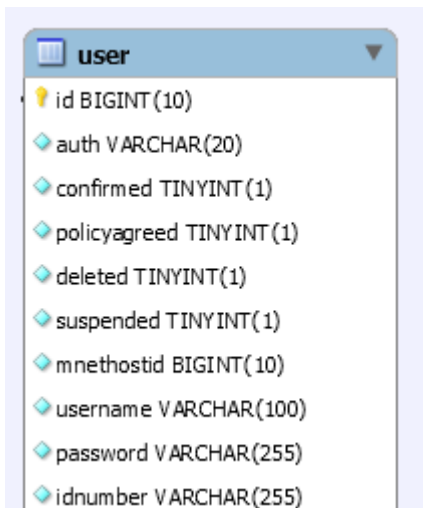
En la Imatge 5 es pot veure una imatge amb la relació entre les dues taules.



Imatge 5: Relació entre les taules grade_grades i grade_items

- `mdl_user`: serà la taula a partir de la qual podrem obtenir el id del usuari que ha realitzat la tasca avaluada. Sobre aquesta taula afegirem una nova columna, la qual serà l'encarregada de rebre el Device Token de l'usuari, mitjançant el qual podrem identificar a quin dispositiu enviar la notificació.

En la Imatge 6 es pot veure una imatge amb la relació entre les dues taules.



Imatge 6: Taula `mdl_user`

2.4. Estudi Notificacions Push

Un cop vam aprofundir amb el sistema Moodle, vàrem realitzar l'estudi i funcionament de les Notificacions Push sobre dispositius Android.

Una Notificació Push són tots aquells missatges que rebem en els nostres dispositius y han sigut enviats des d'un altre punt d'un sistema, tant si la nostra aplicació està sent utilitzada per l'usuari como si està en segon pla, com si encara no s'ha encès l'aplicació o el dispositiu es troba en repòs.

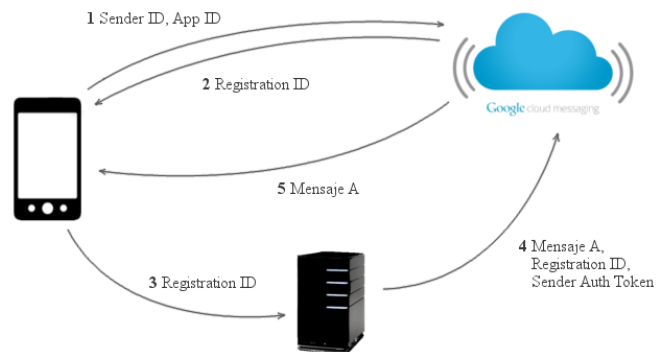
Les notificacions en Android funcionen mitjançant Google Cloud Message, gracies a la intervenció de tres escenaris diferents.

- Google Cloud Message: Servei habilitat per Google per a l'enviament de Notificacions Push.
- Servidor: Màquina amb un servei (REST, SOAP, Aplicació Web..) encarregat de gestio-

nar els identificadors de registre de dispositius als que podem enviar les notificacions i de comunicar-se amb GCM sol·licitant l'enviament de notificacions als dispositius que es desitja.

- Dispositiu Android: Dispositiu que rebirà les notificacions.

En la Imatge 7 es detalla el Diagrama de Seqüència de les Notificacions Push en dispositius Android.



Imatge 7: Diagrama de Seqüència per a l'enviament de Notificacions Push

En el primer pas veiem com el dispositiu envia a GCM, un Sender ID i un App ID. Aquets dos valors s'aconsegueixen gràcies a la pàgina de Google API, a partir de la creació d'un nou projecte. El dispositiu mòbil de l'usuari s'encarregarà d'enviar les dades a GCM per tal de rebre un Registration ID, únic per a cada dispositiu per aquella aplicació. Això es fa gracies a un Intent de registre que llança l'aplicació.

`com.google.android.c2dm.intent.REGISTER`

Si el registre ha estat correctament el Servidor GCM envia un Intent per tornar-nos el Registration ID.

`com.google.android.c2dm.intent.REGISTRATION`

A continuació, el dispositiu envia al nostre servidor el Registration ID per tal de guardar-lo. En el nostre a part, enviem el nom d'usuari el qual farem servir per relacionar-lo amb la pràctica avaluada.

Ara el nostre servidor ja te totes les dades necessàries per enviar-les al Servidor GCM. Nosaltres realitzarem l'enviament de la notificació quan es produeix una modificació en la nota de la pràctica. En aquest cas s'enviarà el missatge que es vol enviar a l'usuari juntament amb el Registration ID i el Sender

Auth Token.

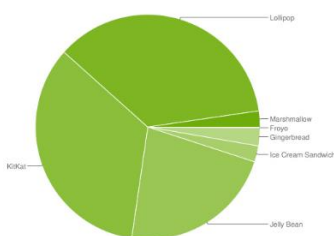
Finalment el Servidor GCM posa en cua el missatge a enviar fins que el dispositiu estigui disponible, un cop disponible l'usuari rebrà el missatge en el seu dispositiu.

En l'Apèndix A2 es pot trobar una petita definició de tots els paràmetres que es fan servir durant l'enviament de Notificacions Push.

2.5. Estudi Mercat Android

El nostre projecte ha estat dissenyat per a ser compatible amb dispositius amb Android 4.0.3 (versió del SDK +15) ja que representa el 96% dels dispositius [7]. Aquestes dades són de Març del 2016.

| Version | Codename | API | Distribution |
|---------------|--------------------|-----|--------------|
| 2.2 | Froyo | 8 | 0.1% |
| 2.3.3 - 2.3.7 | Gingerbread | 10 | 2.6% |
| 4.0.3 - 4.0.4 | Ice Cream Sandwich | 15 | 2.3% |
| 4.1.x | Jelly Bean | 16 | 8.1% |
| 4.2.x | | 17 | 11.0% |
| 4.3 | | 18 | 3.2% |
| 4.4 | KitKat | 19 | 34.3% |
| 5.0 | Lollipop | 21 | 16.9% |
| 5.1 | Marshmallow | 22 | 19.2% |
| 6.0 | | 23 | 2.3% |



Imatge 8: Quota de mercat de les versions de sistemes operatius Android.

3. METODOLOGIA

Per poder aconseguir dur a terme els objectius del projecte s'ha seguit els passos marcats en el Diagrama de Gantt, Apèndix A1, per tal d'establir unes dates més significatives del projecte.

Abans de l'inici del projecte es van realitzar diferents documents bàsics per poder establir les bases del nostre projecte i conèixer tot el que s'havia de tenir en compte abans de començar amb el desenvolupament del codi.

A continuació s'expliquen els documents previs abans del desenvolupament.

3.1. Requeriments del Software

Amb aquest document volem especificar totes les funcionalitats, restriccions o la qualitat dels elements a desenvolupar per poder tenir una base sobre la que poder treballar. A continuació s'especifiquen els 3 requeriments que es van especificar i la definició de

cada una de les seccions. Aquest requeriments es poden trobar a l'Apèndix A3.

3.1.1. Requeriments Funcionals

Aquí es van especificar les funcionalitats bàsiques que hauria de tenir el nostre projecte, així com especificar tot el que ha de realitzar el nostre sistema.

3.1.2. Requeriments de Desenvolupament

Aquí es van marcar les limitacions pròpies del projecte, que és el que ens limita i quins recursos teníem disponibles per poder dur a terme el projecte.

3.2. Diagrama de Seqüència

Els diagrames de Seqüència ens permet veure la seqüència que es realitzaria quan executem varies accions en el nostre propi sistema. En el nostre cas hem realitzat els diagrames més representatius del sistema.

En aquest cas, un correspon a la connexió realitzada entre el dispositiu mòbil, el nostre servidor propi i el servidor GCM, per tal d'enviar i rebre les notificacions. L'altre diagrama correspon al login i registre del Registration ID per part del dispositiu amb els servidors GCM

En l'Apèndix A4 es poden veure els diagrames de Seqüència.

Després de realitzar els documents es va procedir a crear l'estructura dels diferents mòduls.

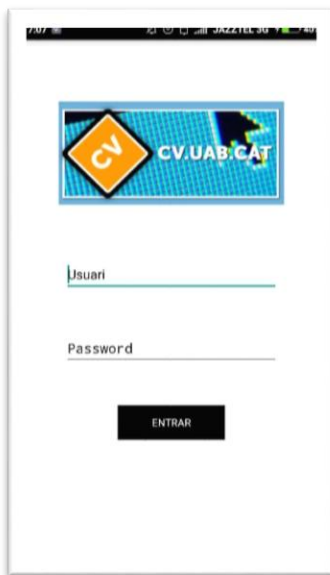
3.3. Mòdul Dispositiu Mòbil

Primerament es va portar a terme el desenvolupament de la primera pantalla, la pantalla on es realitzarà el login mitjançant la BBDD de Moodle.

En la Imatge 9 es pot veure una vista de la primera pantalla que es trobarà l'alumne al obrir l'aplicació.

Seguidament es va realitzar tota la connexió amb el Servidor GCM, la funcionalitat de la qual s'ha explicat anteriorment.

A continuació s'explicarà el les funcions principals que realitzarà el nostre dispositiu.



Imatge 9: Pàgina principal de l'aplicació

3.3.1. Registre Servidor GCM

Primerament el nostre dispositiu haurà de comprovar si està instal·lat correctament GooglePlayService. Per això ens ajudarem de la classe `GooglePlayServiceUtil` que disposa del mètode `isGooglePlayServicesAvailable()` per a realitzar la verificació. Si rebem un valor `SUCCESS`, el següent pas serà comprovar si ja tenim guardat un `RegistrationID` en el dispositiu.

Per a la comprovació utilitzarem el mètode `getRegistrationId()` que recuperarà `PROPERTY_REG_ID` de las Shared Preferences per comprovar si existeix o no.

En cas que existeixi el `RegistrationID` es realitzaran una sèrie de comprovacions per veure si es pot continuar fent servir el mateix, ja que depèn de la versió de l'aplicació, si ja ha expirat i si ha canviat el nom d'usuari.

En cas contrari, executarem una tasca asíncrona, `TareaRegistreGCM`. Aquesta tasca executa tres passos diferents:

- Registrar-nos contra el Servidor GCM
`regid = gcm.register(SENDER_ID);`
- Registrar-nos contra el nostre Servidor
`saveReg(regid, params[0]);`
- Guardar a les preferències el `RegistrationID`
`setRegistrationId(context, params[0], regid);`

3.3.2. Informar al Servidor el `RegistrationID` i nom usuari

Per informar al nostre Servidor el `RegistrationId` i nom usuari, s'utilitza el mètode `saveReg(regid,`

`params[0])` tal i com s'ha comentat abans.

Aquest mètode executa una nova tasca asíncrona encarregada d'enviar al nostre servidor el paràmetre per mètode POST. Un listener PHP s'encarrega de rebre aquestes dades i introduir-les en la BBDD Moodle. Allà es van crear dues noves taules:

- `app_tfg_pract`: encarregada de guardar la informació que es passi des de Moodle mitjançant un Trigger. En el següent punt s'explicarà amb una mica més de detall.
- `app_tfg_user`: encarregada d'emmagatzemar el nom d'usuari i el `RegistrationID` de l'usuari.

3.3.3. Rebre i processar les notificacions

Per a la tasca de recepció i processament de les notificacions primerament necessitem un `BroadcastReceiver` i un `IntentService`. Es fa d'aquesta manera ja que no convé sobrecarregar massa el Broadcast i així deleguem la feina a un Servei.

El nostre Broadcast estendrà de la classe `WakefulBroadcastReceiver` i s'implementarà l'event `onReceive()`. La crida al servei la farem mitjançant `startWakefulService(context, (intent.setComponent(comp)))`; que rebrà com a paràmetres el context actual i el propi intent rebut sobre el que indiquem el servei a executar.

Per al nostre servei creem una classe `GCMIntentService`. Primerament rebem una instància de `GooglePlayService` i obtindrem el tipus de missatge rebut, gràcies a `getMessageType(intent)` i els seus paràmetres, gràcies a `getExtras`. En el nostre cas utilitzarem un mètode auxiliar per poder visualitzar les notificacions `mostrarNotificacion()`.

3.4. Mòdul Moodle

Respecte al mòdul Moodle es va portar a terme una petita modificació en la seva BBDD. Com s'ha comentat amb anterioritat es van crear dues taules per a poder emmagatzemar tota la informació necessària.

En el cas de la taula `app_tfg_pract`, les dades que conté són un id, nom usuari, nota, nom de la pràctica i temps de modificació.

Tota aquesta informació és introduïda gràcies a l'execució d'un trigger en el moment que s'introdueixi les notes de la pràctica per part del professor.

En la Imatge 10 es pot veure aquest trigger.

```

DROP TRIGGER TRI_MOODLE;
DELIMITER //
CREATE TRIGGER TRI_MOODLE
AFTER UPDATE ON mdl_grade_grades FOR EACH ROW
BEGIN
INSERT INTO app_tfg_prac
(usuari,prac,grade,temps)
VALUES (NEW.userid,NEW.itemid, NEW.rawgrade, NEW.timemodified);
END//
DELIMITER ;

```

Imatge 10: Trigger executat en el Servidor Moodle

Sobre la taula `app_tfg_user` es guarden les dades de l'usuari que s'ha registrat amb la seva aplicació. Aquestes dades s'envien per POST al Servidor. Un cop es rep la informació necessari s'executen les Querys per introduir la informació en la BBDD

A continuació es mostra la Query que s'executa.

```

$id = mysql_query("SELECT id FROM mdl_user WHERE username='$user'");
$result = mysql_query("INSERT INTO app_tfg_user (usuari,userid,device) VALUES ('$user','$id','$idGCM')");

```

Imatge 11: Query que s'executa al Servidor Moodle

3.5. Mòdul Servidor

En la part Servidor es va procedir a crear un Web Service, de tal manera que s'encarregués d'executar una tasca CRON cada minut. Gràcies ha això podem anar comprovant que les dades introduïdes en la BBDD no han canviat. En cas de canvi es procedeix a crear el missatge JSON que s'enviarà al Servidor GCM.

En un principi es va decidir crear un plugin PHP per a Moodle. Aquesta opció es va descartar degut a la falta d'experiència en el llenguatge i el desconeixement de la creació de plugins.

Per a enviar aquest missatge es va crear una nova classe `POST2GCM`, en la qual s'executarà el mètode `post(String apiKey, String regID, String nomPract, double nota)`.

Aquest mètode s'encarrega de les següents tasques:

- Crear una connexió URL
- Muntar el cos del missatge JSON
- Enviar el missatge al Servidor GCM

4. RESULTATS

El desenvolupament del Sistema de Notificacions per Aules Moodle ha permès realitzar un projecte d'Enginyeria del Software, aplicant les activitats pròpies d'aquesta disciplina.

Els objectius principals del projecte no s'han pogut

aconseguir. En un principi es volia crear un sistema capaç de notificar totes les alertes que es produïen en les Aules Moodle. Després de les reunions amb els responsables del Campus Virtual i de l'aplicació nativa de Moodle, juntament amb el tutor, es va decidir modificar la idea inicial del projecte i vàrem passar-nos a centrar-nos en la notificació de les notes introduïdes en les entregues de les Tasques.

Això ha provocat un gran retard en la finalització de les etapes marcades en la planificació.

Al final del desenvolupament del projecte, tot no haver pogut aconseguir realitzar l'objectiu inicial, s'ha pogut dur a terme l'alternativa que ens vàrem proposar.

S'ha desenvolupat una aplicació mòbil totalment funcional capaç de registrar-se als servidors GCM i rebre notificacions, tal i com es mostra en la Imatge 12.



Imatge 12: Notificació rebuda per part de l'alumne

5. FUTUR DEL PROJECTE

Des de l'equip de projecte creiem que el projecte final és molt millorable en planificacions futures.

Una funcionalitat que es podria implementar seria donar-li la possibilitat a l'usuari final de modificar les notificacions que vol rebre. També seria de gran utilitzat que es poguessin rebre totes les notificacions

que es produeixen en les Aules Moodle.

Un ultima possibilitat serà afegir una nova funcionalitat orientada al professor. Aquesta utilitat s'encarregaria de notificar al professor quan un alumne realitza qualsevols entrega per a rebre una retroacció.

6. CONCLUSIONS

Com hem anat veient en aquest informe, s'ha aconseguit desenvolupar un servei propi de Notificacions Push. S'ha aconseguit desenvolupar un Web Service encarregat de llançar una tasca CRON cada cert temps, la qual es la clau central pel correcte funcionament del projecte.

També s'ha desenvolupat una aplicació Android encarregada de registrar-se amb els Servidors GCM. També es capaç d'obtenir el nom d'usuari i guardar les dades en el nostre servidor per a una possible utilització més endavant.

Com a reflexió personal, aquest projecte m'ha permès investigar i utilitzar diferents tecnologies innovadores, que al llarg de la meua vida no havia pogut utilitzar abans. Entre elles la tecnologia de notificacions push, la tecnologia utilitzada per a crear tasques CRON, la utilització de crides POST fent servir missatges JSON, la utilització de PHP per a rebre crides POST des de dispositius Android.

A part de l'ús de les noves tecnologies o llenguatges mencionats, aquest projecte m'ha permès profunditzar en tecnologies que havia treballat anteriorment, com per exemple l'ús de repositoris GIT, la configuració d'un servidor XAMPP, el desenvolupament de triggers MySQL.

7. AGRAÏMENTS

Per finalitzar aquest informe, en primer lloc vull agrair a Daniel Ponsa, tutor d'aquest projecte, per orientar-me en el projecte, per la seva paciència i dedicació durant els 4 mesos de desenvolupament.

També vull agrair a la meua família i amics per tot el suport que m'han donat durant tot aquest temps.

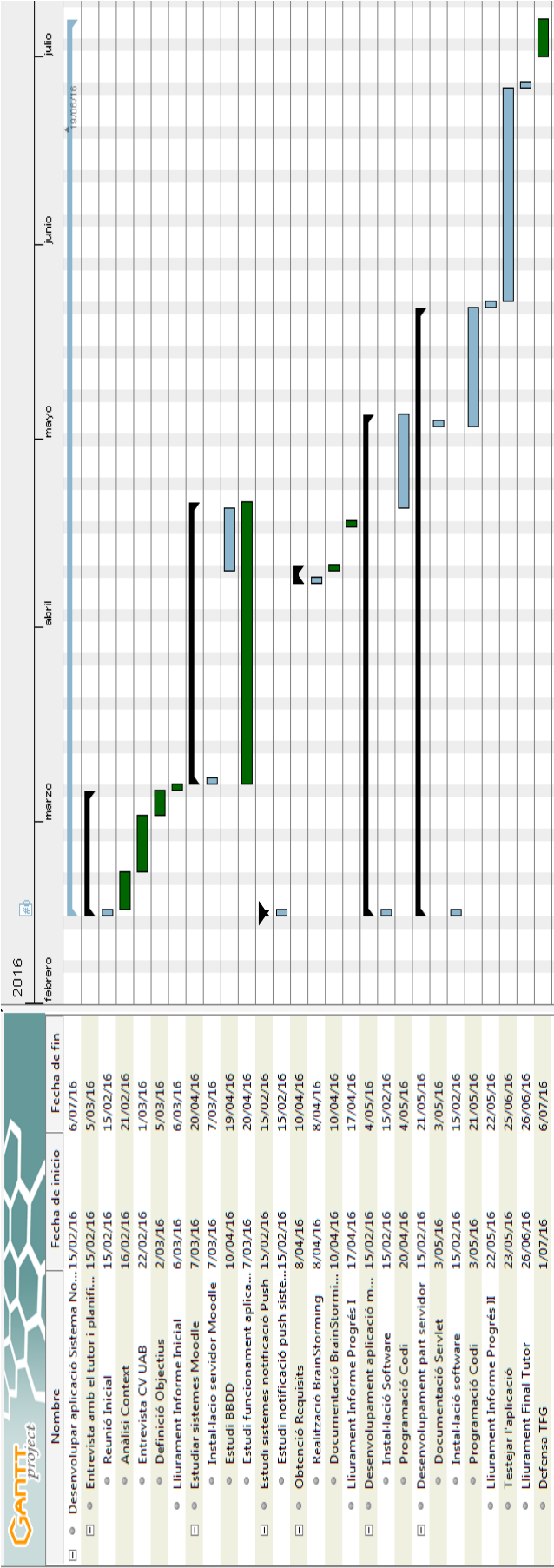
8. BIBLIOGRAFIA

- [1] Documentació Aules Moodle [Online] Available 06/2016
https://docs.moodle.org/30/en/Main_page
- [2] Documentació Notificacions Push [Online] Available 06/2016
<https://www.adictosaltrabajo.com/tutoriales/configurando-notificaciones-push-android/>
- [3] Fòrum Discussions Moodle [Online] Available 06/2016
<https://moodle.org/mod/forum/view.php?id=55>
- [4] Esquema BBDD Moodle [Online] Available 06/2016
https://docs.moodle.org/dev/Database_schema_introduction
- [5] Documentació Google Cloud Message [Online] Available 06/2016
<https://developers.google.com/cloud-messaging/>
- [6] Tasca CRON JAVA [Online] Available 06/2016
http://www.javamexico.org/blogs/jpaul/ejemplo_basico_quartz_221_tomcat_7054_con_edicion_de_expression_cron
- [7] Mercat S.O Android [Online] Available 06/2016
<http://www.xatakandroid.com/mercado/1-de-cada-10-dispositivos-android-ya-estan-actualizados-a-marshmallow>
- [8] Document Brainstorming
https://github.com/vidiJM/CVNotificaciones/blob/master/doc/Informe_Brainstorming_v.1.0.docx

9. APÈNDIX

A1. DIAGRAMA DE GANTT

Planificació realitzada al principi del projecte, com es pot observar, la primera fase de desenvolupament va ser el Mòdul Mòbil, a continuació es va passar al desenvolupament del Mòdul Servidor.



A2. DEFINICIONS

- **Sender ID:** es fa servir en el procés de registre per identificar una aplicació Android a la qual se la permet enviar missatges al dispositiu. S'obté de la pàgina de Google APIs Console.
- **Application ID:** Identificador de l'aplicació que s'està registrant per rebre missatges. Es tracta del nom del paquet de l'aplicació indicat en el fitxer *manifest.xml*.
- **Registration ID:** Identificador enviat pels Servidors GCM a l'aplicació com a resposta a una sol·licitud de registre. Aquest identificador s'ha d'enviar al nostre servidor per identificar cada dispositiu que s'ha registrat per rebre missatges. Està associat a una aplicació concreta existent en un dispositiu concret.
- **Sender Auth Token:** Clau que s'ha d'emmagatzemar dins del servidor per poder tenir accés als serveis de Google,

A3. REQUERIMENTS DEL SOFTWARE

A3.1. REQUERIMENTS FUNCIONALS

En aquest apartat s'especifiquen les funcionalitats que ha de tenir el nostre sistema. Tot el que el nostre sistema ha de ser capaç de fer.

DISPOSITIU MÒBIL

- Ha de poder obtenir el Registration ID de l'aplicació.
- Ha de ser capaç d'actualitzar el Registration ID en cas que hagi expirat.
- Ha de permetre realitzar login a l'usuari mitjançant l'usuari de la BBDD de Moodle.
- Ha de permetre guardar el nom d'usuari i el Registration ID.
- Ha de permetre rebre les notificacions dels Servidors GCM.

SERVIDOR PROPI

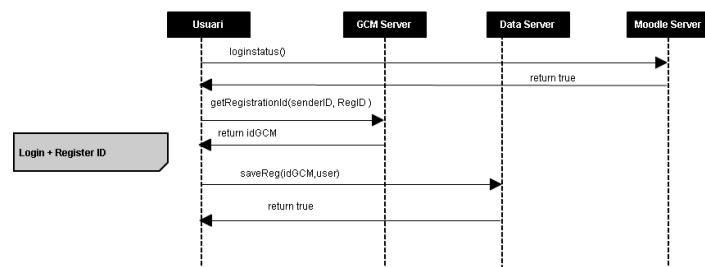
- Ha de permetre realitzar una connexió a la BBDD de Moodle per obtenir la següent informació:
 - Nom usuari.
 - Registration ID.
 - Nota de la Pràctica, ID de la Pràctica.
 - Data d'inserció.
- Ha de permetre llançar una tasca CRON cada minut.
- Ha de permetre enviar tota la informació necessària al Servidor GCM.

A3.2. RESTRICCIONS AL DESENVOLUPAMENT

- Es disposen de 300 hores de desenvolupament repartides en 4 mesos.
- La Data Límit de l'entrega és el 26 de Juny de 2016.
- Els recursos humans disponibles són:
 - Estudiant que dedicarà un mínim de 300 hores.
 - Professor encarregat de la supervisió i tutorització del projecte.
 - Els recursos tècnics són 1 Ordinador Personal amb connexió a internet, amb el següent Software: Eclipse Mars JEE, Android Studio, Filezilla, SourceTree i SublimeText. 1 Servidor Propi on s'instal·larà Moodle. 1 Servidor Apache Tomcat per llançar la tasca CRON. 1 Repositori GIT.

A4. DIAGRAMA DE SEQÜÈNCIA

Login + saveRegidGCM



SendNotification

