

## Memòria del Treball Fi de Grau

### Aplicació per Android per pujar vídeos a un servidor i compartir-los a xarxes socials

**Nom:** David Monllaó Olivé

**Consultor:** Ricard Burriel Maurel

**Assignatura:** Treball Fi de Grau - Àrea d'Enginyeria del Programari

**Llicència:** GPL 2, <http://www.gnu.org/licenses/gpl-2.0.txt>

Universitat Oberta de Catalunya (UOC) - Grau d'Enginyeria Informàtica  
Curs 2011-12 - 2n quadrimestre

# Índex

1. Introducció	-	-	-	-	4
2. Agraïments	-	-	-	-	5
3. Objectius	-	-	-	-	6
4. Anàlisi de la situació actual	-	-	-	-	8
5. Anàlisi de riscos	-	-	-	-	9
6. Metodologia	-	-	-	-	10
7. Planificació	-	-	-	-	11
8. Resum de les PACs	-	-	-	-	12
1. Primera Iteració					
2. Segona Iteració					
3. Tercera Iteració					
9. Requisits	-	-	-	-	14
1. Requisits de producte - funcionals					
1. Requisits de funcionalitat					
2. Requisits de dades					
2. Requisits de producte - no funcionals					
3. Requisits de procés					
10. Estat del art	-	-	-	-	17
1. Android					
2. REST					
3. Zend Framework					
11. Disseny	-	-	-	-	23
1. Programari utilitzat					
2. Diagrama de casos d'ús					
3. Diagrama de components					
4. Aplicació client d'Android					
5. Aplicació web					
12. Implementació	-	-	-	-	34
1. Aplicació client d'Android					
2. Aplicació web					

3. Gestió del codi					
13. Testeig	-	-	-	-	39
14. Usabilitat i interfícies d'usuari	-	-	-	-	40
15. Conclusions	-	-	-	-	44
16. Futurs desenvolupaments	-	-	-	-	46
17. Solució comercials	-	-	-	-	48
18. Glossari	-	-	-	-	49
19. Bibliografia	-	-	-	-	51

## Introducció

Actualment les xarxes socials estan canviant les costums dels usuaris d'Internet, assumint ja des de fa temps, que no son usuaris passius, sinó que son ells qui aporten continguts i donen vida a aquestes xarxes. Les xarxes socials amb més difusió estan interessades en que desenvolupadors externs puguin integrar aplicacions dins la seva xarxa social per poder seguir creixent i aportar noves funcionalitats als seus usuaris, pel que faciliten l'accés a dades i a eines pròpies als desenvolupadors externs.

Per altra banda, s'està ampliant molt el nombre i els tipus de dispositius que poden accedir a Internet; el no estar lligats a un cable de corrent i/o a un dispositiu poc manejable fa que s'ampliï el ventall de possibles usos de xarxes socials. El punt d'unió entre aquests dos mons és que estan sorgint moltes empreses que, basant-se en les característiques dels dispositius mòbils com el GPS, l'acceleròmetre o la brúixola, creen aplicacions o jocs que aporten noves formes d'interactuar amb l'usuari, i aprofitant les facilitats que ofereixen les xarxes socials, s'integren dins d'aquestes per oferir la possibilitat de compartir informació sobre l'ús que fan de l'aplicació, augmentant també així la difusió de les seves aplicacions entre els contactes dels usuaris.

Aquest projecte pretén crear una aplicació per dispositius mòbils que permeti enregistrar vídeos d'una durada de fins a 10 segons i compartir-los a les xarxes socials amb més difusió de forma gairebé immediata. El projecte també implica el desenvolupament d'una aplicació web per permetre la comunicació entre l'aplicació client i la xarxa i per habilitar els usuaris a visualitzar i enllaçar els vídeos que publiquin.

## **Agraïments**

A l'Anna, per ajudar-me tant, per desencallar-me el cap quan el tinc quadrat, per tenir paciència amb les meves absències temporals tan físiques com mentals i per estar sempre amb mi; i al Ricard per l'ajuda, el seguiment i els ànims.

També a Paquito, al Robert, a la Nur, a l'Alexandra, al Jordi, a Andreu, al Joan Vicents, al Pato Luís i a l'SREd per cedir els seus dispositius per provar el funcionament de l'eina i pel feedback que han aportat.

## Objectius

Els objectius estan basats tant en els interessos professionals com en els interessos personals, tot complint sempre amb els requisits de les aplicacions d'aquestes característiques.

- Aprendre a desenvolupar aplicacions pel SO Android

El sistema operatiu Android està augmentant cada cop més la quota del mercat de mòbils, és un sistema operatiu de codi lliure i està basat en el kernel de Linux. Personalment em resulta un tema atractiu. Un dels objectius del projecte és desenvolupar una aplicació pel sistema operatiu Android que permeti accedir a la càmera del dispositiu i enviar el vídeo gravat directament a un servidor.

- Familiaritzar-se amb el concepte de software orientat a serveis i amb el concepte de REST

La relació entre aplicacions afavoreix el reaprofitament de funcionalitats i l'especialització. Amb l'augment de possibilitats que provoca l'obertura d'APIs públiques de serveis, tenir clars els conceptes i conèixer les especificacions / implementacions més utilitzades serà útil per un enginyer de programari, tant com a consumidor d'aquests serveis com com a proveïdor, pel que un dels objectius del projecte és crear una aplicació que compleixi amb els requisits d'aplicacions REST, concretament el servidor se n'encarregarà, entre altres coses, de rebre peticions per afegir vídeos o tornar el llistat de vídeos d'un usuari.

- Analitzar les possibilitats de connexió entre aplicacions d'Android i aplicacions RESTful

La majoria d'aplicacions per mòbils en algun moment envien dades a Internet, analitzar els diversos mecanismes i autenticacions és una part important de la feina quan l'aplicació té una dependència directa amb el servidor.

- Desenvolupar una aplicació complexa amb Zend Framework

Zend Framework és el framework de PHP suportat oficialment per Zend, l'empresa darrere del llenguatge PHP i el framework amb més difusió d'aquest llenguatge. El seu roadmap va lligat al desenvolupament del propi llenguatge PHP, representant també una guia de bones pràctiques d'ús del llenguatge, per aquests motius s'ha escollit PHP i Zend Framework per desenvolupar la part de servidor.

- Aprofundir en l'ús d'OAuth 2.0

OAuth és una especificació per permetre la comunicació entre aplicacions mitjançant connexions HTTP. Les plataformes web més usades com Facebook,

Google Apps o Twitter permeten accedir a les seves APIs a través d'OAuth 2.0 actuant així com a proveïdors de serveis. OAuth s'utilitzarà per permetre l'accés des de l'aplicació mòbil cap als serveis de les xarxes socials que ens permetran compartir l'enllaç cap al vídeo proporcionat pel servidor.

- Conèixer les possibilitats que ofereixen les APIs de les xarxes socials amb més difusió

Conèixer les dades / funcionalitats que ofereixen les APIs públiques de les xarxes socials permeten saber quines possibles aplicacions se'n podrien treure. És un objectiu col·lateral als demés, dins l'àmbit del Treball Fi de Grau les xarxes socials escollides per integrar-s'hi són Facebook i Twitter

- Obtenir dos productes

Tot el procés d'enginyeria del programari que es seguirà durant el curs del Treball Fi de Grau portarà a l'obtenció de dos productes, una aplicació per dispositius mòbils amb el sistema operatiu Android i una aplicació web que actuarà com a proveïdor de serveis a l'aplicació client i com a portal web per a que els usuaris puguin veure i obtenir enllaços cap als seus vídeos

## Anàlisi de la situació actual

En el context actual els dispositius mòbils permeten crear vídeos i compartir-los a les xarxes socials, pel que no hi ha aplicacions que permetin aquests tipus d'accions, el fet diferenciador, funcionament parlant, d'aquest projecte és la durada dels vídeos, limitar a 10 segons el vídeo fa que els usos de l'eina canviïn, especialitzant-se en uns tipus de vídeos més ràpids o de moments especials, es podria fer un símil amb l'infinit número d'eines de missatgeria / blogging que existien i els missatges de 140 caràcters de Twitter.

Un altre punt en el que el projecte destaca per sobre de l'alternativa "nativa" del sistema operatiu Android juntament les aplicacions oficials de les xarxes socials és el el número de passos a seguir i les ambigüitats del camí per fer-ho. El projecte pretén facilitar un camí inequívoc per:

- Enregistrar un vídeo de fins a 10 segons
- Posar-li una petita descripció al vídeo
- Seleccionar a quines xarxes socials es vol compartir el vídeo

Tècnicament parlant el procés per penjar un vídeo no té res a veure amb el procés de l'aplicació nativa d'Android juntament amb les aplicacions per Android de les xarxes socials, aquestes aplicacions tenen la limitació de ser especialment lentes a l'hora de fer aquesta tasca; per una banda hi ha varies aplicacions implicades i s'ha de seguir un gran nombre de passos i per l'altra un vídeo ocupa una gran quantitat de bytes i costa molt temps traspasar el vídeo al servidor abans d'estar disponible, això treu l'espontaneïtat que es busca justament amb aquest Treball Fi de Grau, al ser vídeos de fins a 10 segons, el temps d'espera per pujar el vídeo al servidor no serà significatiu i serà inclús menys representatiu (o fins i tot inexistent) al estar assignant una descripció al vídeo i seleccionant les xarxes on compartir-lo mentre el vídeo es va pujant en background al servidor i el servidor li torna l'enllaç cap al vídeo, per l'altra, al ser un ús tan específic el nombre de passos es redueix fins lo indispensable.



## **Anàlisi de riscos**

Les estimacions temporals mai poden ser exactes per definició, per tal de que l'estimació sigui lo més aproximada possible s'ha tingut en compte l'experiència prèvia en projectes semblants per al del costat del servidor i per al costat del client s'ha tingut en compte que, al ser una tecnologia amb la que no s'està familiaritzat, hi haurà un temps de documentació.

Un altre punt crític és el rendiment, les connexions de xarxa disponibles als dispositius mòbils varien en funció de la ubicació de l'usuari; com s'ha explicat anteriorment, al ser un punt molt important s'incidirà en prepara el flux d'execució per adaptar-se a aquesta restricció.

## Metodologia

La metodologia que més s'adapta a la situació exposada és la metodologia iterativa i incremental. Treballant amb aquesta metodologia a cada fase hi haurà un quelcom presentable i funcional.

Al tenir tres PACs a més d'aquesta de planificació per fer el seguiment del projecte, l'etapa d'inicialització correspondrà a la segona PAC. En el cas que ens ocupa el projecte consta de dues parts, una aplicació del costat del servidor i una aplicació client pel mòbil, per poder tenir quelcom funcional des de la PAC 2, la interfície de comunicació entre ambdós aplicacions haurà d'estar feta i això requerirà un temps d'estudi d'alternatives, de disseny i d'implementació, pel que aquesta PAC serà la que més càrrega de treball representarà. Les demés PACs seran iteracions, incloent a la PAC final la verificació de la llista de control.

Aquesta metodologia també ajudarà a assegurar que es segueixen els procediments i formats correctes, i anar redefinint i corregint detalls conforme el projecte va avançant. Lligat amb el punt anterior, aquesta metodologia ens motiva a fer un anàlisi de riscos iteratiu.

## Planificació

El projecte s'ha planificat tenint en compte els coneixements previs i el temps disponible per desenvolupar-lo, partint de que l'objectiu inicial és, a més de redactar la memòria i l'estudi de les plataformes, obtenir 2 productes finals.

La planificació temporal està especificada en el diagrama de Gantt adjunt a la memòria (/Gantt.html)

## Resum de les PACs

Respectant la metodologia triada s'ha dividit el desenvolupament de l'aplicació en 3 iteracions.

### ***1a Iteració***

A la primera iteració s'ha desenvolupat l'esquelet de les aplicacions i s'ha establert la comunicació entre elles, funcionalment aporta ben poc, però correspon a la fase d'inicialització pròpia de la metodologia iterativa i incremental, on es pretén crear un prototip on començar a afegir funcionalitats iteració a iteració.

Dels objectius descrits a l'apartat d'objectius en aquesta fase s'ha treballat:

- Aprendre a desenvolupar aplicacions pel SO Android
- Familiaritzar-se amb el concepte de software orientat a serveis i amb el concepte REST
- Analitzar les possibilitats de connexió entre aplicacions d'Android i aplicacions RESTful
- Desenvolupar una aplicació complexa amb Zend Framework

### ***2a Iteració***

Corresponent a la PAC 3, la segona iteració ha tractat l'enregistrament de vídeo. S'ha aprofundit en l'ús d'interfícies gràfiques d'usuari del sistema Android, que a la primera PAC ha estat anecdòtic i han sortit a la llum els problemes d'Android relacionats amb el gran nombre de dispositius als que cal donar servei.

Dels objectius descrits a l'apartat d'objectius en aquesta fase s'ha treballat:

- Aprendre a desenvolupar aplicacions pel SO Android
- Desenvolupar una aplicació complexa amb Zend Framework
- Obtenir dos productes

### ***3a Iteració***

A la tercera i última iteració, corresponent a la 4a PAC s'ha treballat la integració de

l'aplicació amb les dos xarxes socials amb més difusió actualment, Facebook i Twitter, ambdós utilitzen OAuth 2.0 per establir els permisos necessaris entre aplicacions però implementades de formes molt diferents.

Dels objectius descrits a l'apartat d'objectius en aquesta fase s'ha treballat:

- Aprendre a desenvolupar aplicacions pel SO Android
- Desenvolupar una aplicació complexa amb Zend Framework
- Aprofundir en l'ús d'OAuth 2.0
- Conèixer les possibilitats que ofereixen les APIs de les xarxes socials amb més difusió

## Requisits

El requisit principal que han de cobrir les dues aplicacions que es desenvolupen com a part d'aquest Treball Fi de Grau és permetre enregistrar vídeos des de dispositius amb sistema operatiu Android i compartir-los directament a les xarxes socials amb més difusió actualment, Facebook i Twitter.

### ***Requisits de producte - Funcionals***

Els **requisits de funcionalitat** son:

- Aplicació client d'Android
  - Que pugui generar i recuperar usuaris a/de l'aplicació web
  - Poder enregistrar vídeos des d'una aplicació d'Android
  - Poder visualitzar els vídeos enregistrats
  - Poder tornar a gravar un vídeo en cas que l'enregistrament no sigui del gust de l'usuari
  - Compartir un vídeo a Facebook i a Twitter
  - Afegir una descripció al vídeo compartit
  - Poder penjar un dels vídeos enregistrats a una aplicació web
  
- Aplicació web
  - Crear usuaris
  - Desar vídeos des de l'aplicació client d'Android
  - Actualitzar vídeos
  - Visualitzar vídeos pujats anteriorment al servidor

Els **requisits de dades** d'aquest projecte son:

- Aplicació client d'Android
  - Credencials d'accés a l'aplicació web

- Per no demanar-lo cada cop que l'usuari inicia l'aplicació
- Compte principal de com.google utilitzat per gestionar el sistema operatiu Android
  - S'extraurà de forma transparent per l'usuari
- La clau d'accés de l'usuari
  - Juntament amb el compte de Google seran les credencials d'accés a l'aplicació servidor
- El token de l'usuari per compartir a Facebook
  - Es desarà a l'aplicació client d'Android el primer cop que l'usuari autoritzi l'aplicació a publicar continguts a Facebook
  - Si l'usuari no disposa de Facebook al dispositiu mòbil aquest requisit de dades queda eliminat
- El token d'usuari per compartir a Twitter
  - Es desarà a l'aplicació client d'Android el primer cop que l'usuari autoritzi l'aplicació a publicar continguts a Twitter
  - Si l'usuari no selecciona l'opció de compartir a Twitter o no té un compte de Twitter aquest requisit queda eliminat
- Aplicació web
  - Tenir constància de tots els usuaris de l'aplicació
  - Saber quan s'ha creat cada usuari de l'aplicació
  - Tenir un registre de l'últim accés de cada usuari
  - Disposar de la relació de vídeos pujats per un usuari amb una descripció

### ***Requisits de producte - No funcionals***

- El client d'Android ha de requerir el mínim número d'interaccions de l'usuari per poder finalitzar la connexió amb l'aplicació web
- L'aplicació ha de suportar múltiples idiomes, però en principi estarà disponible en Català, Castellà i Anglès

- L'aplicació web es comunicarà amb el client d'Android seguint l'estil arquitectònic REST
- L'aplicació ha de funcionar amb dispositius Android versió 2.3.3 o superiors, ja que actualment (Juny del 2012) és on es concentra el major nombre de dispositius en actiu.

### ***Requisits de procés***

Les restriccions al procés son temporals i humanes. La data límit de presentació és el 6 de juny de 2012. La normativa de la UOC imposa l'altra restricció, l'únic element humà que participarà en el disseny, desenvolupament i testeig del projecte serà el propi estudiant.



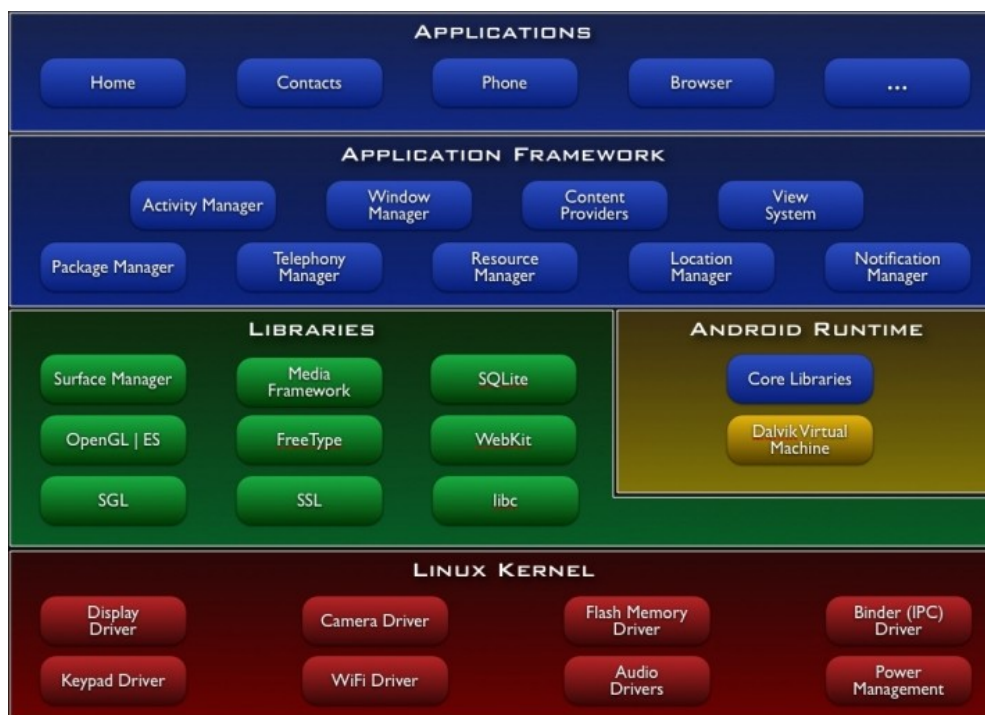
## Estat de l'art

Sent aquest un Treball Fi de Grau de l'àrea d'Enginyeria del Programari, des d'aquesta secció s'intentarà aportar una mica d'informació sobre l'estat actual de les tecnologies utilitzades en aquest projecte des d'un punt de vista tècnic. Part d'aquesta informació s'ha obtingut durant la documentació i altres a mesura que s'ha anat desenvolupant el projecte degut a les necessitats que han anat sorgint.

## Android

És, sinó el que més, el sistema operatiu per dispositius mòbils amb més difusió; la seva arquitectura està basada en el kernel de linux i, encara que les noves branques en desenvolupament estan ocultes, està totalment open sourced. Actualment (Abril del 2012) la última versió és la 4.0.

## Arquitectura



Des del punt de vista arquitectònic el sistema operatiu està organitzat en diverses capes com es pot veure a l'imatge adjunta, l'àmbit d'actuació d'aquest projecte és la capa d'aplicacions, el framework d'aplicació i les llibreries.

## Components d'aplicació

Pel tema que ens ocupa, a l'hora de dissenyar aplicacions és important tenir clar els diferents tipus de components d'aplicació que té Android, aquests son:

### Activities

Son les diferents accions (o pantalles) que es van executant mentre seguim el flux d'execució de l'aplicació

### Services

Utilitzats per mantenir processos en execució en background, degut a la gestió dels recursos que fa Android és habitual que les activitats puguin quedar pausades o es tanquin per estalviar recursos del sistema, els serveis queden protegits d'aquests inconvenients.

### Content providers

Son gestors de dades, permeten accedir a dades persistents del sistema

### Broadcast receivers

Es poden veure com a "listeners" ja que responen a events que es generen al sistema, sigui per la pròpia aplicació o per altres.

## Cicle de vida d'una activitat

Com s'ha comentat anteriorment a "Components d'aplicació" el gestor del processador d'Android canvia l'estat de les aplicacions segons convé per estalviar recursos, a més d'això, el propi sistema operatiu marca una serie d'estats per on passen les activitats; les activitats, per la seva banda, poden implementar mètodes per executar les accions que interessin segons l'estat al que es passa.

Començant per la creació de l'activitat i acabant amb la seva finalització aquests son els mètodes que permeten intercedir durant el cicle de vida.

**onCreate:** Al crear-se l'activitat.

**onStart:** Al iniciar-se la seva visualització, no confondre amb el punt següent.

**onResume:** Quan l'activitat està en primer pla i l'usuari i hi està interactuant.

**onPause:** Quan l'activitat deixa d'estar en primer pla.

**onStop:** Quan l'activitat deixa d'estar visible per l'usuari (una altra activitat està onResume)

**onDestroy:** Al tancar l'activitat.

## Permisos

Un altre aspecte important a destacar pel que fa al desenvolupament d'aplicacions és el tema de permisos; Android gestiona l'accés al hardware i a la informació del sistema a partir d'una serie de permisos, les aplicacions han d'especificar quins permisos necessiten a l'arxiu principal de configuració. Al instal·lar una aplicació els usuaris veuen quins permisos requereix l'aplicació, pel que poden decidir no instal·lar una aplicació en cas que creguin que una aplicació pretén accedir a més informació de la que hauria.

## REST

El concepte REST actualment s'utilitza per referir-se a qualsevol intercanvi d'informació entre aplicacions que funcioni únicament amb peticions HTTP sense cap restricció de format, però en el sentit més estricte REST és un principi arquitectònic basat en la simplicitat del protocol HTTP, on no es manté l'estat entre peticions, cada URI representa un recurs i hi ha diversos mètodes per interactuar amb un recurs.

Els mètodes d'HTTP son POST, GET, PUT i DELETE, junt amb el concepte de recurs podem dir que aquests mètodes ens permeten fer CRUD de les entitats amb les que treballem. Amb això cada mètode d'HTTP representa una acció CRUD:

- El mètode POST per Create
- El mètode GET per Retrieve
- El mètode PUT per Update
- El mètode DELETE per Delete

L'arquitectura també fa ús de les capçaleres HTTP, principalment els codis de resposta d'HTTPS per indicar l'estat del recurs un cop executada l'acció:

- El mètode POST
  - En cas d'èxit el codi HTTP de resposta 201 (Created)
  - En cas d'error afegeix el codi HTTP de resposta 400 (Bad request) o una altra 4XX més específica segons l'error

- Afegeix la capçalera "Location" apuntant cap al recurs creat
- Com a contingut torna el recurs creat
- El mètode GET
  - En cas d'èxit el codi HTTP de resposta 200 (OK)
  - En cas d'error afegeix el codi HTTP de resposta 400 (Bad request) o una altra 4XX més específica segons l'error
  - Com a contingut torna el recurs sol·licitat
- El mètode PUT
  - En cas d'èxit el codi HTTP de resposta 200 (OK)
  - En cas d'error afegeix el codi HTTP de resposta 400 (Bad request) o una altra 4XX més específica segons l'error
  - Com a contingut torna el recurs actualitzat
- El mètode DELETE
  - En cas d'èxit el codi HTTP de resposta 204 (No content)
  - En cas d'error afegeix el codi HTTP de resposta 400 (Bad request) o una altra 4XX més específica segons l'error
  - No torna contingut

## Zend Framework

Zend Framework és el framework de desenvolupament per PHP suportat directament per Zend Technologies, la rama comercial del llenguatge PHP.

Actualment la seva darrera versió estable és la 1.11.11, la versió 2.x està en fase beta (Actualment 2.0.0beta3)

Sent un framework que surt de la rama comercial de la pròpia empresa que desenvolupa el llenguatge PHP, el seu codi sempre serveix de guia de bones pràctiques de programació.

## MVC

Proporciona un esquelet model - vista - controlador per començar a treballar i, junt amb el generador de codi que porta incorporat, permet afegir els components més típics (controladors, models, vistes, taules de la base de dades...) via línia de comandes.

L'estructura de directoris també separa el codi de l'aplicació de la part pública per no deixar-la accessible via URL. Aquesta és l'estructura de l'esquelet:

```
nom_aplicacio
|_ application
  |_ controllers
    |_ nomcontrolador
  |_ models
    |_ nommodel
  |_ views
    |_ layouts
    |_ scripts
      |_ index
      |_ nomcontrolador
      |_ nomaccio.html
  |_ library
    |_ Zend
  |_ public
    |_ index.php
```

Una de les bones pràctiques és afegir les inicialitzacions de components extres i de helpers a l'arxiu Bootstrap.php, deixant així sense tocar la implementació de la llibreria.

## Components

A més de proporcionar l'estructura bàsica d'un MVC incorpora una serie de components per facilitar el desenvolupament amb PHP, aquesta llibreria es pot utilitzar de forma independent al MVC, així qualsevol projecte pot incorporar els components que calgui.

Alguns dels components més significatius son:

- Zend\_Form: Per gestionar formularis
- Zend\_Layout, Zend\_View i adjunts: Per gestionar les vistes
- Zend\_CodeGenerator: L'eina de generació de codi
- Zend\_Reflection: Per obtenir dades de classes, mètodes i funcions en temps d'execució
- Zend\_Db, Zend\_Db\_Adapter, Zend\_Db\_Table i altres: Per gestió de connexions de bases de dades i tractament de resultats

- Zend\_Date, Zend\_Locale, Zend\_Translate i altres: Per internacionalització i localització
- Zend\_Auth i Zend\_Session: Per autenticacions d'usuaris
- Zend\_Acl: Pel control d'accés a funcionalitats
- Zend\_Rest\_Client, Zend\_Json\_Server, Zend\_Rest\_Server i molts altres: Per aplicacions que requereixen intercanvi de dades amb altres aplicacions
- Zend\_Pdf, Zend\_Json i altres: Per formats de sortida

## Disseny

El disseny de l'aplicació ha passat per diverses fases, com passa quan es treballa amb metodologies iteratives i incrementals, a la primera fase va caldre establir la comunicació entre aplicació client i aplicació servidor, a la segona es va treballar l'enregistrament de vídeo a dispositius de tipus Android, i a la tercera fase es va treballar amb les APIs de diverses xarxes socials per integrar-les dins de l'aplicació.

El flux d'execució de l'aplicació és el següent:

1. Validació de l'usuari del client Android a l'aplicació web
  1. Si no existeix l'usuari se li demana una clau d'accés i es crea l'usuari a l'aplicació web
2. Càrrega de la interfície d'usuari per part del sistema
  1. Vista prèvia de la càmera
  2. Botons d'accions
  3. Comptador de segons
3. Enregistrament de la gravació per part de l'usuari
4. Visualització de la gravació per part de l'usuari
  1. Amb la possibilitat de tornar a gravar si el resultat no convenç a l'usuari
5. Descripció de l'enregistrament per part de l'usuari
  1. En background es va enviant el vídeo a l'aplicació web
6. Càrrega de l'enllaç cap a l'arxiu del servidor per part del sistema un cop s'ha pujat el vídeo del servidor.

### ***Programari utilitzat***

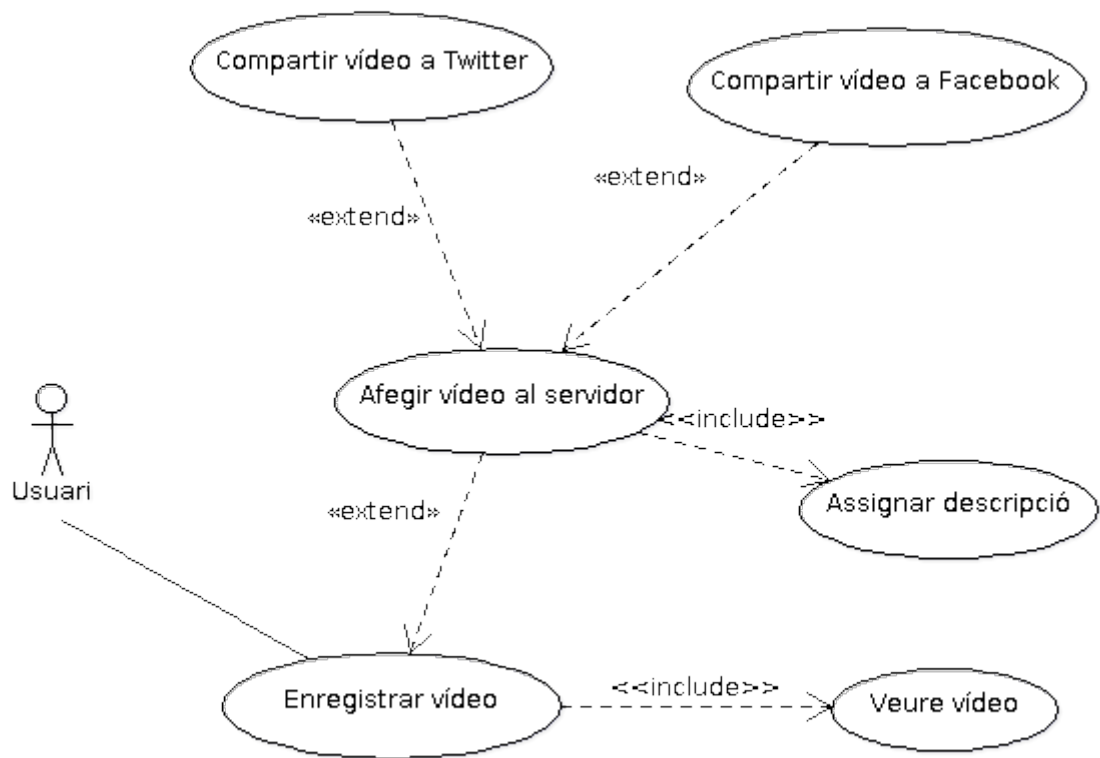
Per facilitar el disseny ens ajudem d'una serie d'eines per especificar diagrames UML i fer prototips d'interfícies gràfiques d'usuari.

S'ha utilitzat ArgoUML pels diagrames UML i la versió de proves de Balsamiq Mockups pels prototips d'interfícies gràfiques d'usuari.

### ***Diagrama de casos d'ús***

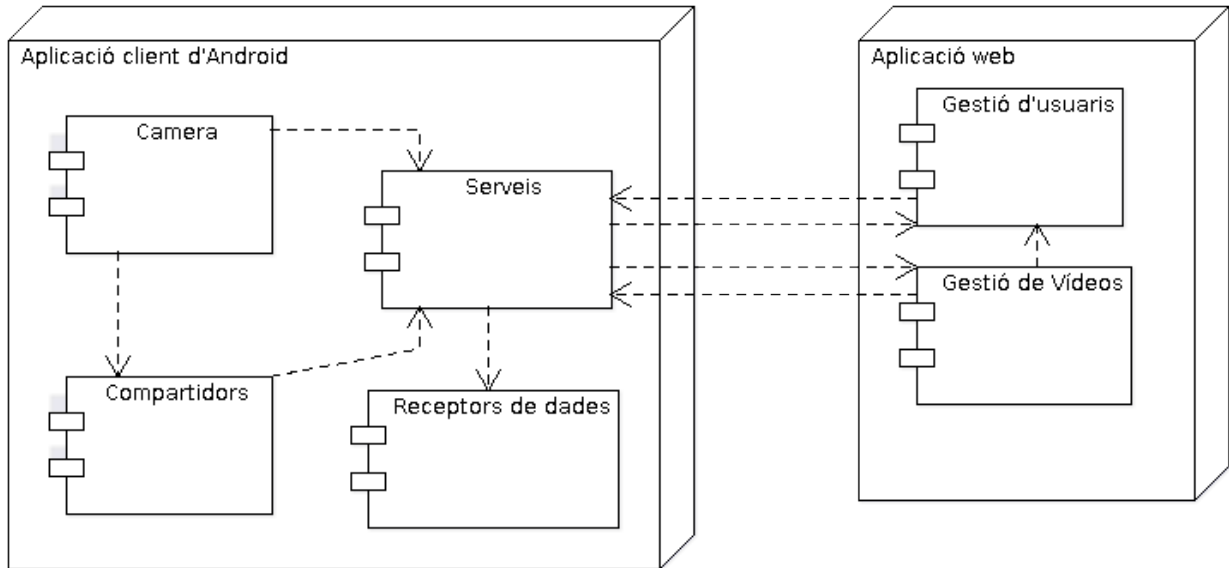
Es considera que "Afegir vídeo al servidor" és una extensió d'"Enregistrar vídeo" perquè els vídeos queden enregistrats al dispositiu client encara que no s'acabin pujant al servidor. De la mateixa manera el vídeo pot afegir-se simplement al servidor i finalitzar aquí l'execució, pel que "Compartir vídeo a Twitter" i "Compartir vídeo a Facebook" també son extensions.





## Diagrama de components

Aquest diagrama de components reflexa la relació entre els principals components de cadascuna de les aplicacions i la relació entre paquets de diferents aplicacions.



## ***Aplicació client d'Android***

El desenvolupament sobre dispositius Android és amb el llenguatge Java, també suporta l'execució de codi natiu en C o C++, però per aquest projecte no és necessari utilitzar-ho.

### **Components**

L'aplicació client d'Android es dissenya tenint en compte quins són els tipus de components d'aplicació més adequats per cada necessitat, respectant sempre les pràctiques recomanades per no tenir problemes posteriors, així:

- Les "pantalles" es tradueixen a activitats (Activity)
- Les peticions al servidor remot com a serveis (Service) més receptor (Broadcast Receiver)

Sobre la decisió d'utilitzar "Services" i "Broadcast receivers" per fer peticions i rebre dades del servidor, el tipus de component d'aplicació "Activity" és molt vàlid per les transicions entre pantalles, però quan demanem dades a un servidor remot estem exposats a problemes de connexió o amplis temps d'espera, pel que el gestor del processador d'Android podria parar o pausar l'activitat; la solució són els serveis, que queden en execució en background i es pot recuperar el seu estat al tornar d'una pausa o una aturada per part del gestor del processador, els "Broadcast receivers" s'utilitzen per respondre a events, en el cas que ens ocupa s'han creat dos events nous ja que el seu àmbit d'actuació és la pròpia aplicació.

### **Serveis**

Hi ha diversos tipus de serveis, segons com mantenen el lligam amb el context que l'ha iniciat, en el nostre cas, l'aplicació no gestionarà peticions concurrents, pel que podem fer ús de la classe IntentService, una extensió de Service que facilita el lligam de serveis amb la resta de components.

### **Comunicació amb el servidor remot**

Per gestionar les peticions a l'aplicació web s'han estudiat diverses opcions, des d'una solució ad-hoc amb URLConnection o HTTPClient, a frameworks com Spring Android o RESTlet.

Finalment s'ha escollit RESTlet per ser més lleuger que Spring Android i donar una sol·lució fiable i provada als problemes comuns d'aplicacions client que ens trobaríem al

desenvolupar la connexió des de zero. També hi ha un motiu acadèmic, ja que integrar una llibreria d'aquestes característiques representa una dificultat afegida.

## **Views**

Una View és una part d'interfície gràfica d'Android, l'arquitectura del sistema operatiu està preparada per separar la vista de la lògica de negoci, les Views corresponen a la vista i s'especifiquen mitjançant arxius en XML, tot i que també es poden afegir / modificar des del codi en Java sent aquesta la opció menys recomanable; lo que si s'ha de fer en Java son la gestió dels events associats a aquestes vistes, però les vistes en sí i les seves propietats es defineixen des dels arxius .xml. Hi ha molts tipus de Views, des de agrupacions de vistes fins a graelles, botons, vistes genèriques... L'enllaç adjunt il·lustra la funció i el gran número de vistes que hi ha disponibles.

## **RESTlet**

RESTlet és un framework de REST per Java, pot funcionar tant com a servidor com a client i disposa de classes per fer el mapeig d'objectes serialitzats en JSON a recursos amb els que poder treballar. On s'aprofiten plenament les funcionalitats de RESTlet és en un escenari on tant el client com el servidor funcionen amb Java, així ambdós disposen del mateix joc de recursos, en el nostre cas no podem fer-ho perquè la part del servidor funciona amb Zend Framework, que és per PHP.

## **Enregistrament de vídeo**

En aquest projecte es fa un ús extensiu del MediaRecorder d'Android per desar els vídeos. El procés d'enregistrar un vídeo passar per carregar-ne la previsualització del input de la càmera, això requereix afegir un tipus de View d'android anomenat SurfaceView que s'utilitza per dibuixar-hi, en aquesta cas, la seva extensió ens permet associar-hi les dades que es van reben des de la càmera del dispositiu Android.

S'ha decidit mantenir a les activitats com a controladors del flux d'execució de l'aplicació i deixar les particularitats de l'enregistrament de vídeos per a la classe VideoRecorder i CameraVideoPreview per separar el codi.

## **Traspàs del vídeo entre l'aplicació client i l'aplicació web**

La mida del vídeo és significatiu, pel que s'han valorat diverses opcions per fer el traspàs de dades, s'ha escollit base 64 per codificar el vídeo al enviar-lo i al costat del servidor per descodificar-lo. L'aplicació web treballa amb fluxos de dades per no carregar en una variable tot el contingut, lliurant així al servidor de temps de procés i aconseguint una resposta més ràpida.

## **Internacionalització**

Android proporciona un format per tot lo relacionat amb la internacionalització / localització pel que no cal dissenyar res en aquest sentit

## **Compartir continguts a Facebook**

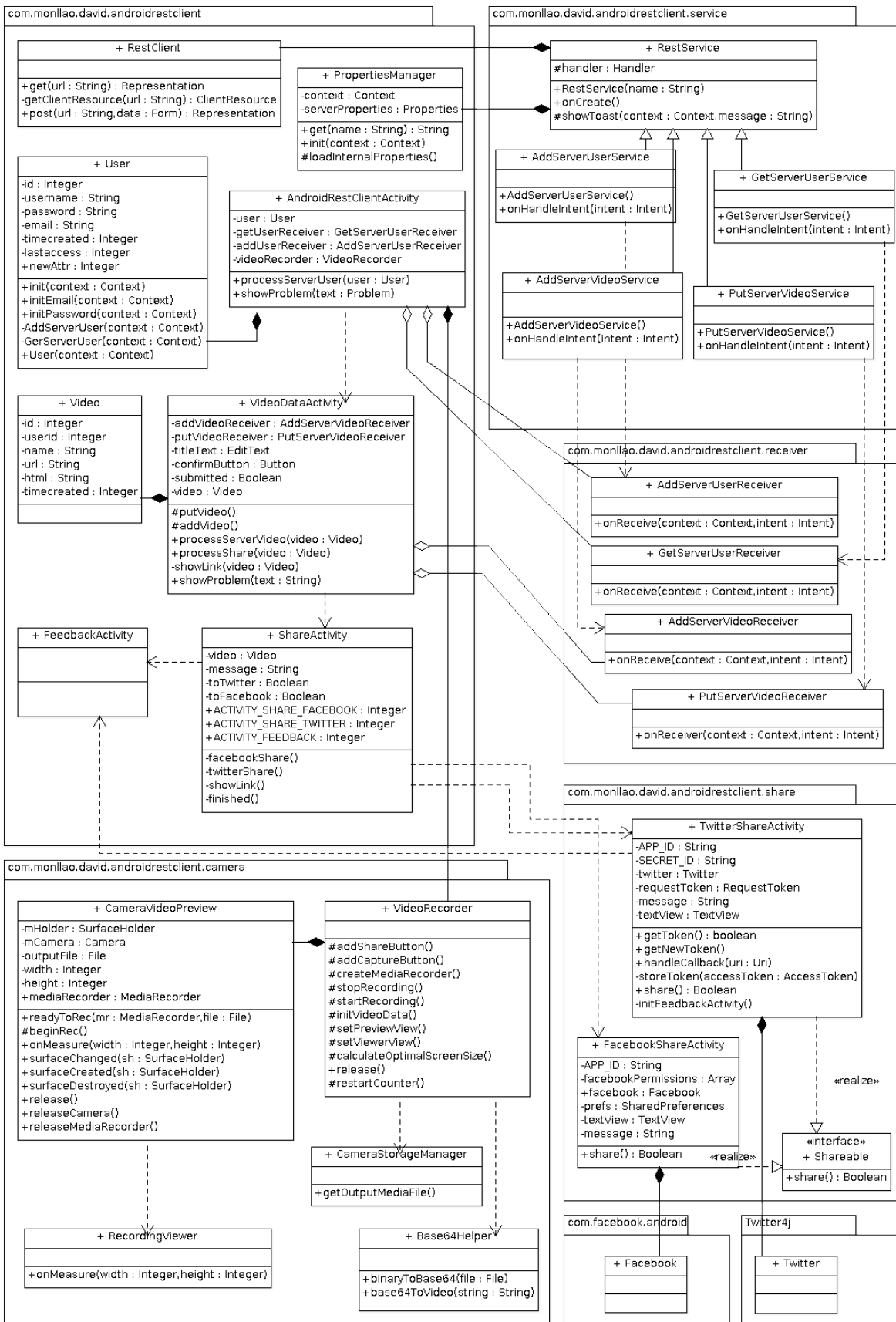
Facebook disposa d'un complet SDK per integrar-se a aplicacions d'Android i el proporciona de forma gratuïta. És un projecte independent, pel que s'ha de compilar juntament amb l'aplicació on s'integra.

## **Compartir continguts a Twitter**

Twitter proporciona una API basada en peticions HTTP per gestionar els permissos d'aplicacions externes que volen utilitzar serveis de Twitter, tot basat en OAuth 2.0 al igual que Facebook. El fet que estigui basat en HTTP condiciona el disseny, ja que el fluxe de l'aplicació es veurà alterat per la necessitat d'obtenir un token vàlid.

## **Diagrama de classes**

En cas que no es vegi correctament l'arxiu està adjunt està adjunt a la carpeta pics/ amb nom `Android_Class_Diagram.png`



## Capa de persistència

Les dades a desar per part de l'aplicació client d'Android inclouen la clau d'accés seleccionada per l'usuari i l'identificador d'usuari un cop s'ha creat al servidor.

Per altra banda també cal desar els tokens de connexió a Facebook i Twitter per no tenir que demanar autorització cada vegada a l'usuari, millorant així l'experiència d'usuari.

Basant-se amb la quantitat de dades requerida, el seu tipus (tot son parells de clau - valor) i les eines disponibles s'ha escollit usar "SharedPreferences" sobre SQLite o les altres alternatives ja que és una estructura extremadament senzilla. Les dades queden accessibles només des de la pròpia aplicació i no ens cal crear estructures de bases de dades.

## Aplicació web

Un dels objectius del projecte és familiaritzar-se amb el concepte REST, pel que l'aplicació web acceptarà peticions segons les pràctiques imposades per aquesta arquitectura. Això implica, com s'ha exposat anteriorment, utilitzar els diferents mètodes proporcionats pel protocol HTTP:

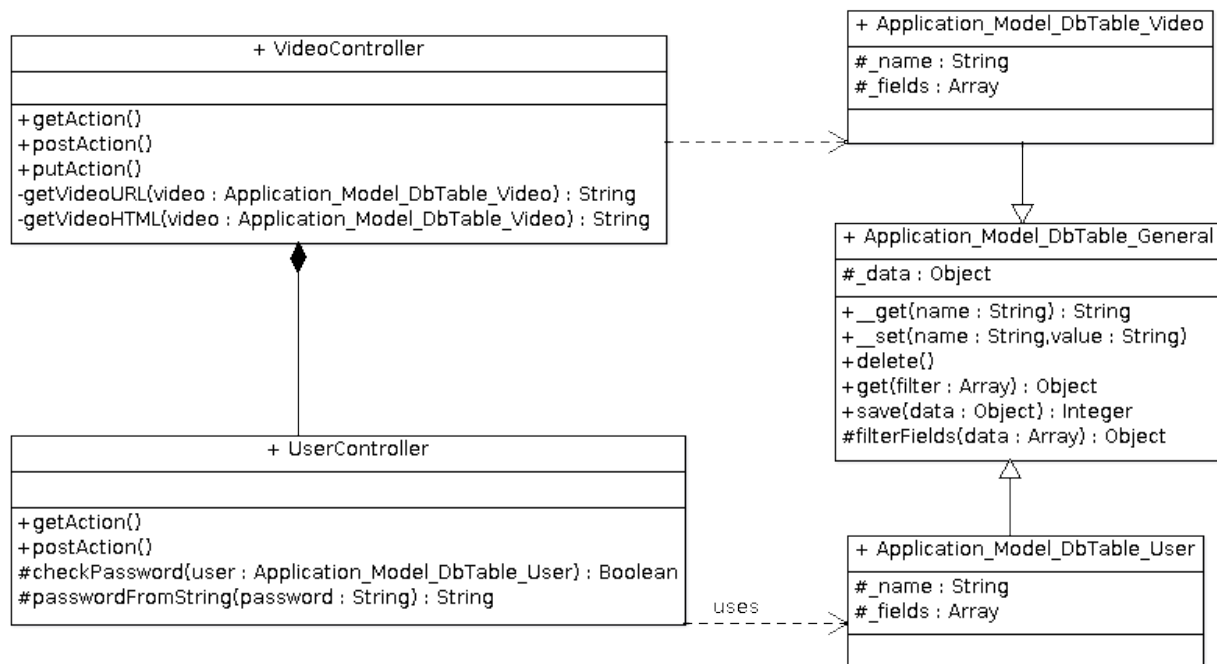
- POST per insercions
- GET per recuperació de dades
- PUT per actualitzacions
- DELETE per eliminacions.

Zend Framework no està dissenyat originalment per ser respondre a peticions REST, però la seva arquitectura és suficientment extensible per poder tractar els diferents mètodes d'HTTP i redirigir-los a accions CRUD segons defineix l'arquitectura d'aplicacions REST.

Preveient que en un futur (fora de l'àmbit d'aquesta assignatura) l'aplicació web podrà ser de consulta des de navegadors com qualsevol lloc web, l'aplicació no ha de ser tancada en aquest aspecte; per tal de solucionar aquest punt, el disseny contempla l'ús de la capçalera "Accept" d'HTTP per discernir quines peticions venen des d'un navegador i quines esperen una resposta menys "humana", així quan treballem des de l'aplicació client les peticions cap al servidor inclouran la capçalera d'HTTP "Accept: application/json", això permetrà a l'aplicació web saber quan una petició espera un format JSON (o qualsevol altre format de transmissió de dades entre aplicacions que es vulgui suportar més endavant) o una petició normal d'un usuari des d'un navegador web. Tota la lògica de negoci de les accions es mantindrà, només canvia el format de retorn. Per altra banda, s'ha afegit també una interfície "tonta" per identificar tots els controladors que permeten redirigir peticions cap a vistes en JSON.

## Diagrama de classes

En cas que no es vegi correctament l'arxiu està adjunt a la carpeta pics/ amb nom Web\_Class\_Diagram.png

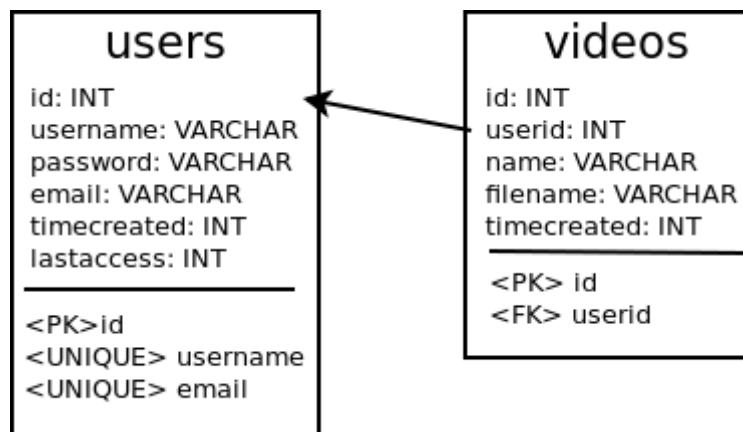


## Capa de persistència

Zend Framework no disposa d'un ORM, pel que s'ha estudiat la incorporació d'un framework de mapeig d'objectes a base de dades com pot ser Doctrine; finalment s'ha descartat la seva incorporació, ja que el projecte amb el que s'està treballant no fa un ús extensiu de la base de dades i treballa bàsicament amb entitats CRUD molt simples, pel que s'ha optat per no integrar un framework; no es descarta en fases posteriors o en un futur pròxim incorporar-ho si la gestió de les dades es complica. L'alternativa és desenvolupar una classe abstracta entre `Zend_Db_Table_Abstract` (la classe que gestiona l'accés a la base de dades per part d'objectes) i les entitats que crearem.



El diagrama entitat / relació queda així de simple.



# Implementació

## *Aplicació client d'Android*

### SDK

Google ens proporciona un SDK amb totes les eines necessàries per començar a desenvolupar de forma gairebé immediata, el paquet inclou aplicacions d'exemple per aprendre a utilitzar diverses funcionalitats dels mòbils (la càmera, el GPS...) eines per "debugging" i fins i tot una serie d'emuladors d'Android per cadascuna de les versions alliberades.

### IDE

En cas de desenvolupar utilitzant l'IDE Eclipse, com és el cas, Google també ha desenvolupat un plugin (Android Development Tools) que facilita l'empaquetat i el desplegament al dispositiu triat, sigui un dispositiu virtual o un dispositiu real connectat al PC amb un cable USB; per aquest projecte es disposa d'un dispositiu real, amb el sistema operatiu Android 2.3.5

### Debugging

L'eina logcat proporcionada junt amb l'SDK d'Android és clau per registrar tota activitat anòmala i veure les excepcions llançades pel sistema

### Instal·lació / Ús

Es disposa d'un servidor públic disponible durant el desenvolupament d'aquest Treball Fi de Grau per provar aquesta iteració, pel que l'aplicació client d'Android connectarà amb aquest servidor i hi desarà les dades.

1. Activar l'opció "Ajustes" -> "Aplicaciones" -> "Fuentes desconocidas" del dispositiu Android
2. Connectar per USB el dispositiu Android al PC

3. Copiar l'arxiu AndroidRestClient.apk proporcionat al dispositiu Android
4. Desconnectar el dispositiu Android del PC
5. Iniciar l'aplicació "Mis archivos"
6. Seleccionar el lloc on s'ha guardat AndroidRestClient.apk i seleccionar-lo per iniciar-ne la instal·lació

## **Problemes trobats durant el desenvolupament**

Els problemes desenvolupant pel sistema operatiu Android han estat molt més nombrosos que desenvolupant amb Zend Framework degut a que és una API relativament nova i s'ha de dissenyar l'aplicació per un gran nombre de tipus de dispositius, densitats de pantalla i resolucions de pantalla.

Un dels problemes més significatius que s'ha trobat degut al nombre de dispositius, pantalles i versions d'Android ha estat al concretar un format reproducible per tots els navegadors, s'ha aprofitat la nova etiqueta <video> d'HTML 5, substituint-la per un reproductor de Flash vídeo en els casos on no és possible utilitzar-la per restriccions del propi navegador.

Un altre inconvenient ha estat la rigidesa del enregistrator de vídeo, que dificulta mantenir un codi net ja que algunes dependències no queden cobertes fins que es genera l'event de surface changed, a més això retrasa sensiblement l'inici de la gravació.

També s'han trobat problemes al actualitzar la versió de l'Android Development Tools per Eclipse a la versió 18, hi havia un problema que impedia l'exportació de llibreries de Java, per sort el problema es soluciona amb una serie de tasques manuals.

## **Documentació de referència**

S'ha utilitzat l'eina doxygen per generar la documentació de referència a partir del codi Java escrit. Aquesta documentació està disponible a la carpeta `AplicacioClientAndroid_reference/` i s'hi pot accedir a través del seu `index.html`.

## ***Aplicació web***

## **IDE**

També s'ha escollit l'IDE Eclipse pel desenvolupament, en aquest cas s'ha optat per fer una instal·lació nova de la última versió (actualment la 3.7.2) disponible de Eclipse

Classic, posteriorment s'hi ha afegit el plugin PDT (PHP Development Tools) per facilitar el desenvolupament en PHP.

## **Debugging**

PHP disposa de paràmetres de configuració per mostrar informació de depuració, però a més d'això, s'ha afegit l'extensió XDebug per obtenir informació del flux d'execució de l'aplicació i detectar colls d'ampolla.

## **Generador de codi**

Una de les avantatges de desenvolupar amb Zend Framework és que disposa d'una eina de generació de codi, que suporta les tasques més comuns, com afegir models, controladors, accions... Per restringir-nos a les bones pràctiques d'ús de Zend Framework s'ha treballat amb el generador de codi enlloc de "picar" el codi d'inici bàsic.

## **Instal·lació / Ús**

Com s'ha comentat anteriorment es disposa d'un servidor públic amb tots els requisits de programari necessaris per instal·lar-hi l'aplicació web i amb suficient espai de disc per afegir-hi els vídeos que es generin durant el període de desenvolupament d'aquest Treball Fi de Grau.

La URL d'aquesta instal·lació temporal és <http://www.thejuju.net/restserver/public/>, conté vídeos de prova i està destinada a verificar que el desenvolupament del projecte és correcte.

## **Problemes trobats durant el desenvolupament**

L'únic problema significatiu que s'ha trobat durant el desenvolupament ha estat relacionat amb el mètode PUT de l'arquitectura REST; PHP el suporta com qualsevol altre mètode, però PUT envia els paràmetres adjunts per POST, el problema és que Zend Framework, tot i que reb tots els paràmetres, inicialment no els posa a disposició del desenvolupador i queden filtrats pel mètode d'HTTP que ha fet la petició. Per solventar el problema s'ha creat el Helper "App\_Controller\_Helper\_PutParamsGetter" per afegir aquest paràmetres encara que vinguem des d'una petició PUT.

## **Documentació de referència**

S'ha utilitzat l'eina phpxref per generar la documentació de referència a partir del codi PHP escrit. Aquesta documentació està disponible a la carpeta `AplicacioWeb_reference/` i s'hi pot accedir a través del seu `index.html`

## ***Gestió del codi***

El codi derivat del projecte s'ha distribuït sota llicència GPL 2, així tant l'aplicació client d'Android com l'aplicació web que fa de servidor estan disponibles per descarregar, modificar i utilitzar per qualsevol.

El desenvolupament del projecte s'ha portat a terme amb el sistema de control de versions GIT i amb un compte gratuït a github.com per mantenir-lo de forma centralitzada i remota.

S'ha treballat sempre a partir de branques de desenvolupament, pel que cada iteració ha tingut la seva branca fins que s'ha acabat la PAC i s'ha juntat a HEAD (o master segons les convencions utilitzades a GIT) deixant així la branca master com a última versió estable del codi.

Per exemplificar la explicació anterior i resumidament, la gestió del codi de l'aplicació client d'Android ha seguit aquest fluxe:

Creem el repositori i iniciem la fase d'inicialització a master

```
"git checkout master"
```

Hi treballem

```
"git commit ....."
```

Al iniciar la fase 2 creem una branca de desenvolupament nova, deixant master com a versió estable

```
"git checkout master"
```

```
"git checkout -b iteration-recorder"
```

```
"git commit ...."
```

Un cop finalitzada la fase 2 juntem els canvis a master

```
"git checkout master"
```

```
"git merge iteration-recorder"
```

Creem la branca per la fase 3 repetint el pas 3

```
"git checkout master"
```

```
"git checkout -b iteration-share"
```

"git commit ....."

Juntem els canvis a master deixant-la com a la darrera versió estable

"git checkout master"

"git merge iteration-share"

L'aplicació web ha seguit el mateix funcionament però amb una altra nomenclatura de branques

Seguint aquest model es pot treballar inclús amb diverses branques de desenvolupament, solucionant els bugs a master i fent rebase a les branques que en fan tracking.

Els push al repositori central no han seguit cap patró, han estat a mode de còpia de seguretat i de publicació de versions estables.

Codi de l'aplicació client d'Android:

[http://www.github.com/dmonllao/Android\\_Video\\_Rest\\_Client](http://www.github.com/dmonllao/Android_Video_Rest_Client)

Codi de l'aplicació web:

[http://www.github.com/dmonllao/Zend\\_Framework\\_Video\\_Rest\\_Server](http://www.github.com/dmonllao/Zend_Framework_Video_Rest_Server)

## Testeig

### ***Aplicació client d'Android***

Degut a la complexitat de l'aplicació i la gran quantitat de dispositius Android al mercat, els recursos necessaris per portar a terme un testeig unitari realment profitós escapen als recursos disponibles per portar a terme aquest Treball Fi de Grau; el problema rau en que on més problemes ens podem trobar és en l'adaptació de l'aplicació a les diferents versions d'Android, i als diferents tipus d'encoders de vídeo i formats de pantalla, que varien de dispositiu a dispositiu.

En aplicacions amb interfícies d'usuari més senzilles, aquesta feina es pot portar a terme de forma fàcil utilitzant les pròpies eines de testeig d'Android; en el cas que ens ocupa però, els emuladors de dispositius Android no suporten coses tan bàsiques per aquesta aplicació com l'enregistrament de vídeo, pel que el testeig unitari que podríem fer seria d'una granularitat tan petita que no resultaria profitós.

Com a alternativa s'ha optat per utilitzar l'eina "Android Monkey". Aquesta eina proporcionada junt amb SDK d'Android permet fer un testeig integral al generar N events aleatoris buscant excepcions del sistema, els events generats no segueixen cap patró de fluxe de treball habitual d'un usuari, com el seu nom indica és el que faria un mono o un nen d'un any si tingués el dispositiu a les mans, pel que és una eina idònia per detectar excepcions no capturades, fluxos d'execució no controlats o combinacions d'aplicacions no provades.

Degut a que els emuladors que proporciona el SDK d'Android no permeten l'enregistrament de gravacions, com ja s'ha comentat, s'ha executat Android Monkey sobre dispositius reals. S'ha provat amb un Samsung Galaxy S I9000 i amb un Samsung Galaxy Nexus i9250 generant 1000 events aleatòris cada vegada en cadascun. En ambdós casos i en repetides ocasions els tests s'han superat satisfactòriament.

Adjunt al Treball Fi de Grau està disponible el volcat del monitoratge i el feedback d'una de les execucions d'Android Monkey sobre el Samsung Galaxy S i9000 amb nom d'arxiu "AndroidMonkey.monitor.txt".

## Usabilitat i interfícies d'usuari

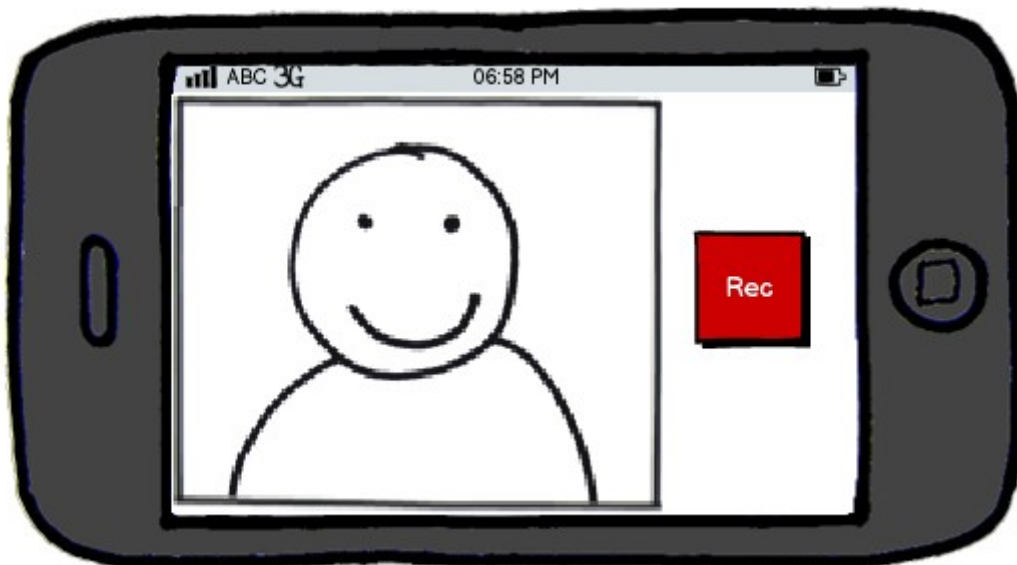
Les interfícies d'usuari s'han dissenyat per donar la idea de que és una aplicació ràpida i senzilla d'utilitzar; per reforçar aquesta idea s'ha escollit un estil semblant als còmics, tant la tipografia utilitzada com les icones son d'aquest estil.

Junt amb aquest estil s'ha seguit la guia de bones pràctiques de disseny d'aplicacions per Android (enllaç adjunt a la bibliografia) per assegurar una experiència d'usuari satisfactòria i motivadora.

Sobre l'usabilitat, un dels reptes d'aquest projecte ha estat que l'experiència d'usuari no es vegi ressentida tot i estar transmetent una gran quantitat de dades cap a un servidor remot; encara que, com es detallarà a l'apartat "Futurs desenvolupaments o possibles millores", encara s'ha de treballar sobre el tema, s'ha fet un esforç en distribuir els processos en background de forma que l'usuari estigui fent altres accions mentre es van pujant el vídeo, que és el que més temps requereix.

Aquest és el fluxe d'execució normal

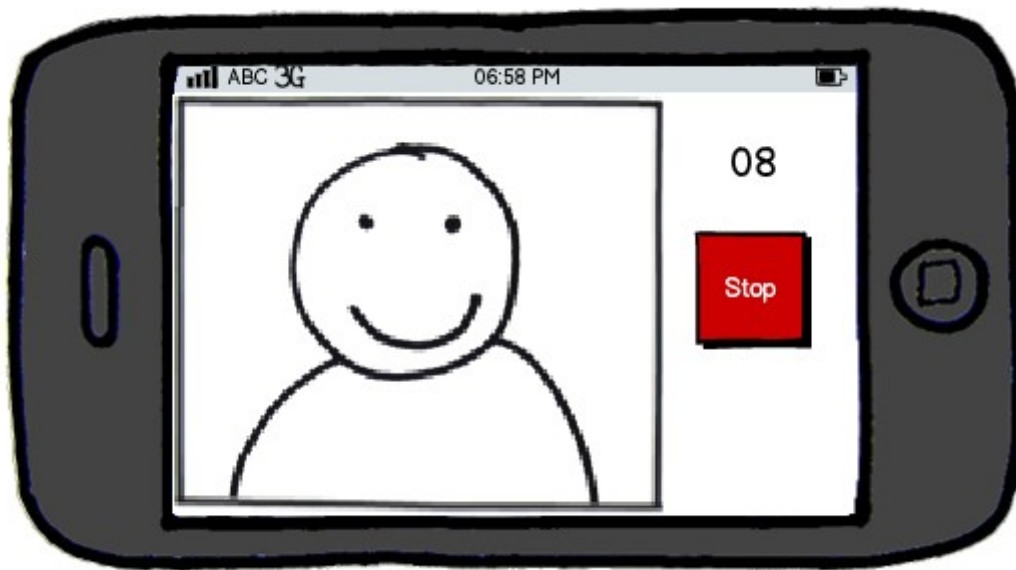
### 1.- Previsualització de la càmera amb el botó per iniciar l'enregistrament



created with Balsamiq Mockups - [www.balsamiq.com](http://www.balsamiq.com)



2.- Enregistrament en curs, apareix el comptador de 10 segons baixant i el botó per parar l'enregistrament en qualsevol moment



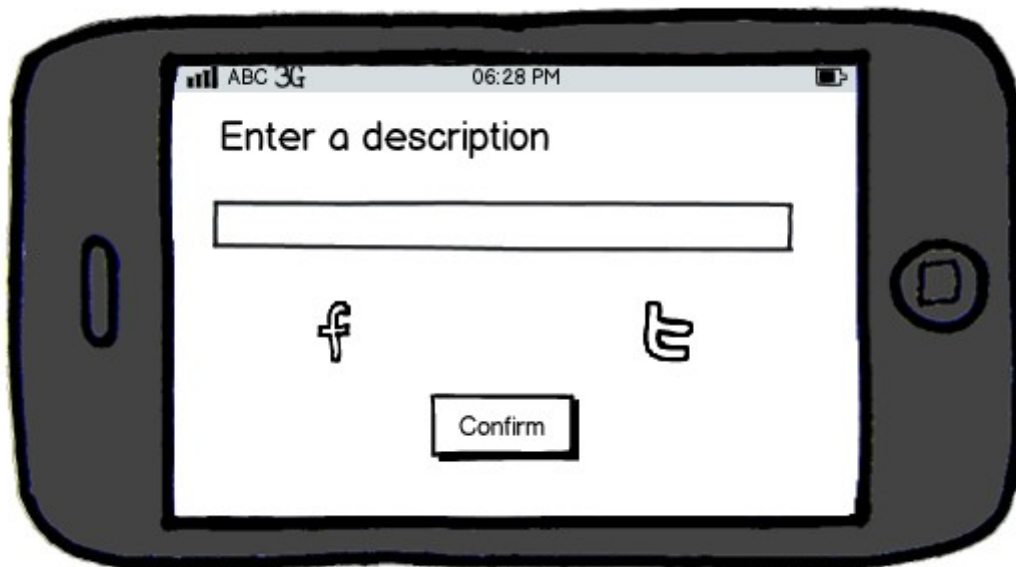
created with Balsamiq Mockups - [www.balsamiq.com](http://www.balsamiq.com)

3.- Visualització del vídeo enregistrat, hi ha la possibilitat de torna a enregistrar-ne un o de compartir-lo



created with Balsamiq Mockups - [www.balsamiq.com](http://www.balsamiq.com)

4.- Afegir una descripció textual del vídeo i seleccionar a quines xarxes socials es vol compartir



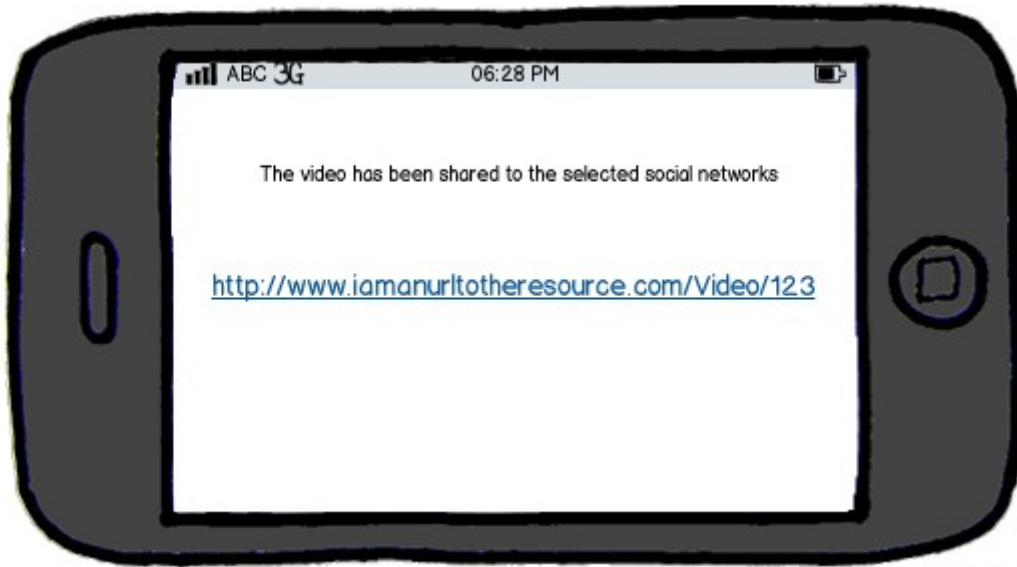
created with Balsamiq Mockups - [www.balsamiq.com](http://www.balsamiq.com)

5.- Un cop entrada la descripció i seleccionades les xarxes on compartir-lo els elements del formulari queden deshabilitats i les xarxes seleccionades en color enlloc de en blanc i negre



created with Balsamiq Mockups - [www.balsamiq.com](http://www.balsamiq.com)

6.- Es mostra el feedback a l'usuari amb un enllaç per si també el vol compartir per correu, Android permet compartir qualsevol enllaç deixant el ratolí 2 segons damunt.



created with Balsamiq Mockups - [www.balsamiq.com](http://www.balsamiq.com)

## Conclusions

A continuació detallo conclusions relacionades amb cada punt per després seguir amb conclusions generals

- Aprendre a desenvolupar aplicacions pel SO Android

El punt fort del sistema operatiu Android, de moment i en alguns aspectes, pot ser alhora el seu punt dèbil, hi ha moltes diferències de hardware i de software entre dispositius i una gran varietat de versions, a més, a dia d'avui la fragmentació de versions s'està estancant a la branca 2.x, quan ja fa mesos que ha sortit la versió 4. Això fa que s'hagi de desenvolupar pensant amb tots els dispositius, cosa que complica els desenvolupament, les solucions que proporciona Android amb les versions que he treballat, a partir de 2.3.3 son bastant limitades, i pel que he llegit queden solucionades amb la versió 4, on s'unifica el desenvolupament del sistema operatiu per mòbils i el sistema operatiu per tablets.

Com a punts positius, és un plaer disposar d'un kit de desenvolupament tan complet i de tantes eines diferents de debug i testeig, a més de funcionar amb un llenguatge tan popular com Java, cosa que facilita la introducció de nous desenvolupadors.

- Familiaritzar-se amb el concepte de software orientat a serveis i amb el concepte de REST

Tenia la idea de que REST era més un concepte teòric i que no tenia tantes implicacions tècniques relacionades amb el protocol HTTP. Si que el concepte s'utilitza molt sovint i fins i tot es podria dir que REST i SAAS s'estan convertint en paraules de moda, però al practicar tant la part del servidor com la del client he entès les implicacions tècniques que té, he conegut més profundament el protocol HTTP i he entès realment el que hi ha sota, recursos, components (dos aplicacions en el cas que ens ocupa) representacions (objectes o col·leccions d'objectes en JSON en el nostre cas) i connectors entre components.

Com a representació pura de les comunicacions sense estat entre aplicacions sembla un pèl irònic que s'utilitzi tant en la web 2.0, encara que a la pràctica sigui la millor opció que hi ha.

- Analitzar les possibilitats de connexió entre aplicacions d'Android i aplicacions RESTful

Restlet destaca per sobre de les altres opcions, i és molt més completa que les

altres opcions que s'han valorat, realment fins que no hi he treballat extensivament no ha sigut quan he començat a entendre el concepte REST. De rebot he practica amb el format JSON i he descobert que hi ha una cosa meravellosa que es diu Jackson i que junt amb JSON i servidors REST poden donar-nos moltes alegries

- Desenvolupar una aplicació complexa amb Zend Framework

Zend Framework consta d'un esquelet de MVC i d'un conjunt de components independents, utilitzar-ne els components és donar-li ús de llibreria i això no és el que es perseguia amb aquest objectiu, pel que s'ha treballat amb l'esquema de MVC proporcionat per la última versió estable de Zend Framework. Encara que l'aplicació web té una finalitat molt limitada dins l'àmbit d'aquest Treball Fi de Grau, tota el disseny ha tingut en compte que en un futur, com s'explica a l'apartat següent "Futurs desenvolupaments", hi ha feina a fer. S'ha fet ús de helpers seguint les bones pràctiques de programació proposades per Zend i s'ha implementat una aplicació 100% RESTful sense delegar absolutament res als controladors ni als models que se n'encarreguen de la lògica de negoci pròpia de l'aplicació.

- Aprofundir en l'ús d'OAuth 2.0

Tant Facebook com Twitter permeten als desenvolupadors connectar amb els seus serveis a través d'OAuth 2.0. Facebook posa a disposició dels desenvolupadors SDKs adaptats a l'entorn / llenguatge de treball del programador, cosa que facilita molt la feina, per altra banda Twitter basa tot el procés d'autorització de l'aplicació en HTTP, cosa que complica el retorn, però resulta molt més uniforme.

- Conèixer les possibilitats que ofereixen les APIs de les xarxes socials amb més difusió

Aquest és el punt que més descuidat ha quedat, tot i que ja era una mica previsible, ja que només s'ha treballat un servei de cada xarxa social; sobre tot facebook disposa d'un conjunt de permissos molt ampli.

- Obtenir dos productes

S'han obtingut els dos productes i s'han compartit amb llicència GPL, encara hi ha marge per millores i per derivar-ne productes comercials com s'explicarà a les seccions "Futurs desenvolupaments" i "Solució comercial".

## **Futurs desenvolupaments**

L'aplicació es pot millorar, com tot, des de moltes vessants, però des del meu punt de vista hi ha diversos aspectes que caldria apurar abans de pensar en solucions comercials reals.

### ***Millorar la codificació de vídeo***

Actualment la forma com cada dispositiu codifica els vídeos no és estàndard, tot i que, teòricament, cada versió d'Android conté un llistat tancat de codificadors de vídeo i d'àudio, això fa que, tot i utilitzar H264 per codificar els vídeos i seleccionar MP4 com a contenidor, alguns dispositius utilitzen el codificador 3GP, que no és compatible amb tots els navegadors web, cosa que fa que no es reproduïxi la imatge del vídeo, només el so.

Per sort és un petit nombre de dispositius (1 de 7 dispositius reals diferents on s'ha provat l'aplicació) però per solucionar aquest inconvenient caldria passar el vídeo per un recodificador com FFmpeg per empaquetar-lo en un format compatible amb tots els navegadors moderns, com MP4 (sense àtoms de 3GP) o Flash Video.

Altres portals amb funcionalitat semblant com <http://www.viddy.com>, tot i disposar d'una gran infraestructura i molts recursos no han solucionat aquest problema, només cal consultar el seu espai web amb el navegador Firefox per comprovar-ho.

### ***Millorar la qualitat del vídeo***

S'ha de treballar amb l'aspect ratio més adequat per cada dispositiu per tal que el codificador de vídeo optimitzi recursos; una opció bastant viable és fer els botons transparents podent aprofitar així la totalitat de la pantalla i la resolució òptima del dispositiu.

### ***Dotar de funcionalitat "web" a l'aplicació web***

L'aplicació web fa la funció de servidor de l'aplicació client d'Android dins l'àmbit del Treball Fi de Grau, però qualsevol aplicació comercial que s'hagi de treure d'aquest projecte necessitarà un portal web on llistar els vídeos de cada usuari, que els usuaris puguin eliminar o modificar les descripcions dels vídeos o coses per l'estil, que si no estan es consideraran mancances.

El disseny de l'aplicació ja té en compte aquest aspecte i diferencia les peticions que venen d'una aplicació client d'Android de les que venen d'un navegador web a partir del valor d'una capçalera d'HTTP, mantenint així una aplicació 100% REST, identificant recursos i accions a partir de URL úniques i canviant la vista segons la capçalera d'HTTP que se li passi, pel que implementar aquest apartat no requerirà cap redisseny de l'eina.

### ***Opció de fer els vídeos no públics***

Com a subapartat a destacar del punt anterior, ara mateix tots els vídeos que s'enregistren són públics, caldria implementar una opció per a que els vídeos que l'usuari marqui com a "disponibles només per a qui jo li doni l'enllaç" no puguin ser llistats junt amb els altres vídeos de l'usuari quan un usuari extern a la web consulta el "canal" (o índex en termes REST) de l'usuari.

## **Solució comercial**

Durant el desenvolupament del projecte s'ha trobat un producte competidor ja ben establert al mercat <http://www.viddy.com>, això dificulta molt la implantació de l'aplicació com a eina de propòsit general, com a alternatives es pot treballar amb temàtiques més concretes, per exemple utilitzant els efectes que proporciona la càmera, com el blanc i negre per fer vídeos antics.

Una de les altres temàtiques a explotar serien les manifestacions, on molts cops, en alguns països més que en d'altres, és bastant habitual que les autoritats requirixin els dispositius que estan enregistrant fets que no interessin.

## **Costos associats**

Els costos associats estan bàsicament relacionats amb el hosting de l'aplicació, una manera de cobrir aquestes despeses seria afegir publicitat al portal web des d'on es veurien els vídeos o a l'aplicació client d'Android, tot i que aquesta segona opció seria menys recomanable per temes d'espai en pantalla. Segons el volum d'usuaris els costos de hosting augmentarien, però també augmentarien els ingressos per publicitat al portal web per compensar-ho.

Una alternativa habitual a la publicitat són els serveis "premium", si l'aplicació permet registrar vídeos de 10 segons sense cap efecte o filtre, un possible servei premium permetria augmentar la durada del vídeo i afegir efectes als vídeos com blanc i negre...; altres tipus de serveis premium més complicats de gestionar podrien estar relacionats amb publicitar vídeos o usuaris al portal web.



## Glossari

- API : Application Programming Interface, fa referència a el conjunt de mètodes d'una llibreria o aplicació que pot utilitzar un desenvolupador
- ADT: Android Development Tools, conjunt d'eines del SDK d'Android per al desenvolupament per la plataforma.
- Android Monkey: Eina proporcionada junt amb Android Development Tools (ADT) per simular tot tipus d'events.
- ArgoUML: Programari d'edició de diagrames en UML, és de codi obert. (<http://argouml.tigris.org/>)
- Balsamiq Mockups : Programari de disseny de maquetes, útil per prototips d'interfícies gràfiques (<http://www.balsamiq.com/products/mockups>)
- CRUD : Son les sigles de Create, Retrieve, Update i Delete, les accions típiques que es porten a terme amb entitats
- Eclipse: IDE de desenvolupament per diversos llenguatges (<http://www.eclipse.org/>)
- GIT: Sistema de control de versions distribuït (<http://git-scm.com/>)
- HTTP : Protocol utilitzat per la WWW
- IDE : Integrated Development Environment, un entorn de desenvolupament amb una serie de funcionalitats per facilitar el desenvolupament d'aplicacions
- Internacionalització : Fa referència a arquitectures de programari que s'adapten a l'idioma de l'usuari final permetent afegir traduccions de forma fàcil
- Java : Llenguatge de programació orientat a objectes i compilat

- JSON : JavaScript Object Notation, format lleuger per contenir informació serialitzada, útil per intercanvi de dades entre aplicacions
- Kernel de Linux : Core del sistema operatiu Linux, inclou els controladors que interactuen amb el hardware
- Localització : Fa referència a arquitectures de programari que s'adapten a la localització de l'usuari final proporcionant-li valors adequats a la seva localització
- MediaRecorder : La classe de l'API del sistema operatiu Android encarregada de l'enregistrament a través del driver de la càmera del dispositiu.
- ORM : Son les sigles de mapeig objecte relació, és un tipus de programari que serveix per abstraure l'accés a la base de dades per part d'objectes, fent de pont entre el mon Entitat / Relació i el mon de la Orientació a Objectes
- PHP : Llenguatge de programació d'scripting amb possibilitats d'orientació a objectes (<http://www.php.net>)
- Restlet: Llibreria per facilitar el desenvolupament d'aplicacions REST per Java
- SDK : Software Development Kit, conjunt d'eines per facilitar el desenvolupament d'aplicacions informàtiques
- Spring : Framework per Java amb molta difusió i molt modular (<http://www.springsource.org/>)
- SQLite : Motor de base de dades lleuger (<http://www.sqlite.org/>)
- UML : Llenguatge de modelatge d'aplicacions, inclouen una sèrie de diagrames per especificar tots els aspectes d'un producte informàtic
- Twitter4j: Llibreria per facilitar el desenvolupament d'aplicacions de Java que treballen sobre la API pública de Twitter
- Zend: Rama comercial del llenguatge de programació PHP, és que se n'encarrega del manteniment de Zend Framework entre d'altres coses (<http://www.zend.com>)

## Bibliografia

- <http://developer.android.com/design/index.html>, Disseny per Android
- <http://developer.android.com/index.html>, Desenvolupament per Android
- <http://developer.android.com/guide/topics/media/camera.html>, Ús avançat de la càmera dels dispositius Android
- <http://www.stack.nl/~dimitri/doxygen/>, Doxygen
- <https://github.com/facebook/facebook-android-sdk>, SDK de Facebook per Android
- <http://twitter4j.org/en/index.html>, Twitter per aplicacions en Java
- <http://www.restlet.org/>, Llibreria Restlet
- <http://framework.zend.com/>, Lloc web de Zend Framework
- <http://phpxref.sourceforge.net/>, PHPxref
- <http://mediaelementjs.com/>, MediaElementJS
- <http://www.gnu.org/licenses/gpl-2.0.txt>, GPL versió 2