

El meu àlbum digital:

Model-Vista-Controlador, PHP i Android

Victor Agudo

Resum —

El projecte "El meu àlbum digital" neix de la necessitat de guardar les nostres imatges en àlbums digitals. Per a això, s'ha desenvolupat una aplicació que permet sincronitzar les nostres imatges i associar-les a àlbums que anteriorment hem creat. La solució, una aplicació mòbil i una aplicació web amb una interfície moderna i intuïtiva per oferir un servei encara inexistent al mercat.

Gràcies a les noves tecnologies podem capturar instantànies en el moment que vulguem i sincronitzar-ho en el núvol en qüestió de segons.

Paraules clau —API;CakePHP;C;CSS;Framework;GUI;MD5;MySQL;MVC;PERL;PHP;SDK;UML;XP

Abstract—

The project "My Digital Album" is born from a need to save our digital pictures into albums. For that reason, a new application has been developed to allow our pictures to synchronise and combine in albums previously created. The solution, a mobile phone and a web application with a modern and intuitive interface to provide a service still existing in the market.

Thanks to new technologies we can capture snapshots anytime and even synchronise them in the cloud in a question of seconds.

Index Terms—API;CakePHP;C;CSS;Framework;GUI;MD5;MySQL;MVC;PERL;PHP;SDK;UML;XP

1 INTRODUCCIÓ

Aquest projecte anomenat "El meu àlbum digital" té com a idea principal la sincronització en el núvol de les imatges que fem amb el nostre smartphone i la creació d'àlbums digitals de manera senzilla i ràpida. Moltes vegades ens oblidem de realitzar còpies de seguretat dels nostres documents, en aquest cas fotografies personals. Aquest projecte ofereix la possibilitat de mantenir les nostres imatges al núvol organitzades en àlbums i accessibles des de qualsevol lloc i solament amb un ordinador o smartphone.

Avui dia existeixen aplicacions d'escriptori i serveis web que permeten l'organització de fotos en àlbums digitals. Al-

guns serveis web permeten enviar les fotos digitals a una pàgina web i fer un àlbum de fotos virtual. Diversos serveis en línia ofereixen opcions tals com l'edició de fotos, disposicions, etc. Alguns serveis fins i tot amplien la idea d'un àlbum de fotos permetent que una fotografia es converteixi en part de molts àlbums, agregant comentaris i paraules claus a aquestes fotografies i utilitzant aquestes paraules com a mitjà per crear nous àlbums de fotos. Alguns àlbums de foto en línia proporcionen serveis tals com a impressió i opcions per carregar fotos en els telèfons mòbils.

Projectes semblants a aquest es troben Fotolog, Flickr, entre d'altres.

Aquets projectes tenen la problemàtica de la sincronització mòbil – web. En aquest projecte s'afegeix la funcionalitat de tenir un “banc” d'imatges personals al núvol amb la possibilitat de sincronitzar amb el teu smartphone.

El motiu pel qual m'ha portat a desenvolupar aquest projecte consisteix bàsicament en la necessitat de mantenir les nostres fotografies organitzades en diferents àlbums. Aquest projecte ofereix la possibilitat de tenir un banc d'àlbums i imatges sincronitzat al núvol. S'ofereix també l'alternativa de pujar imatges per aquesta interfície, eliminar àlbums i fotografies sempre que vulguis. S'ha desenvolupat també una interfície de missatgeria instantània per poder conversar amb altres usuaris sempre que ho desitgis.

La idea d'aquest projecte es millorar els que ja existeixen, per això, s'afegeix la possibilitat de tenir sincronitzades les imatges i els àlbums des de dues aplicacions, mòbil i web.

A continuació es detallen els objectius principals del projecte.

- Desenvolupar una aplicació mòbil per a dispositius mòbils per a capturar i sincronitzar imatges amb un servidor.
- Desenvolupar una interfície web per a gestionar la creació, edició i eliminació d'imatges d'un usuari.
- Dissenyar i crear un sistema de base de dades per a gestionar àlbums, imatges i usuaris.
- Creació, edició i eliminació d'àlbums digitals tant en l'aplicació mòbil com en la aplicació web.
- Dissenyar un sistema per a generar àlbums digitals a partir de les imatges seleccionades per l'usuari.
- Implantar un sistema de validació i control d'usuaris en l'aplicació mòbil i en l'aplicació web.

L'estructura de la resta de l'article es detalla a continuació: a les seccions 2 i 3 s'explica els aspectes tècnics del projecte i la metodologia que s'ha seguit. La secció 3 està dedicada als resultats obtinguts i finalment la secció 4 conté les principals conclusions del projecte.

2 ASPECTES TÈCNICS UTILITZATS

2.1 TECNOLOGIA XP

L'aplicació per Smartphone ha estat desenvolupada en llenguatge Java disponible per a la plataforma Android versió 4.0 o superior. Des d'aquesta aplicació, també es pot gestionar els àlbums i imatges, podent crear, modificar o eliminar-los.

El projecte s'ha desenvolupat seguint Xtreme Programming.

XP sorgeix com una nova manera d'encarar projectes de Software, proposant una metodologia basada essencialment en la simplicitat i agilitat. Les metodologies de desenvolupament de programari tradicionals (cicle de vida en cascada, evolutiu, en espiral, iteratiu, etc.) apareixen, comparats amb els nous mètodes proposats en XP, com a pesats i poc eficients. La crítica més freqüent a aquestes metodologies “clàssiques” és que són massa burocràtiques. Cal fer tantes coses per

seguir la metodologia que, de vegades, el ritme sencer del desenvolupament es retarda. Com a resposta a això, s'ha vist en els últims temps el sorgiment de “Metodologies Àgils”. Aquests nous mètodes busquen un punt mitjà entre l'absència de processos i l'abús dels mateixos, proposant un procés l'esforç del qual valgui la pena.

Els mètodes àgils canvien significativament alguns aspectes de les metodologies “clàssiques”:

- Els mètodes àgils són adaptables en lloc de predictius. Els mètodes “clàssics” tendeixen a intentar planejar una gran part del procés del programari en gran detall per a un termini llarg de temps.
- Els mètodes àgils són orientats a la gent i no orientats al procés. L'objectiu dels mètodes “clàssics” és definir un procés que funcionarà bé independentment de qui ho utilitzi.

XP és una de les anomenades metodologies àgils de desenvolupament de Software amb més èxit. La metodologia proposada en XP està dissenyada per lliurar el Software que els clients necessiten en el moment en què ho necessiten. XP proposa als desenvolupadors a respondre als requeriments canviants dels clients, encara en fases finals del cicle de vida del desenvolupament.

La metodologia també fa èmfasi el treball en equip. Tant gerents com clients i desenvolupadors són parts del mateix equip dedicat a lliurar programari de qualitat.

A la següent figura es mostren les parts en les que consisteix una iteració en aquesta metodologia.

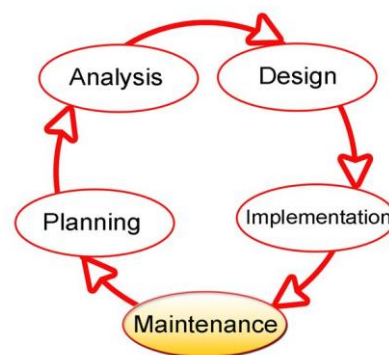


Fig. 1 - Iteració Xtreme Programming

Els apartats que el formen són:

- **Planificació:**
 - Fase de planificació del projecte, calendaris, timings, entregables, membres de l'equip, stakeholders, rols, etc...
- **Anàlisi**
 - Fase d'anàlisi del projecte, objectius, limitacions, requeriments, entorn de treball, etc...
- **Disseny**
 - Fase de disseny del llenguatge de programa-

ció, diagrames UML, diagrames entitat-relació, diagrama de classes, etc...

- **Implementació**
 - Fase d'implementació, desenvolupament de l'aplicació i treball en equip, posta en funcionament i test de qualitat.
- **Manteniment**
 - Fase de manteniment de l'aplicació, millores a afegir o errors a corregir.

Els rols d'aquesta metodologia acostumen a ser:

- **Programador:** Escriu les proves unitàries i produeix el codi del sistema.
- **Client:** Escriu les històries d'usuari i les proves funcionals per validar la seva implementació. Assigna la prioritat a les històries d'usuari i decideix quins s'implementen en cada iteració centrant-se a aportar el major valor de negoci.
- **Tester:** Ajuda al client a escriure les proves funcionals. Executa proves regularment, difon els resultats en l'equip i és responsable de les eines de suport per a proves.
- **Tracker:** És l'encarregat de seguiment. Ha de verificar el grau d'encert entre les estimacions realitzades i el temps real dedicat, comunicant els resultats per millorar futures estimacions.
- **Entrenador:** Responsable del procés global. Guia als membres de l'equip per seguir el procés correctament.
- **Consultor:** És un membre extern de l'equip amb un coneixement específic en algun tema necessari per al projecte. Ajuda a l'equip a resoldre un problema específic.

A continuació, es mostra una llista amb la planificació establerta des de l'inici del projecte, es divideix en diferents apartats així com les hores que s'han estimat per a cada tasca.

- Cerca d'informació sobre les eines a utilitzar, així com les API's de referència dels llenguatges a utilitzar. Aprenentatge de patrons de disseny de l'aplicació (50h)
 - Recerca d'informació en les diferents fonts; internet, llibres, articles, etc...
 - Recerca de manuals i guies de desenvolupament en els llenguatges que utilitzarem
 - Aprenentatge de les diferents tecnologies i patrons de disseny que utilitzarem
- Documentació del projecte, com requeriments, traçabilitat, documents del projecte com pressupost, etc... (40h)
 - Informe inicial
 - Anàlisi de requeriments
 - Anàlisi de dades
 - Informe de progrés I
 - Informe de progrés II
 - Article final
- Anàlisi i disseny de l'aplicació, com disseny UML, classes, etc... (15h)
 - Disseny del diagrama UML
 - Disseny del diagrama de classes
- Anàlisi i disseny de la base de dades relacional (15h)
 - Disseny del diagrama entitat relació
 - Disseny del diagrama relacional

- Implementació de l'aplicació mòbil. (25h)
 - Desenvolupament de l'aplicació mòbil amb Java + Android SDK
 - Implementació del sistema de validació d'usuaris
- Implementació de l'aplicació web. (25)
 - Desenvolupament de l'aplicació web amb CakePHP i MySQL basat en MVC
- Realitzar els test de validació sobre l'aplicació (60h)
 - Disseny i creació dels Test de validació
 - Execució del test de validació
 - Correcció dels possibles errors
- Disseny de les interfícies gràfiques (GUI). (45h)
 - Disseny de les diferents pantalles de l'aplicació mòbil, com la pantalla de login, registre, àlbums, fotografies, etc...
 - Disseny de la interfície gràfica de l'aplicació web amb Bootstrap 3.

2.2 ARQUITECTURA MOBIL

Per a l'arquitectura mòbil s'ha utilitzat el llenguatge Java juntament amb Android SDK.

Android és un sistema operatiu i una plataforma software, basat en Linux per a telèfons mòbils. El sistema operatiu de Google, Android, té només 6 anys i en aquest temps ho hem vist evolucionar d'una manera realment impressionant, mitjançant les versions que periòdicament treu al mercat Android és un clar exemple de tots els canvis que el sistema ha sofert.

Són molts els avantatges que aquest sistema operatiu té, però els principals són:

- Utilització d'APIs
- Compatible amb tecnologies Web, ja que utilitza el motor WebKit
- Totes les aplicacions són creades per igual
- Un desenvolupament ràpid i fàcil, gràcies al seu complet SDK
- Aplicacions corrent en paral·lel, entorn multiprocés

A continuació es descriuen les eines més importants que ens proporciona el SDK:

- Android Emulator
- Dalvik Debug Monitor Service (DDMS)
- Plugin de Desenvolupament para Eclipse (ADT)
- Android Debug Bridge (ADB)
- Eina d'empaquetat (AAPT)

Dalvik virtual Machine es la màquina virtual pròpia d'Android que permet executar codi natiu en els dispositius amb aquest sistema operatiu. Esta optimitzada per a requerir poca memòria i esta dissenyada per a permetre executar varies màquines virtuals simultàniament.

Les característiques més importants d'aquesta màquina virtual són:

- Utilitzada en entorns amb restriccions de memòria i processador.
- Eina dx converteix .class (classe o aplicació en JAVA) a dex.
- Executa aplicacions en format .dex (Dalvik Executable).
- Permet l'execució de diverses instàncies alhora.
- No és una JVM, no executa bytecode sinó registres.
- Classes de Java EM, AWT o Swing no estan suportades.

2.2.1 LENGUATGE DE PROGRAMACIÓ MÒBIL

El llenguatge de programació utilitzat per al dispositiu mòbil, és Java amb Android SDK (Software Development Kit).

Android està basat en el nucli de Linux, té accés als seus recursos, podent gestionar-ho, gràcies a que es troba en una capa per sobre del Kernel, accedint així a recursos com els controladors de pantalla, càmera, memòria flash, etc... Es possible també accedir als diferents sensors i components de hardware que ofereix Android, com la càmera, sensors de moviment, etc...

En la següent figura es mostren les diferents capes que conformen el sistema operatiu d'Android:

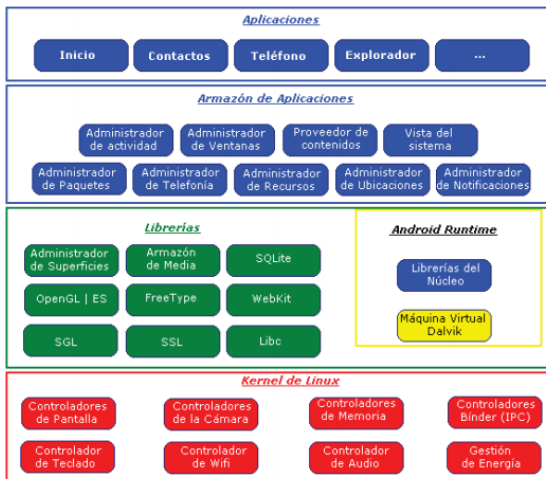


Fig. 2 – Nucli sistema operatiu Android

2.2.2 ENTORN DE DESENVOLUPAMENT MÒBIL

Per a poder desenvolupar amb Android, cal instal·lar el software que ofereix Google, Android Studio des de la direcció: <http://developer.android.com/sdk/index.html>

La versió més actual es la 4.4 Kitkat, però la nostra aplicació ha estat desenvolupada per a la versió 4.0 endavant per tal d'abastar més usuaris.

Android Studio disposa també d'un Emulador d'Android. Bàsicament, es un dispositiu amb Android virtual. Serveix principalment per debugar la teva aplicació. Disposada d'un tel·lat virtual i els diferents botons que pot tenir un dispositiu

amb Android.

2.3 ARQUITECTURA WEB

Per a la part web s'ha utilitzat el patró de disseny Model Vista Controlador (MVC), que s'explica en la secció 2.3.1 més detalladament.

Cal destacar el llenguatge de programació utilitzat per desenvolupar l'aplicació web ha estat PHP. Es tracta del llenguatge script de servidor més popular.

Avui dia es pot instal·lar mòduls per interpretar PHP en gairebé tots els servidors d'aplicacions web. Especialment PHP té una gran relació amb Apache.

És un llenguatge basat en C i en Perl, que s'ha dissenyat pensant a donar-li la màxima versatilitat i facilitat d'aprenentatge, per sobre de la rigidesa i coherència semàntica.

Els avantatges d'utilitzar PHP són els següents:

- **Multi plataforma;** Es pot executar en qualsevol plataforma solament instal·lant l'interpret de PHP.
- **Obert i gratuït;** Pertany al programari llicenciat com GNU, llicència de Linux. La comunitat és l'encarregada de realitzar les millores en el codi.
- **Gran comunitat d'usuaris;** Comunitat d'usuaris àmplia i dinàmica.
- **Apache, MySQL;** Apache és el servidor més utilitzat en l'actualitat. MySQL és el servidor de base de dades relacional més popular en Internet, propietat de Oracle i de codi d'obert. L'estreta relació entre PHP i MySQL fa que treballin perfectament entre ells.
- **Extensions;** Disposa d'un gran numero d'extensions que permeten ampliar les capacitats del llenguatge, facilitant la creació d'aplicacions web complexes.
- **Fàcil de programar;** La seva corba d'aprenentatge no és tan elevada com pot ser la d'ASP, encara que existent diferents opinions sobre aquest tema.

A continuació es mostra una comparativa entre els llenguatges més populars avui dia:

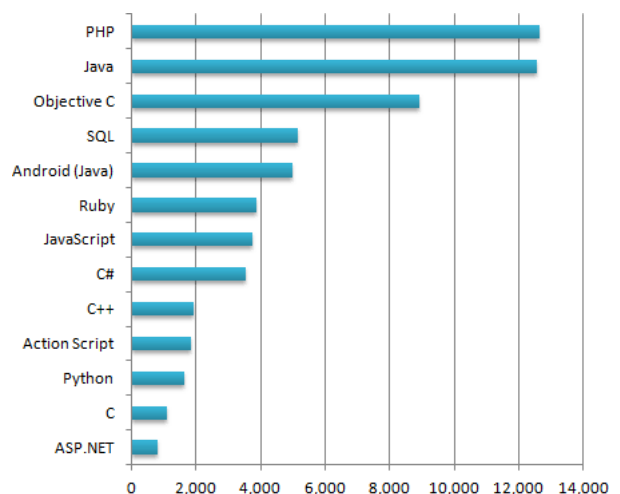


Fig. 3 – Comparativa de llenguatges de programació

PHP i Java són els mes estesos per la comunitat, seguit de Objective-C propietat d'Apple.

2.3.1 PATRÓ DE DISSENY MVC

El patró MVC és una arquitectura de disseny programari per separar els components d'aplicació en tres nivells, interfície d'usuari, lògica de control i lògica de negoci.

És una especialització d'un model de capes, amb la diferència que s'utilitza per a entorns web com a patró per excel·lència.

Exemple: Struts, Spring, ASP.NET MVC... Etc.

La següent figura mostra el diagrama bàsic d'un patró MVC;

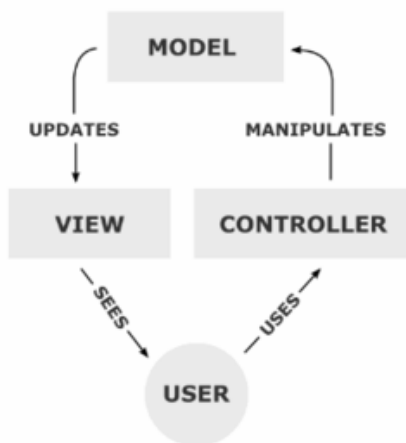


Fig. 4 – Diagrama Model Vista Controlador

Les diferents responsabilitats de les capes MVC es detallen a continuació:

- **Model:** És la capa encarregada d'encapsular tota la lògica de negoci de la nostra aplicació. Aquesta capa es pot subdividir en varies:
 - Lògica de negoci: Conté classes o diverses per constituir el referent a la capa d'aplicació, s'encarrega d'atendre a les peticions dels controladors i així donar una resposta d'acord amb el rebut.
 - Capa de dades: S'encarrega de gestionar tota la interconnexió amb el SGBD, així mateix, pot contenir un gestor ORM per al seu aprofitament màxim i millor manteniment. Solament es comunica amb la lògica de negoci.
 - Helpers: Anomenats "ajudants" recolzen tant al controlador com a la vista per fer més lleugeres algunes tasques.
- **Controlador:** És l'eix central de la nostra arquitectura, encarregada de gestionar totes les peticions, validar els inputs rebuts i dirigir qualsevol petició de qualsevol tipus. Solament es comunica amb el model i respon a través de vistes.
- **Vista:** És la resposta de cada controlador i el que se li presenta a l'usuari final, es pot comunicar amb el controlador, els "helpers" i el model (en algunes ocasions).

2.3.2 FRAMEWORKS

Existeixen multitud de Frameworks que implementen l'arquitectura ja definida (Symfony2, CakePHP, Yii, Laravel o CodeIgniter entre altres), l'usuari només s'ha de preocupar d'implementar la lògica de l'aplicació.

En aquest projecte, s'ha implementat el Framework CakePHP, ja que té una gran comunitat d'usuaris i suport. Es pot descarregar la versió 2.4 de <http://cakephp.org/>. A més, aquest Framework, incorpora un mòdul PHPUnit per realitzar els test de l'aplicació de forma automàtica, així com un mòdul per a "code coverage", la quantitat de codi que es cobreix.

2.4 COMUNICACIÓ

Per tal de poder sincronitzar les dades entre mòbil i web, s'ha configurat un servidor central on guardar les nostres dades. Hem utilitzat una màquina virtual allotjada en Google Cloud Platform. A aquesta màquina es pot accedir de manera de segura per SSH amb una clau privada RSA. El sistema operatiu instal·lat es Ubuntu Server versió 14.

Com a Servidor s'ha utilitzat Apache 2.4, però existeixen altres alternatives d'Open Source com Nginx, Lighttpd, G-WAN o Cherokee que s'estan fent molt populars en els últims mesos, són també molt lleugers i ràpids.

Aquest servidor escolta peticions HTTP pel port 80 de la nostra aplicació tant web com mòbil.

A continuació, es mostra una gràfica comparativa amb els servidors web amb un processador de 8 cores.

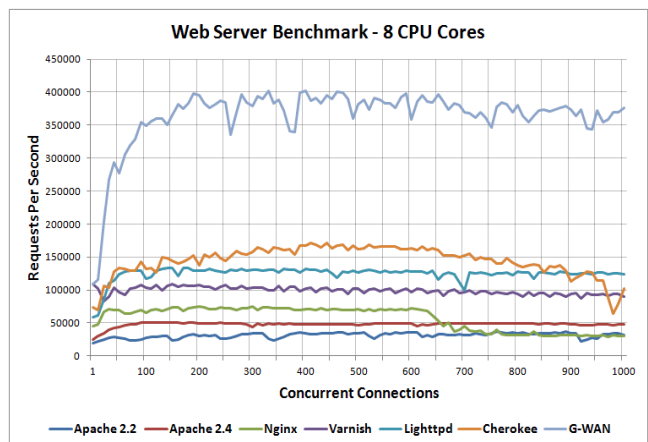


Fig. 5 – Rendiment servidors web

Es pot observar en la gràfica, que la càrrega de CPU de G-WAN és molt elevada respecte a la resta. Apache 2.2 i Apache 2.4 es mantenen estables durant tot el cicle de proves. Lighttpd i Cherokee augmenten una mica més respecte a la resta. S'ha de destacar, que G-WAN encara no és compatible amb PHP.

2.4.1 SERVIDOR DE BASE DE DADES MYSQL

Hem utilitzat MySQL ja que és un sistema gestor de bases de dades molt conegut i àmpliament utilitzat per la seva simplicitat i notable rendiment.

Encara que manca d'algunes característiques avançades disponibles en altres SGBD del mercat, és una opció atractiva tant per a aplicacions comercials, com d'entreteniment precisament per la seva facilitat d'ús i temps reduït d'engegada. Això i la seva lliure distribució en Internet sota llicència GPL li atorguen com a beneficis addicionals (no menys importants) comptar amb un alt grau d'estabilitat i un ràpid desenvolupament.

MySQL està disponible per a múltiples plataformes. No obstant això, les diferències amb qualsevol altra plataforma són pràcticament nul·les, ja que l'eina utilitzada en aquest cas és el client mysql-client, que permet interactuar amb un servidor MySQL (local o remot) en mode text.

MySQL ens ofereix grans prestacions com poden ser:

- Està desenvolupat en C/C++.
- La API es troba disponible en C, C++, Eiffel, Java, Perl, PHP, Python, Ruby i TCL.
- Està optimitzat per a equips de múltiples processadors.
- És molt destacable la seva velocitat de resposta.
- Es pot utilitzar com a client-servidor o incrustat en aplicacions.
- Compta amb un ampli conjunt de tipus de dades.
- Suporta múltiples mètodes d'emmagatzematge de les taules, amb prestacions i rendiment diferents per poder optimitzar el SGBD a cada cas concret.
- La seva administració es basa en usuaris i privilegis.
- Les seves opcions de connectivitat abasten TCP/IP, sockets UNIX i sockets NT, a més de suportar completament ODBC.
- Els missatges d'error poden estar en castellà i fer ordenacions correctes amb paraules accentuades o amb la lletra 'ñ'.

És altament fiable en quant a estabilitat es refereix.

3 L'ÀLBUM DIGITAL

L'àlbum digital està format per dues aplicacions desenvolupades amb l'objectiu d'interconnectar mòbil i web en un servidor central.

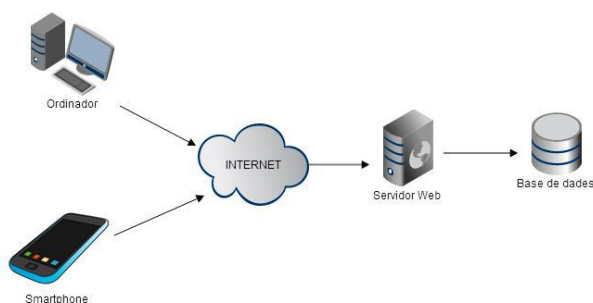


Fig. 6 – Estructura del sistema

En l'anterior figura es mostra l'estructura bàsica que té l'aplicació i els dispositius físics que interactuen entre sí.

- **Mòbil:** Es connecta al Servidor web per Internet mitjançant Wifi o connectivitat 3G. Una vegada connectat, el servidor respon les peticions del dispositiu i accedint a la base de dades, com: Llista d'àlbums, fotografies, etc...
- **PC:** Molt semblant al funcionament del Smartphone, però es connecta a una pagina web d'administració.
- **Servidor Web:** Dispositiu que s'encarrega de respondre les peticions HTTP rebudes pels altres dispositius. També es l'encarregat d'accedir a la base de dades per a realitzar les operacions (consultes) necessàries.
- **Base de dades:** Estructura de dades en MySQL que emmagatzema tota la informació dels usuaris.

Tant l'ordinador com l'Smartphone es connecten a internet mitjançant Wifi, Xarxa o 3G, s'obre una connexió amb el servidor central, seguidament s'envia la informació personal.

Les imatges s'envien mitjançant el protocol HTTP pel port 80, el servidor la rep, crea una nova instància de la imatge en la base de dades i la guarda en un directori segur.

3.1 PART WEB

L'aplicació web a estat desenvolupada amb PHP sota el framework CakePHP. De la mateixa manera que en l'aplicació mòbil, consta de una pantalla de login i un altre de registre on l'usuari ha d'introduir les seves dades personals. Una vegada registrat, té l'opció de pujar imatges des de l'ordinador, així com etiquetar imatges, crear àlbums digitals, modificar o eliminar tant àlbums com imatges.

Fig. 7 – Login web

En l'anterior figura podem veure la pantalla de login de l'usuari, on ha d'introduir el seu correu electrònic i un password de 6 caràcters.

A la part superior hi ha un botó per a registrar un nou usuari en el cas de que encara no estiguem registrats al sistema.

Una vegada hem entrat al sistema, veurem una interfície gràfica com la següent:

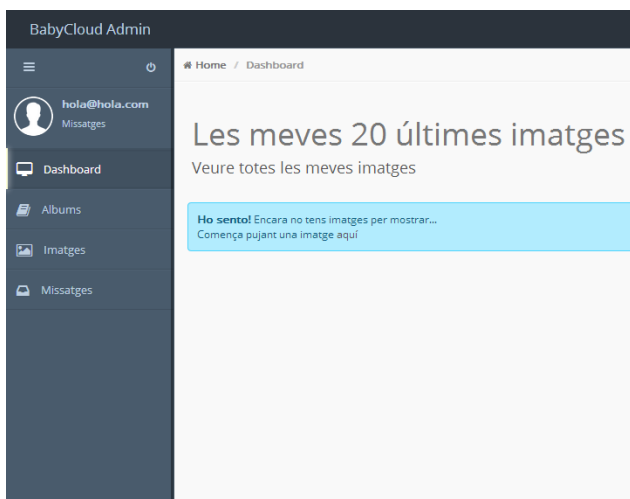


Fig. 8 – Pantalla d'inici web

Com veiem a la figura anterior, tenim un menú a la part esquerra amb les opcions principals; Àlbum, imatges i missatges. A la part d'àlbum podem veure els àlbums que tenim creats, podem crear de nous o podem modificar els que tenim. També es possible seleccionar un àlbum per a afegir noves imatges, com es veu a la següent figura:

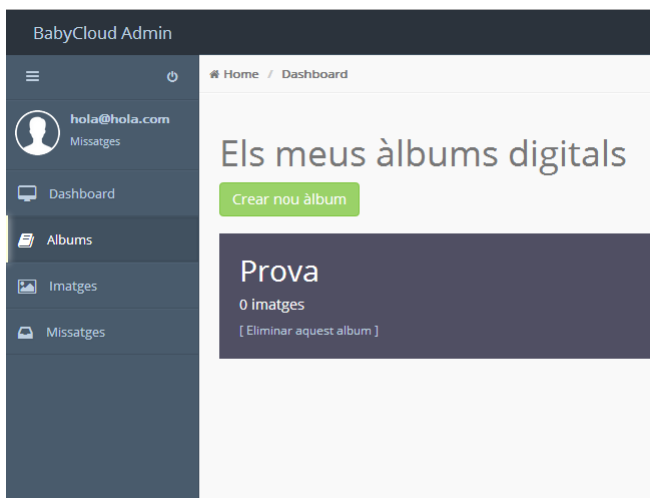


Fig. 9 – Llista d'àlbums

En la figura anterior, veiem un àlbum creat, podem eliminar-ho o seleccionar-ho per veure les imatges assignades a aquest àlbum.

L'opció d'imatges ens mostraria totes les imatges que tenim pujades al nostre compte.

A la part de missatges tenim un xat privat per a enviar missatges a altres usuaris del sistema.

3.2 PART MÒBIL

L'aplicació per a mòbil ha estat desenvolupada per a la plataforma Android, consta de 5 pantalles: Login, registre, àlbums, imatges i zoom d'imatge. L'usuari s'ha de registrar proporcionant un correu electrònic i un password de 6 caràcters com a mínim. Una vegada registrat, ja pot realitzar les operacions sobre l'aplicació, com: capturar una nova imatge, crear un nou àlbum, esborrar, etc...



Fig. 10 – Pantalles d'Android login i àlbums

En l'anterior figura podem veure la pantalla de login i la pantalla d'àlbums de l'usuari, juntament amb dos botons a la part superior: afegir nou àlbum i actualitzar la llista.

Una vegada es logueja l'usuari en l'aplicació mòbil, una classe anomenada JSONParser.java, es connecta al servidor central per validar les dades d'aquest usuari. Si les credencials són correctes, es porta a l'usuari a una pantalla on troba una llista amb els àlbums digitals creats. Té la possibilitat de crear un de nou, modificar un ja existent o directament eliminar-ho si ho desitja.

Una vegada se selecciona un àlbum, es mostra una pantalla amb les imatges que es troben dins d'aquest àlbum. Igual que en la pantalla d'àlbums, l'usuari pot assignar una etiqueta a la imatge o eliminar-la si ho desitja. Finalment, pot veure la imatge en una mida més gran i pot ampliar-la mitjançant els gestos tàctils que ofereixen els smartphones actuals.

Cal destacar, que s'ofereix la possibilitat de capturar noves imatges a través de la càmera que integra el dispositiu. Una vegada finalitzat el procés de captura, la imatge es sincronitza automàticament amb el servidor allotjat en el núvol.

4 RESULTATS

S'ha realitzat els test de qualitat tant de la aplicació web com de la aplicació mòbil.

Concretament, per a l'aplicació web s'ha utilitzat l'eina ja esmentada, PHPUnit. PHPUnit és un framework de proves unitàries per al llenguatge de programació PHP.

En les proves unitàries (al contrari que en les proves funcionals en les quals s'avalua l'aplicació com un tot) s'analitza una porció de codi que pugui ser analitzada de manera aïllada, com per exemple funcions i mètodes. Els quals després de passar-li uns paràmetres d'entrada hem d'obtenir altres paràmetres de sortida clarament definits.

4.1 PHPUNIT TESTING

Per poder realitzar les proves adients amb PHPUnit, s'ha duplicat la base de dades únicament per a fer Testing, ja que les proves unitàries també interactuen amb la base de dades i això podria influir en el funcionament de l'aplicació.

Les proves de Unit Testing amb PHPUnit han estat les següents:

- Test de Controlador
 - Es realitzen les proves de tots els controladors així com tots els seus mètodes.
- Test de Model
 - El mateix que controlador però ara amb els models.
- Code Coverage
 - Aquesta opció ens mostra el percentatge de codi cobert de la nostra aplicació.

En la següent figura es mostra el resultat del test dels controladors, es pot observar que s'han realitzat 157 comprovacions, 56 mètodes dels quals els 56 han passat el test.

```
Running ControllerTest
Running ControllerTest::testRedirectByCode
Running ControllerTest::testRedirectByMessage
56/56 test methods complete: 56 passes, 0 fails, 157 assertions and 0 exceptions.
Time: 0.26401305198669 seconds
Peak memory: 14,881,048 bytes
Run more tests | Show Passes | Enable Debug Output | Analyze Code Coverage
```

Fig. 11 – Diagrama Model Vista Controlador

En la següent figura es mostra les proves unitàries per als models.

```
Running All Model related class tests
Running CakeValidationSetTest
Running CakeValidationRuleTest
Running ModelReadTest
Running ModelWriteTest
16/16 test methods complete: 16 passes, 0 fails, 23 assertions and 0 exceptions.
Time: 0.13283627272297 seconds
Peak memory: 8,321,913 bytes
Run more tests | Show Passes | Enable Debug Output | Analyze Code Coverage
```

Fig. 12 – Diagrama Model Vista Controlador

Com es pot observar en l'anterior figura, s'han testejat 16 mètodes dels quals 16 han passat correctament el test. Així mateix 23 comprovacions que han estat correctes.

Com a reflexió comentar que PHPUnit es un framework molt elaborat que proporciona certes funcionalitats que altres frameworks de Testing no ofereixen. Es pot instal·lar fàcilment en un projecte amb PHP sota Linux o Windows.

4.2 JUNIT ANDROID SDK

Per a les proves de l'aplicació mòbil s'ha utilitzat el framework JUnit juntament amb el SDK d'Android.

JUnit és una llibreria Java que ens ajudarà en la realització dels nostres tests automàtics de prova.

JUnit proporciona, entre d'altres:

- Afirmacions per verificar resultats.
- Anotacions per definir mètodes de prova.
- Anotacions per definir mètodes addicionals pre i post prova.
- Controlar excepcions, temps d'espera.
- Parametrització de dades.

Els conceptes bàsics de JUnit són:

- Test class
 - Test method
 - Assert
 - Fixture method
- Test suite
- Test runner

Un Test Class és una classe java que contindrà un conjunt de mètodes de Test.

Amb la funció pròpia de JUnit anomenada `assertEquals`, comprovarem el resultat sigui correcte en una funció de prova.

5 CONCLUSIONS

S'han complert tots els objectius principals del projecte així com els requisits que es van plantejar a l'inici del projecte. El principal objectiu del projecte és desenvolupar dues aplicacions diferents però alhora vinculades en quant a tecnologia es refereix. D'una banda, s'ha desenvolupat una aplicació per a Internet on l'usuari pot interactuar amb el seu contingut. Crear, editar i eliminar són les accions que pot dur a terme mitjançant la interfície. D'altra banda, s'ha desenvolupat una aplicació per smartphone per a la plataforma Android. En aquest cas, l'usuari pot realitzar les mateixes accions que en el cas web, però a més a més, pot realitzar captures amb la càmera del dispositiu i així poder pujar-les al seu àlbum multimèdia.

M'he trobat amb alguns problemes a l'hora de desenvolupar algunes funcionalitats referents a la transmissió d'imatges de smartphone a servidor. Gràcies al hardware tan avançat dels dispositius mòbils, ens permet realitzar fotografies amb la nostra càmera amb molta grandària, al voltant de 3200x2400 mpx. La transmissió d'imatges entre el dispositiu i el servidor que allotja el contingut es realitzava amb força lentitud per culpa de les dimensions tan grans. Per falta de pressupost no he pogut contractar un servidor amb un maquinari potent que permeti agilitzar aquesta transmissió. L'única solució que he trobat ha estat re-dimensionar les imatges en el dispositiu a una mida considerable perquè el dispositiu mitjançant WiFi pugui enviar la imatge en un temps raonable.

Per a canvis futurs o pendants hi ha la possibilitat de poder sincronitzar amb comptes d'altres xarxes socials com Facebook, Instagram o Dropbox. Aquesta sincronització permetria poder carregar imatges en els nostres àlbums digitals, amb la utilització de OAuth 2.0 es podria dur a terme amb gran facilitat.

Com a ampliació al projecte comentar que es podria realitzar la mateixa aplicació per a plataformes Mac. Això ampliaria el número d'usuaris que podrien utilitzar aquest servei. Amb un Apple i una llicència de desenvolupar seriosa possible crear una aplicació simi-lar.

6 AGRAÏMENTS

Aquest treball no s'hauria pogut realitzar sense la col·laboració de moltes persones que m'han brindat la seva ajuda, els seus coneixements i el seu suport. Vull agrair-los a tots ells quant han fet per mi, perquè aquest treball tirés endavant de la millor manera possible.

Quedo especialment agraït a Aura Hernández Sabatè que m'ha ajudat i recolzat en tot moment. Ha corregit minuciosament aquest treball i m'ha donat la possibilitat de millorar-ho. Haig d'agrair-li els seus comentaris, suggeriments i les correccions amb les quals he pogut elaborar una adequada documentació de tot el treball realitzat.

7 BIBLIOGRAFIA

- [1] Ron Jeffries, What is Extreme Programming?, Novembre 2001
<http://xprogramming.com/xpmag/whatisxp>
- [2] Craig Strong, A Really Simple Explanation Of MVC, Març 2013
<http://c2.com/cgi/wiki?ModelViewController>
- [3] Lars Vogel. Android Development – Tutorial. Març 2014
<http://www.vogella.com/tutorials/Android/article.html>
- [4] Juan Treminio, Introduction to PHPUnit, Març 2013
<https://jtreminio.com/2013/03/unit-testing-tutorial-introduction-to-phpunit/>
- [5] Lars Vogel, Unit Testing with JUnit, Octubre 2013
<http://www.vogella.com/tutorials/JUnit/article.html>
- [6] Lars Vogel, Apache HttpClient, Agost 2012
<http://www.vogella.com/tutorials/HttpClient/article.html>
- [7] Bernard Kohan, PHP vs ASP.net Comparison, Agost 2010
<http://www.comentum.com/php-vs-asp.net-comparison.html>