



**Universitat Oberta  
de Catalunya**

[www.uoc.edu](http://www.uoc.edu)



# PROJECTE FINAL DE MÀSTER

## aLlum

**Màster Programari Lliure  
Administració Web i comerç electrònic**

**Autor:** Núria Garcia Torralva  
**Consultor UOC:** Javier Noguera  
**Consultor extern:** Xavier Tribó

**Gener 2012**

## Llicència

Aquesta memòria està sota una llicència de Creative Commons Reconeixement-NoComercial-CompartirIgual 3.0 Genèrica (CC BY-NC-SA 3.0).

Un resum de llicència és:

### Sou lliure de:

copiar, distribuir i comunicar públicament l'obra  
fer-ne obres derivades

### Amb les condicions següents:



**Reconeixement** — Heu de reconèixer els crèdits de l'obra de la manera especificada per l'autor o el llicenciador (però no d'una manera que suggereixi que us donen suport o rebeu suport per l'ús que feu l'obra).



**No comercial** — No podeu utilitzar aquesta obra per a finalitats comercials.



**Compartir Igual** — Si altereu o transformeu aquesta obra, o en genereu obres derivades, només podeu distribuir l'obra generada amb una llicència idèntica a aquesta.

### Entenent que:

**Renúncia** — Es pot **renunciar** a alguna d'aquestes condicions si obteniu el permís del titular dels drets d'autor.

**Domini Públic** — Aquesta llicència no afecta a la situació de l'obra o algun dels seus elements quan es trobi en el **domini públic**, segons la llei vigent aplicable.

**Altres drets** — Els drets següents no queden afectats de cap manera per la llicència:

- Els vostres drets de repartiment just o **ús just**;
- Els drets **morals** de l'autor;
- Drets que altres persones poden ostentar sobre l'obra o sobre l'ús que se'n fa, com per exemple drets **de publicitat** o privacitat.



## RESUM

aLlum és un projecte per desenvolupar una eina senzilla per aprofitar la ubiqüitat i facilitat d'ús dels telèfons mòbils.

L'idea va sortir de la necessitat que tenen els directors de fotografia quan van a les localitzacions de prendre nota de tots els productes que necessitaran per la il·luminació del rodatge i enviar ràpidament aquesta comanda per demanar pressupost.

Per portar a terme aquesta idea, el projecte és va dividir en dues parts principals.

La primera part és una aplicació client que permet realitzar comandes. La aplicació mostra una llista de tots els productes disponibles i l'usuari pot realitzar una comanda escollint quins productes vol i quantes unitats de cada producte necessita. Finalment pot demanar un pressupost de compra o lloguer d'aquest material a un proveïdor.

La segona part permet poder realitzar l'actualització del catàleg de productes que té el dispositiu, és a dir, poder establir una connexió des del terminal (client) a la base de dades del proveïdor (servidor) per actualitzar el catàleg dels productes que mostrarà per fer les comandes.

El programa mostra mitjançant un interfície intuïtiva com accedir a la llista de comandes que l'usuari té desades al seu terminal (perquè encara estan pendents de fer l'enviament) o com crear una nova comanda. Una comanda no és més que un nom i una selecció d'unitats de cadascun dels productes del catàleg. La comanda sempre mostra la llista sencera de productes disponibles.

Sempre que l'usuari vol pot demanar fer una sincronització del seu catàleg amb la base de dades que conté el catàleg del servidor i que és el que es va actualitzant amb nous productes o eliminant productes que són obsolets. Quan l'usuari demana aquesta sincronització el programa connecta amb el servidor per demanar la data del catàleg del servidor i compara aquesta data amb la data de la versió local al dispositiu. Si la data és la mateixa mostra un missatge a l'usuari indicant que no cal fer la sincronització perquè el terminal ja està actualitzat. En cas que la data del servidor hagi canviat sol·licita el nou catàleg i l'actualitza a la base de dades del dispositiu, actualitzant també la data de la nova versió.

La principal premissa del projecte era la utilització exclusiva de software lliure, per tant, el desenvolupament de la part del client (mòbil) està realitzat sobre el sistema operatiu *Android*.

## TAULA DE CONTINGUTS

PRESENTACIÓ DEL PROJECTE .....	6
Introducció .....	6
Objectius .....	6
Estructura de la memòria del projecte .....	7
ESTAT DE L'ART .....	9
Dispositius mòbils .....	9
Android .....	10
Accés a les dades .....	11
ENTORN DE TREBALL .....	12
Android .....	12
Eclipse .....	16
Emulador d'Android .....	16
Servidor HTTP Apache i MySQL .....	17
ANÀLISI .....	18
Requisits .....	18
Casos d'ús .....	18
Interfícies d'usuari .....	21
DISSENY .....	24
Arquitectura .....	24
Model de dades .....	28
Llicències .....	29
DESENVOLUPAMENT .....	31
Decisions .....	31
Disseny final de pantalles .....	33
IMPLANTACIÓ .....	35
CONCLUSIONS .....	39
BIBLIOGRAFIA .....	41
ANNEX1. SCRIPT DE CREACIÓ DEL CATÀLEG .....	42
ANNEX2. VERSIO.PHP .....	43
ANNEX3. PRODUCTES.PHP .....	44

## TAULA D'IL·LUSTRACIONS

Imatge 1: Comparativa quota de mercat dels smartphones .....	10
Imatge 2: Entorn de desenvolupament.....	12
Imatge 3: Cicle de vida d'una Activity .....	13
Imatge 4: Estructura d'un projecte .....	15
Imatge 5: Emulador Android.....	16
Imatge 6: Arquitectura client-servidor .....	18
Imatge 7: Cas d'ús alta/modificació comanda .....	19
Imatge 8: Cas d'ús desar comanda.....	19
Imatge 9: Cas d'ús enviar comanda .....	20
Imatge 10: Cas d'ús sincronització catàleg .....	20
Imatge 11: Maqueta en <i>wireframes</i> de les interfícies .....	22
Imatge 12: Patró d'arquitectura MVC .....	24
Imatge 13: Diagrama de classes.....	25
Imatge 14: Diagrama de seqüència de la inicialització de l'aplicació .....	26
Imatge 15: Diagrama de seqüència de la creació d'una comanda .....	27
Imatge 16: Diagrama de seqüència de l'enviament d'una comanda .....	28
Imatge 17: Diagrama E-R del dispositiu .....	28
Imatge 18: Estructura de dades de la llista de comandes .....	32
Imatge 19: Pantalla inicial .....	33
Imatge 20: Pantalla del llistat de comandes .....	33
Imatge 21: Pantalla comanda.....	34
Imatge 22: Pantalla producte .....	34
Imatge 23: Creació de la base de dades.....	35
Imatge 24: Creació de les taules.....	36
Imatge 25: Visualització de les taules .....	36
Imatge 26: Execució de l'script d'inicialització .....	37
Imatge 27: Resultat de la inicialització .....	37

## PRESENTACIÓ DEL PROJECTE

---

### Introducció

El projecte *aLlum* deu el seu nom al seu homònim *iLuz* creat per *iPhone* d'Apple [1].

El projecte inicial va néixer de la necessitat dels directors de fotografia quan anaven a les localitzacions de poder prendre nota de la il·luminació que necessitarien pel rodatge i enviar-la sense cap demora als proveïdors per veure la seva disponibilitat i cost.

El projecte *iLuz* consisteix en mostrar un catàleg fix en el dispositiu mòbil de productes del sector de la il·luminació i permet la selecció d'unes determinades unitats de cada producte. Un cop finalitzada la selecció genera un text amb el recull dels productes seleccionats i les unitats de cada producte i permet l'enviament d'aquest text per *sms* (servei de missatges curts) a un proveïdor.

El projecte *aLlum* canvia el sistema operatiu de iOS [2] a *Android* i desenvolupa des de zero la nova aplicació, però a més, pretén anar una mica més enllà:

- Permet desar les feines al dispositiu per no tenir que crear-les i enviar-les en el mateix moment.
- Realitza la comanda d'enviament per correu electrònic.
- El catàleg no és fix, l'usuari pot actualitzar el catàleg de productes amb la base de dades del proveïdor sota demanda.

El proveïdor mantindrà actualitzats els seus productes a la base de dades del servidor i és cada client, sense necessitat de cap actualització de programari, qui s'encarrega de tenir la seva còpia local del catàleg actualitzada.

### Objectius

L'objectiu principal és completar el projecte amb la major productivitat i qualitat possible, i tenint sempre present que:

*“Només el client determina si s'han complert els objectius i si el sistema és acceptable”.*

Partint d'aquesta premissa els principals objectius del projecte són:

- Poder disposar d'una aplicació, com a mínim, amb les mateixes funcionalitats que *iLuz* però per *Android*.
- Que un perfil d'usuari no tècnic però si acostumat a l'ús d'un *smartphone* (telèfon intel·ligent) sigui capaç d'utilitzar l'aplicació sense dificultats.
- Poder realitzar una comanda de productes seleccionant les unitats concretes per a cada producte.
- Poder desar o enviar immediatament a un proveïdor la comanda generada per demanar pressupost de compra o lloguer.
- Que el catàleg no sigui fix o calgui una actualització del programari per actualitzar-lo sinó que el client pugui demanar en qualsevol moment la sincronització del catàleg amb la base de dades del proveïdor i disposar automàticament de tots els productes nous.
- Realitzar un projecte de Software Lliure per poder permetre que l'aplicació pugui ser millorada per la comunitat d'usuaris que puguin estar interessats en ell.pro

## Estructura de la memòria del projecte

La memòria d'aquest projecte està estructurada amb les següents seccions:

- Presentació del projecte: és la secció actual que inclou la introducció, els objectius principals del projecte i una visió general dels continguts.
- Estat de l'art: estudi del mercat actual dels aspectes tècnics relacionats amb el projecte com són els dispositius mòbils, Android i la connectivitat entre el terminal mòbil i un servidor.
- Entorn de treball: descripció del *host* (màquina on s'ha fet el desenvolupament) i del *target* (terminals on s'executa l'aplicació)
- Anàlisi: especificació detallada del sistema que es vol implementar amb els casos d'ús i les interfícies d'usuari.
- Disseny: especificació del disseny tecnològic de la solució.

- Desenvolupament: detalla el desenvolupament del client del dispositiu mòbil Android així com la connectivitat amb el servidor.
- Implementació: detall de les classes i components bàsics desenvolupats.
- Conclusions: resum de l'experiència en l'elaboració del projecte i dels resultats obtinguts, comparant aquests resultats amb els objectius marcats inicialment. També detalla les possibles millores i línies futures del projecte.
- Bibliografia: recull de les referències de textos citats.



## ESTAT DE L'ART

---

### Dispositius mòbils

Les noves plataformes mòbils han obert noves línies per aplicacions informàtiques que fins fa molt poc estaven limitades a ordinadors personals.

La disponibilitat de les tecnologies sense fil i la proliferació dels dispositius mòbils fan que els usuaris d'aquests dispositius puguin accedir a Internet en qualsevol moment i des de qualsevol lloc, per tant, la realització de funcionalitats mitjançant els dispositius mòbils creix dia a dia.

S'entén per dispositiu mòbil un sistema de mida reduïda que típicament disposa d'una pantalla de visualització amb una entrada tàctil i/o un mini teclat.

Entre els tipus de dispositius mòbils existents estan:

- **Carputer:** computador mòbil instal·lat i utilitzat en l'interior d'un automòbil. Combinen tecnologies de sistemes de so, lectors de DVD, software de processament de paraules, PDA, telèfons intel·ligents, Bluetooth, navegació per GPS, etcètera.
- **PDA (Assistent Digital Personal):** és un dispositiu mòbil petit (*"de butxaca"*) que funciona com un gestor de informació personal amb agenda, adreces de contacte, calendari, etc. Les PDA actuals també ofereixen la possibilitat de la connexió a Internet, pantalles de visualització adequades per la navegació Web, capacitats d'àudio per habilitar-les com a telèfons mòbils i tecnologia de pantalles tàctils.
- **Telèfons intel·ligents o smartphones:** es pot considerar com una PDA amb les funcionalitats d'un telèfon mòbil integrades. Utilitzen sistemes operatius adaptats per a mòbils i tenen la característica clau de que poden ser instal·lades en el dispositiu aplicacions addicionals desenvolupades tant pel fabricant, com per l'operador com per qualsevol desenvolupador de software.
- **Tablet PC:** és un ordinador mòbil complert (superior a una PDA) que integra una pantalla tàctil com a principal mitjà de comunicació amb l'usuari. La característica principal es no tenir teclat físic sinó que aquest ha estat substituït per un teclat virtual que apareix a la pantalla o per un llapis digital.

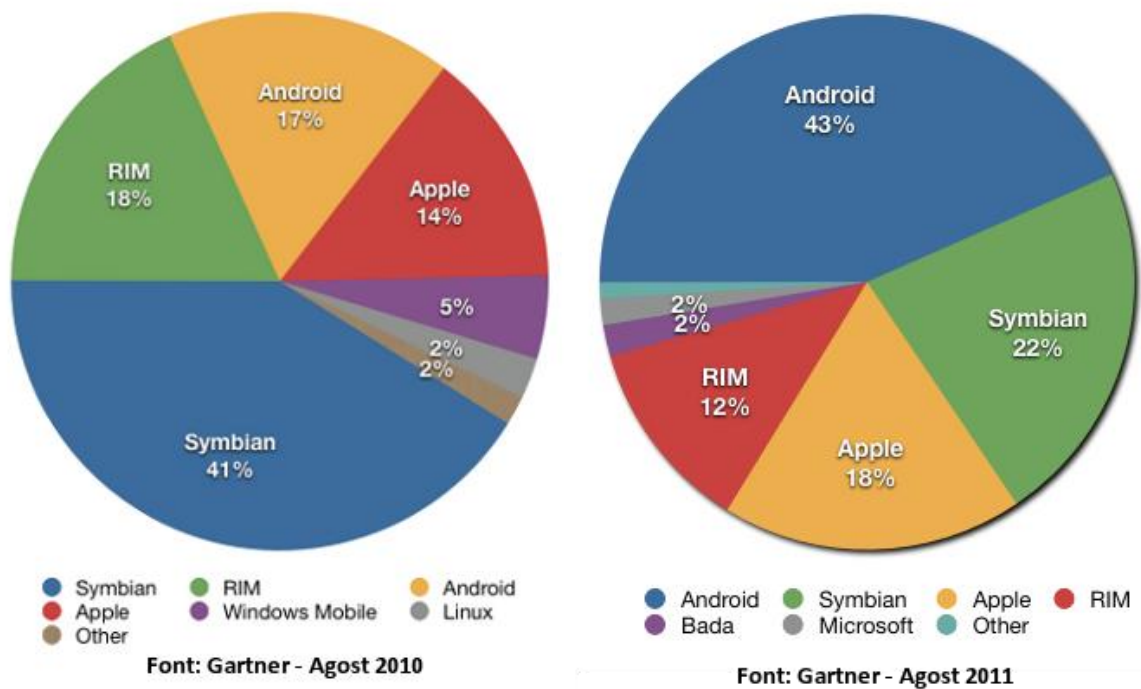
Les connexions en els dispositius mòbils són sense fils i per tant estan subjectes a les limitacions en la connectivitat degudes principalment a l'alta latència i la connexió intermitent.

## Android

### Què és Android? [3]

*“Android is a software stack for mobile devices that includes an operating system, middleware and key applications. The [Android SDK](#) provides the tools and APIs necessary to begin developing applications on the Android platform using the Java programming language.”*

Existeixen molts sistemes operatius per tota la gama de dispositius mòbils existents, però Android és un dels sistemes operatius per mòbils amb més creixement des de la seva sortida. La següent gràfica mostra una comparativa de la quota de mercat dels smartphones a l'agost del 2010 i a l'agost del 2011:



Imatge 1: Comparativa quota de mercat dels smartphones

## Accés a les dades

Per guardar dades, és a dir, fer els objectes persistents en terminologia de programació orientada a objectes, Android ofereix varies possibilitats. Algunes són específiques d'Android i pensades fonamentalment per guardar configuracions de cada aplicació (és el cas de les Shared Preferences i dels Content Providers). D'altres adapten conceptes més tradicionals, com les bases de dades, a l'entorn mòbil.

Per accedir durant l'execució de l'aplicació a una base de dades tenim dues possibilitats: accedir a una base de dades local, que haurà d'estar actualitzada o bé connectar-nos remotament a un servidor.

En sistemes de connectivitat limitada com és el mòbil la segona alternativa és arriscada tot i que factible des d'Android mitjançant el JDBC (Java DataBase Connectivity).

Per a la primera alternativa Android ens proporciona un conjunt de classes en Java per treballar en el SGBD (Sistema de Gestió de Bases de Dades) SQLite. Cal, però, resoldre el problema de l'actualització. Novament disposem de diverses alternatives, la més senzilla de les quals és obligar a l'usuari a actualitzar l'aplicació cada cop que volem actualitzar la base de dades. Una solució més elegant i flexible és sincronitzar la base de dades resident al mòbil amb la base de dades resident al servidor. L'avantatge de la sincronització vers la connexió via JDBC és que aquesta es pot fer a petició de l'usuari, en una sola transacció quan la connectivitat sigui l'adequada.

Des de l'aparició del rfc (request for comments) 4627 de JSON (Javascript Object Notation), aquest format de dades d'intercanvi ha anat guanyant terreny gradualment com a solució als problemes de sincronització deixant en desús alternatives com XML (Extensible Markup Language ) en general i SOAP (Simple Object Access Protocol) en particular. D'aquesta manera, l'aplicació Android és connecta a un webservice del servidor, programat per exemple en llenguatge PHP, que és el que fa la consulta a la base de dades del servidor, genera la resposta en format JSON i l'envia al client Android.

## ENTORN DE TREBALL

---

### Android

Android és una plataforma oberta pel desenvolupament d'aplicacions propietat de Google. Està construïda sobre la versió 2.6 del kernel de Linux, les aplicacions es programen en Java i són executades en la màquina virtual Dalvik especialment dissenyada per aquesta plataforma amb compilació en temps d'execució.

La llicència de distribució escollida per Android ha estat Apache 2.0 i GNU GPL, per tant, és una llicència de software lliure.

Un resum de les característiques més destacables seria:

- Nucli: Linux
- Màquina virtual: Dalvik
- Gràfics: API gràfica OpenGL ES i motor gràfic SGL.
- Base de dades relacional: SQLite
- Entorn de desenvolupament: programació en Java, emulador de dispositius i plugin per Eclipse.
- Executables: .apk (Android application package)
- Software lliure

Podem disposar d'un entorn de desenvolupament per Android amb:

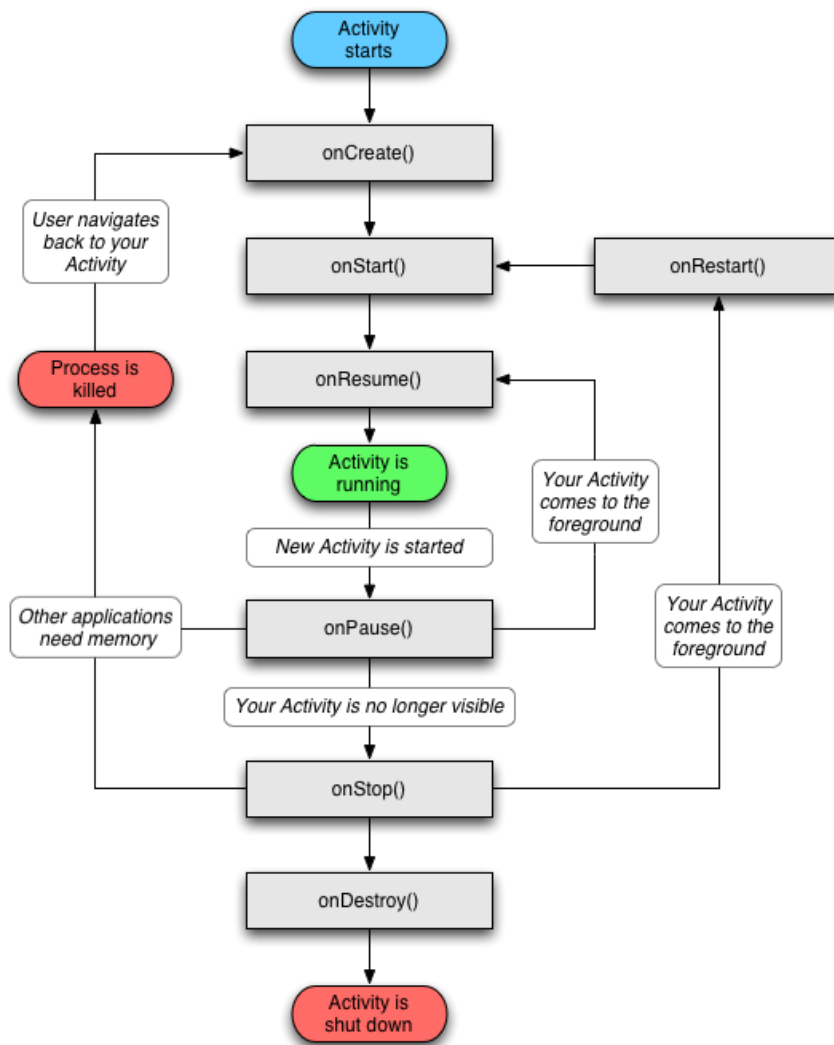


**Imatge 2: Entorn de desenvolupament**

Components principals d'una aplicació Android.

- Activity

Són el component principal de la interfície gràfica (anàloga a una *finestra* en altres llenguatges). S'implementen derivant la classe Activity i cadascuna té un cicle de vida propi:



Imatge 3: Cicle de vida d'una Activity

Està formada per *Views* (vistes) que són els components bàsics de la interfície (*controls* en altres llenguatges), podem ampliar la funcionalitat d'aquests controls bàsics o bé crear controls nous personalitzats.

- Service

Són components sense interfície gràfica que se executen en segon pla o *background* i que normalment duren un període llarg de temps (exemple: un gestor de descàrrega). Se les pots assignar diferents prioritats.

- Content Provider

Les aplicacions poden guardar la informació en fitxers o bases de dades (SQLite) però també es pot compartir la informació entre aplicacions. Aquesta classe implementa un conjunt de mètodes estàndard perquè les aplicacions que comparteixen dades pugin emmagatzemar i extreure la informació.

- Intent

És l'element bàsic de comunicació entre components, és a dir, és un objecte missatge de sol·licitud per realitzar una acció.

- Android Manifest

Tota aplicació Android ha de tenir un fitxer XML anomenat *AndroidManifest* que descriu els aspectes principals de l'aplicació com la seva identificació, els seus components o els permisos necessaris per la seva execució.

L'estructura de fitxers d'una aplicació Android és com es detalla a continuació:



**Imatge 4: Estructura d'un projecte**

- **src**: és on es troben els fitxers .java, és a dir, conté tot el codi font de l'aplicació.
- **gen**: conté codi generat automàticament al compilar el projecte i que són de control dels recursos de l'aplicació. Un dels fitxer més importants és el R.java que conté la classe R amb constants amb els identificadors dels recursos de l'aplicació inclosos en la carpeta de recursos (res).
- **assets**: és la carpeta on s'emmagatzemen fitxers auxiliars de l'aplicació. La diferència amb els recursos inclosos en la carpeta /res es que per aquests recursos es genera un identificador en la classe R i en canvi per aquests de la carpeta *assets* no.
- **res**: conté els fitxers amb els recursos del projecte com imatges o textos. Els recursos es van distribuïnt en diferents carpetes segons la seva tipologia.

- **AndroidManifest.xml**: conté la definició en XML dels principals aspectes de l'aplicació com el nom, la versió, la icona, els components (pantalles) o els permisos necessaris per la seva execució.

## Eclipse

Eclipse [4] és un entorn de desenvolupament integrat (IDE) multiplataforma desenvolupada per una comunitat de software lliure. Utilitza mòduls (*plug-in*) per proporcionar la seva funcionalitat característica que el diferencia d'altres entorns monolítics.

La definició que dona el projecte Eclipse a propòsit del seu software és: "*una especie de herramienta universal - un IDE abierto y extensible para todo y nada en particular*" [5].

Per treballar amb Android cal la instal·lació d'un d'aquests mòduls: ADT (Android Development Tool) Plugin for Eclipse [6].

## Emulador d'Android

L'Android SDK inclou un emulador virtual que s'executa en l'ordinador permeten així testear i depurar l'aplicació sense necessitat d'utilitzar un dispositiu físic. L'Android Virtual Device (AVD) conté les configuracions dels diferents emuladors.



**Imatge 5: Emulador Android**



## Servidor HTTP Apache i MySQL

El servidor web Apache és un software multiplataforma de codi obert que implementa el protocol HTTP. Permet incloure suport per diferents llenguatges de programació de servidor, entre ells el PHP que és utilitzat en aquest projecte.

MySQL és un sistema de gestió de base de dades relacional i multiusuari que utilitza el llenguatge SQL i que té suport de forma nativa per part del llenguatge PHP.

## ANÀLISI

---

### Requisits

Els requisits principals de l'aplicació aLlum són:

Client Android:

Poder crear i modificar una comanda.

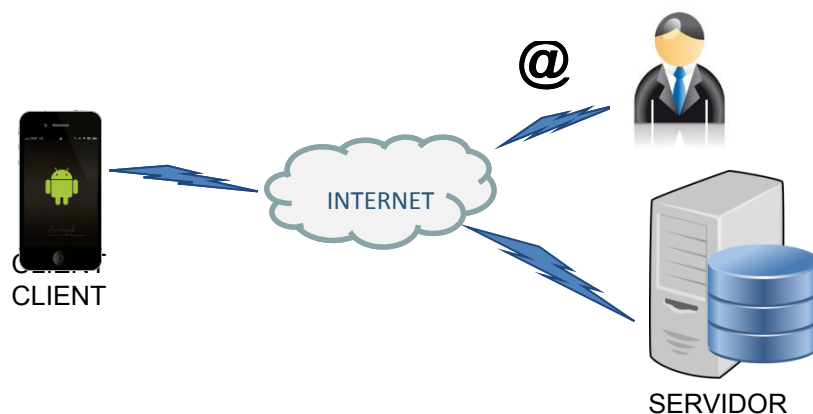
Poder desar una comanda localment al dispositiu.

Poder enviar per correu electrònic una comanda a un proveïdor.

Poder actualitzar la base de dades de productes local.

Servidor:

Poder atendre i donar resposta a la sol·licitud del catàleg de productes.



**Imatge 6: Arquitectura client-servidor**

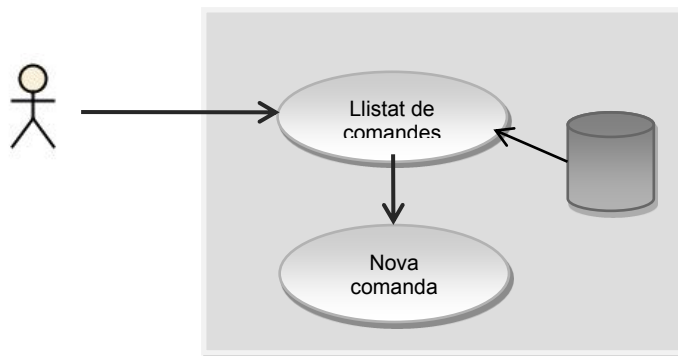
### Casos d'ús

A continuació es detallaran els casos d'ús del sistema amb les interfícies utilitzades en cada cas i la interacció amb l'usuari:

### Alta/modificació de comanda

Per crear o modificar una comanda cal accedir primerament al llistat de comandes. Aquest llistat sempre presentarà com a primera opció “Nova comanda”, seguit per totes aquelles comandes locals al dispositiu que encara no hagin estat enviades.

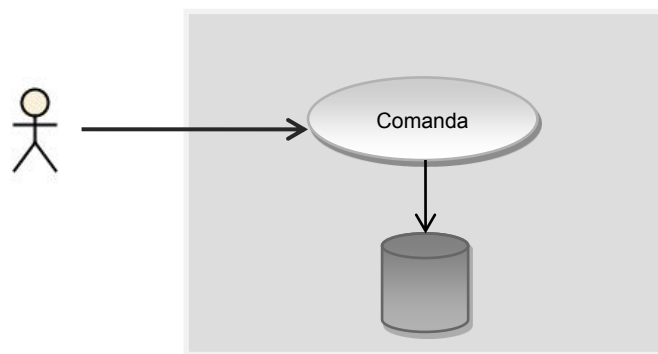
Un cop seleccionat un element del llistat (l’opció de nova comanda o la comanda que es vol modificar) s’accedirà a la pantalla de les dades de la comanda on cal introduir un nom si es tracta d’una nova comanda o mostrarà el nom donat a la comanda anteriorment si es tracta d’una modificació (però aquest nom de comanda també és modificable) i on tindrem disponible la llista de productes existents actualment a la base de dades local del dispositiu per seleccionar quins d’aquests productes volem afegir o modificar a la comanda i quantes unitats de cada producte.



**Imatge 7: Cas d'ús alta/modificació comanda**

### Desar comanda

Per desar una comanda des de la pantalla de la comanda cal prendre el botó de desar i això farà que les dades de la comanda actual es desin a la base de dades del dispositiu.



**Imatge 8: Cas d'ús desar comanda**

### Enviar comanda

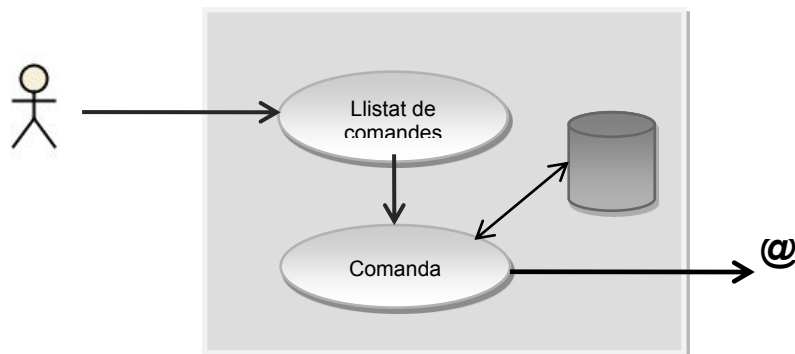
Els enviaments de les comandes es fan per correu electrònic, per tant, cal donar l'adreça del proveïdor destinatari.

Per fer l'enviament també cal detallar si el pressupost que s'està demanant és de compra o és de lloguer.

El cos del correu que s'envia es genera automàticament amb el llistats dels productes i les unitats de cadascun dels productes seleccionats prèviament a la comanda.

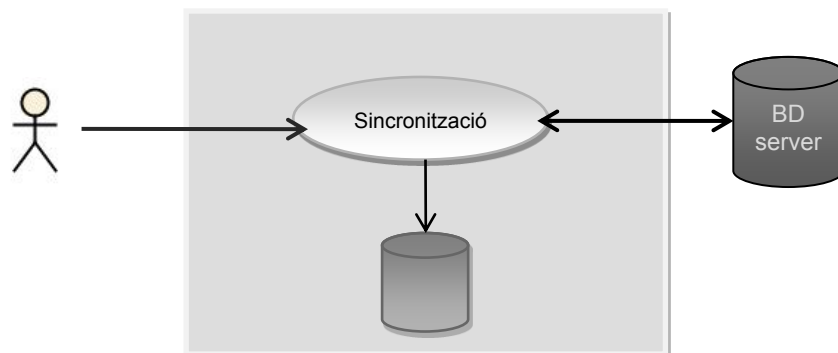
El programa permet afegir al correu una capçalera i un peu específic per a cada comanda.

Un cop s'ha enviat la comanda aquesta s'esborra de la base de dades i per tant també deixa de aparèixer a la llista de comandes locals al dispositiu.



**Imatge 9: Cas d'ús enviar comanda**

### Sincronització del catàleg



**Imatge 10: Cas d'ús sincronització catàleg**

## Interfícies d'usuari

### Perfil d'usuaris

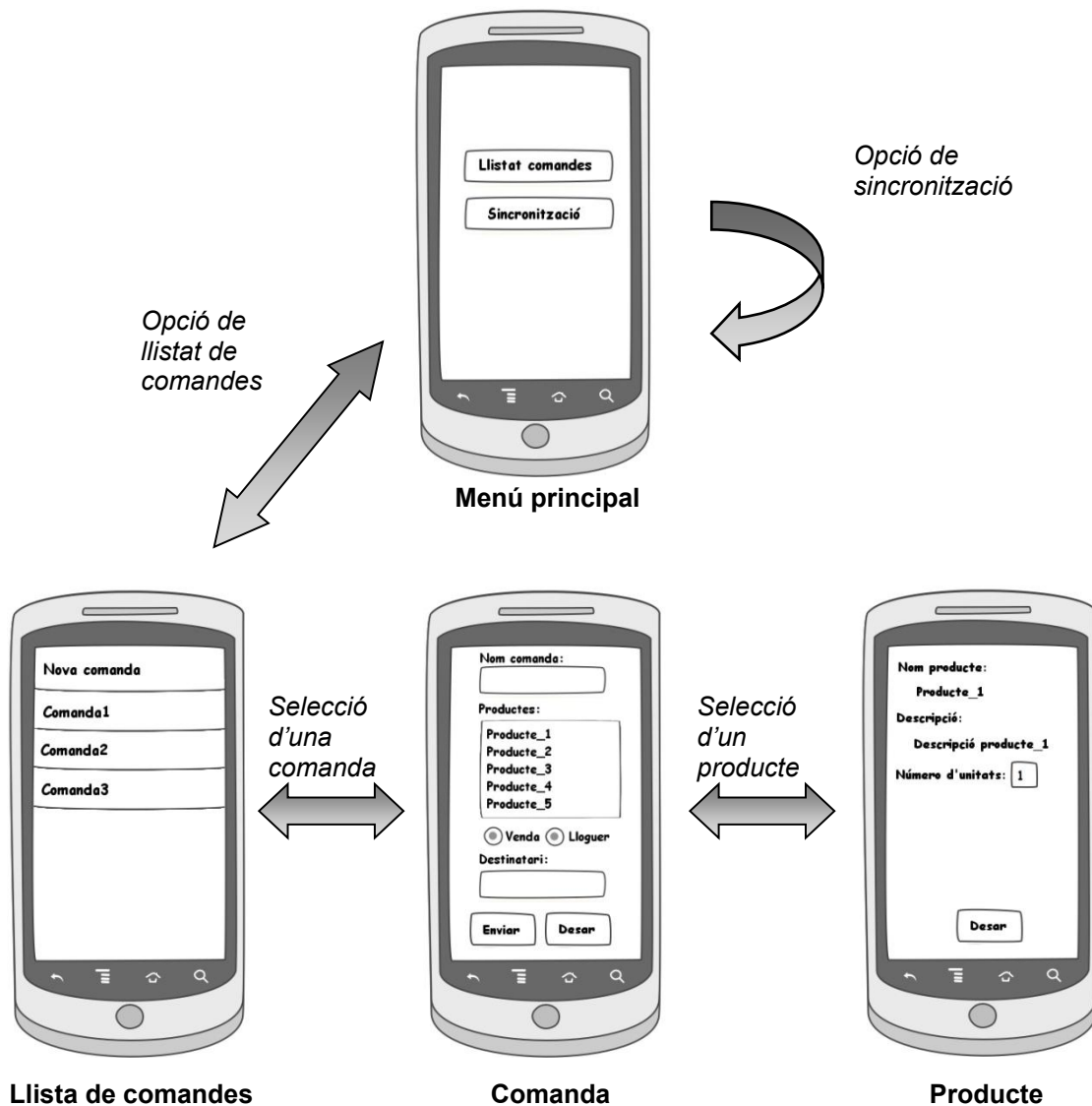
El projecte aLlum està format per dues parts i per tant per dos perfils d'usuari molt diferents.

El frontend és una aplicació per terminal mòbil dissenyada per un perfil de director de fotografia o similar. Es tracta d'un perfil gens tècnic (informàticament parlant) i molt poc acostumat a la utilització de cap programari.

El backend del servidor serà mantingut per un administrador de sistemes per tant és un perfil totalment tècnic.

### Interfícies

Abans de la descripció individual de cada interfície, la següent gràfica mostra una maqueta en *wireframes* (representació esquemàtica sense elements gràfics) per poder tenir una visió global de les interaccions entre interfícies:



**Imatge 11: Maqueta en wireframes de les interfícies**

### Menú principal

El menú principal és la primera pantalla que apareix quan s'executa l'aplicació i consisteix en un accés a les opcions del llistat de comandes o la sincronització del catàleg de productes del servidor.

## Llistat de comandes

La interfície del llistat de comandes mostra una llista amb totes les comandes locals als dispositiu que encara no han estat enviades. Sempre apareix com a primera opció del llistat l'opció de "Nova comanda" que és la que permet iniciar una comanda des de zero i que serà l'únic element del llistat en la primera execució de l'aplicació.

## Comanda

La pantalla de la comanda mostra totes les dades de la comanda: el nom i les unitats de cada producte seleccionades són les dades d'una comanda que es poden desar en la base de dades local del dispositiu. L'opció de compra o lloguer així com l'adreça de correu del destinatari són dades que només són utilitzades en l'enviament de la comanda i per tant no es desen.

La comanda mostra en tot moment una llista amb la totalitat dels productes existent a la base de dades del dispositiu. Quan l'usuari selecciona un element d'aquesta llista (un producte) accedeix automàticament a la pantalla del producte.

Si es modifica o s'envia una comanda en tornar a la pantalla anterior la llista de comandes ja mostrarà reflectit aquests canvis.

## Producte

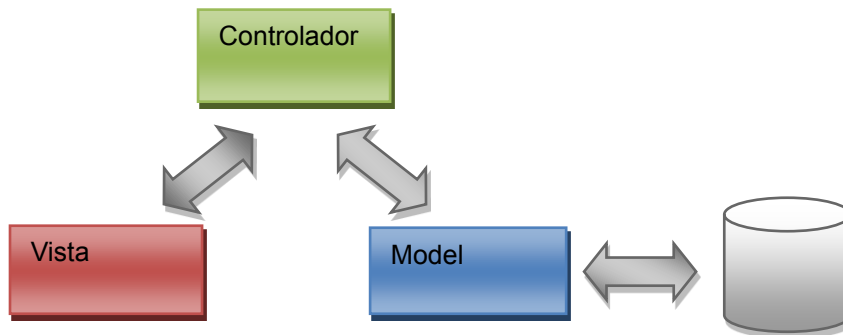
La pantalla del producte mostra el nom i la descripció del producte seleccionat. És en aquesta pantalla on l'usuari selecciona el número d'unitats d'aquest producte que vol afegir a la comanda actual de l'aplicació.

## DISSENY

---

### Arquitectura

El sistema segueix el patró d'arquitectura MVC (Model View Controller) adaptat per Android. L'objectiu d'aquest patró es separar les dades (*model*), la lògica del negoci (*controller*) i la interfície d'usuari (*view*):



**Imatge 12: Patró d'arquitectura MVC**

#### *View*

És la capa de la interfície d'usuari, per tant, en aquest projecte Android estarà formada per les classes Activity del sistema: el menú principal, la llista de comandes, la comanda i el producte. Els principals objectius d'aquesta capa són la presentació i la interacció amb l'usuari. Aquesta capa no accedeix mai directament a la capa de dades sinó que sempre ho fa a través del controlador.

#### *Controller*

És la capa pont entre la presentació i les dades. Aquesta capa és on està tota la lògica del negoci de l'aplicació.

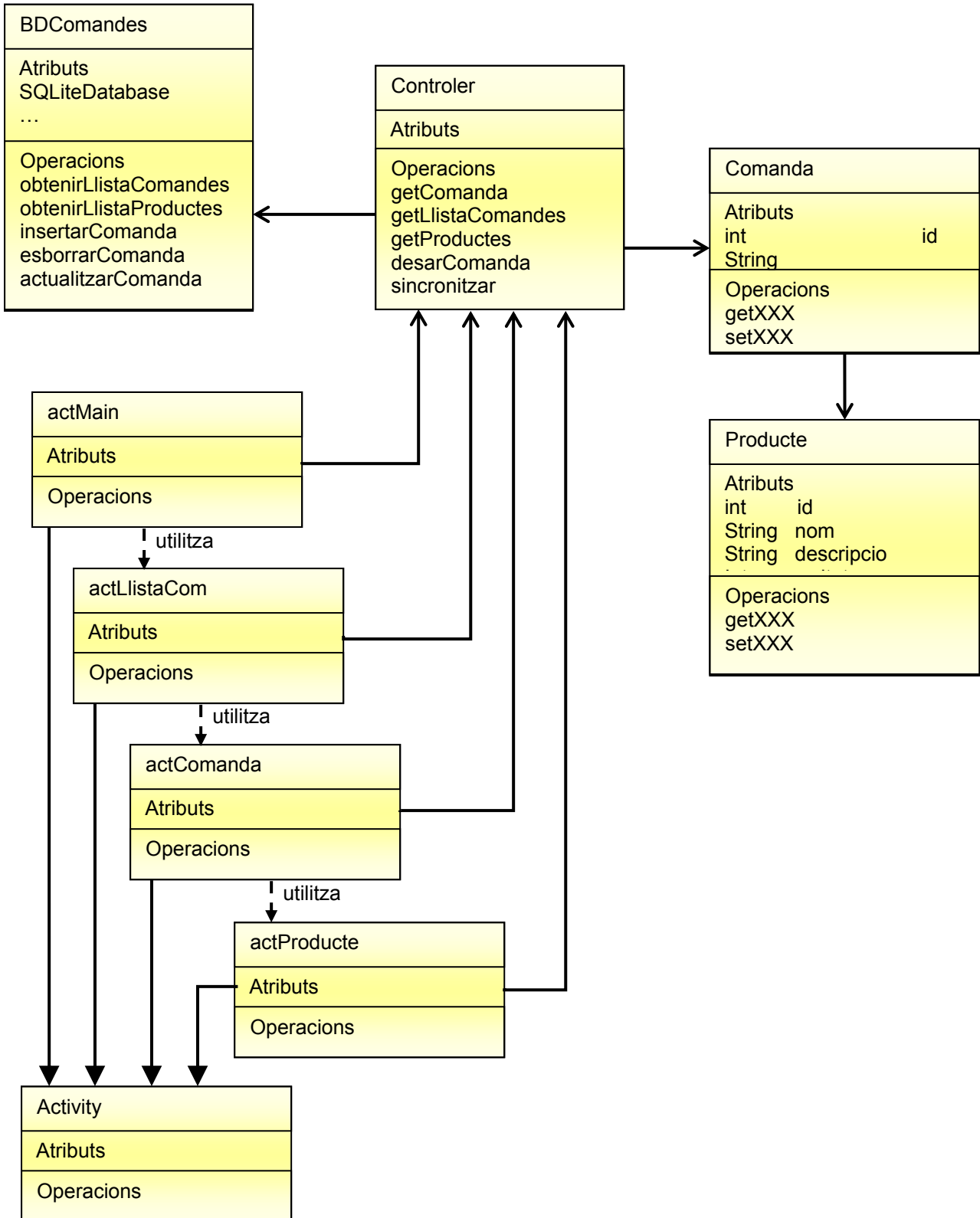
#### *Model*

És la capa de dades, i per tant, l'única capa que accedeix a la base de dades.

### Diagrames de classes

El diagrama de classes de l'aplicació en UML (Uniform Modeling Language) s'ha fet seguint les referències [7] i [8]:





Imatge 13: Diagrama de classes

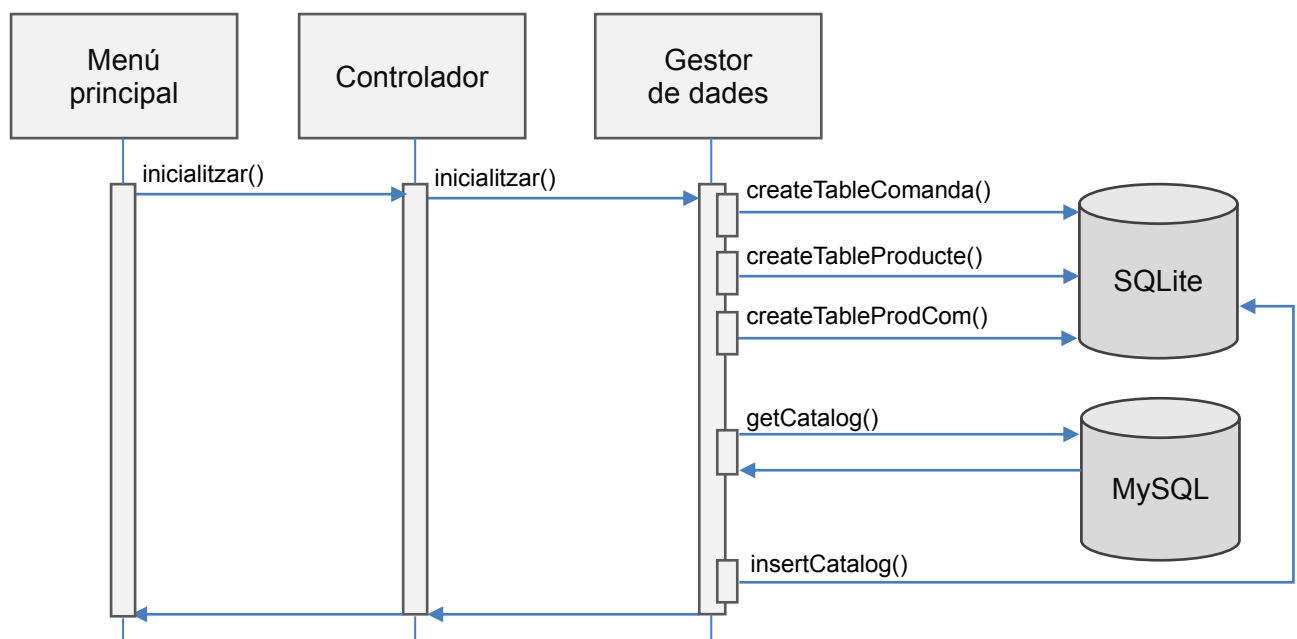
## Diagrames de seqüència

### Inicialització del sistema

La primera vegada que s'executa l'aplicació en un terminal (*client*) és necessari inicialitzar la base de dades i fer la càrrega inicial del catàleg.

L'esquema de la base de dades es crearà des del codi però la inicialització del catàleg es farà accedint al servidor i demanant la versió actual del catàleg que després inserirem en la base de dades local del terminal (SQLite), és a dir, es crearan les taules i tot seguit es forçarà una operació de sincronització del catàleg.

També obtindrem del servidor quina és la data de la versió actual del catàleg i guardarem aquesta data en local ja que no sempre que l'usuari inicia la funció de sincronització aquesta es realitzarà completa sinó que primer es comprova la data de la versió i només es cas que hagi canviat i per tant el catàleg hagi estat actualitzat s'importarà del servidor el nou catàleg.



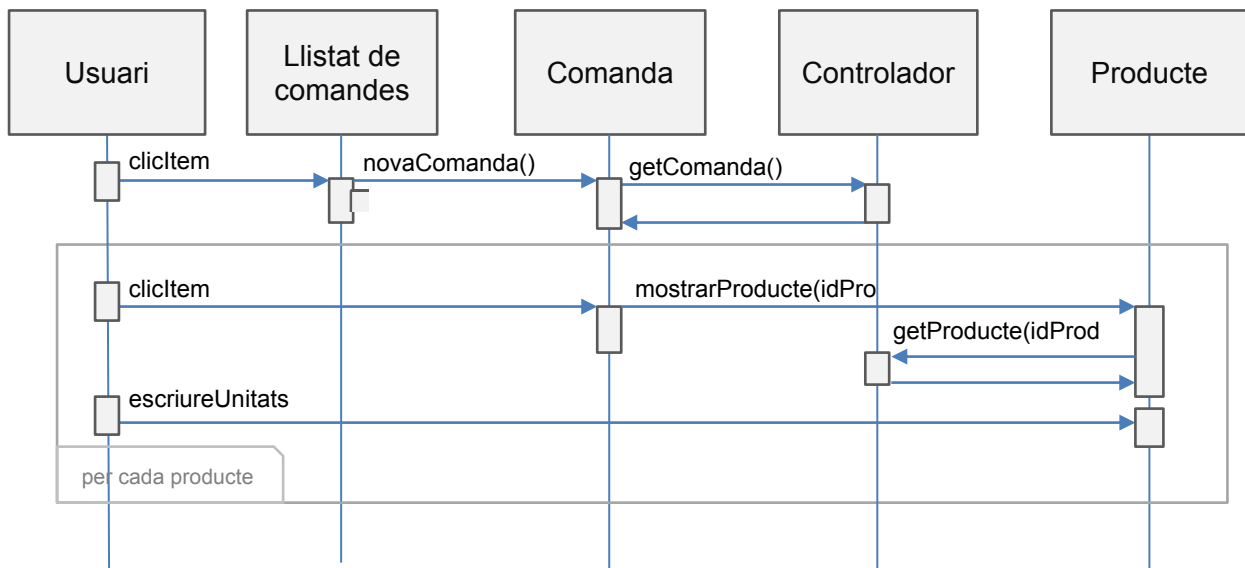
Imatge 14: Diagrama de seqüència de la inicialització de l'aplicació

### Creació d'una nova comanda

Per crear una nova comanda cal polsar sobre la primera opció de la llista de comandes que sempre és “Nova comanda”. Això farà que automàticament es salti a la pantalla de la comanda.

La pantalla de la comanda mostrarà un camp per introduir el nom de la comanda i mostrarà el catàleg (l'listat de productes actuals de l'aplicació). Seleccionant un element del catàleg s'accedirà a la pàgina d'aquest producte concret on es mostra el nom, la descripció i el número d'unitats seleccionades per la comanda actual (inicialment sempre zero). Aquí és on l'usuari seleccionarà les unitats que vol del producte actual i al desar tornarà de nou a la pantalla de la comanda amb les unitats del producte ja actualitzades.

La pantalla de la comanda també conté un camp per seleccionar si es tracta d'un pressupost de lloguer o venda i un camp per afegir l'adreça de correu del proveïdor, aquests camps només són obligatoris si la comanda es vol enviar al moment, si es vol desar aquests camps no cal omplir-los perquè actualment no es desen a la taula de la comanda on es guarden les comandes temporals que encara no han estat enviades.

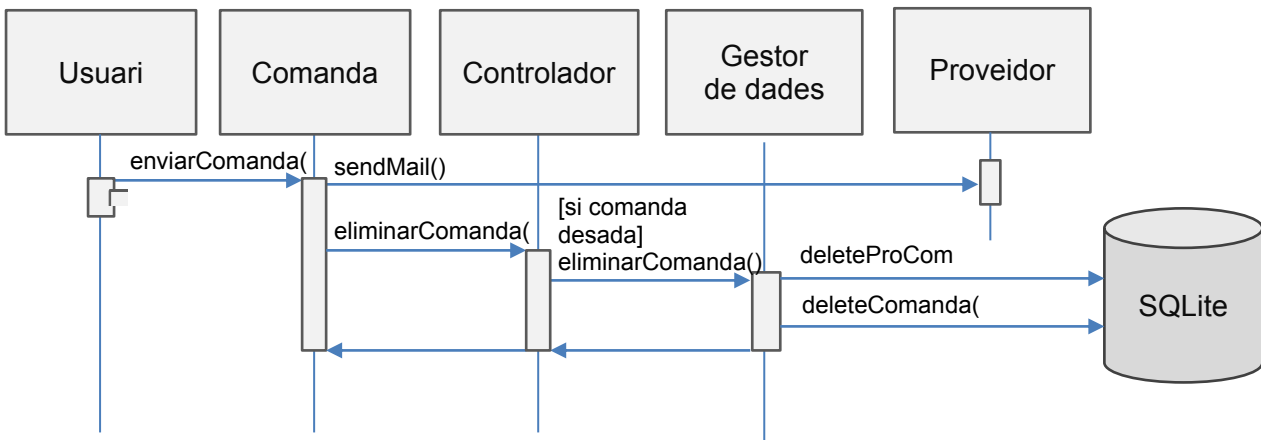


Imatge 15: Diagrama de seqüència de la creació d'una comanda

### Enviament d'una comanda

L'enviament d'una comanda es fa des de la pantalla de la comanda. Cal seleccionar si es tracta d'una petició de pressupost de lloguer o de venda i donar l'adreça del destinatari.

Si la comanda estava desada prèviament en el dispositiu s'esborrarà de la base de dades. Si es tractava d'una comanda que s'ha fet al moment no caldrà fer aquest accés a la base de dades ja que encara no estarà creada.

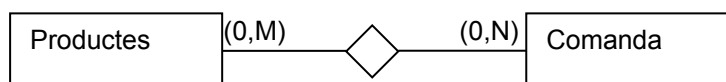


Imatge 16: Diagrama de seqüència de l'enviament d'una comanda

### Model de dades

El sistema disposa de dues base de dades relacionals, una al terminal que ha de permetre l'emmagatzemat de les comandes i una altre al servidor que només conté el catàleg de productes.

A nivell del model conceptual, el diagrama entitat-relació de la base de dades del dispositiu és:



Imatge 17: Diagrama E-R del dispositiu

Tant i com reflexa el diagrama, un producte pertany a 0 o moltes comandes i una comanda conté 0 o molts productes.

Per tant, la traducció d'aquest model dona lloc a les següents taules:

Productes	
PK	<u>_id</u>
	nom
	desc
	vigent

Comandes	
PK	<u>_id</u>
	nom

ProdCom	
PK	idCom
	idProd
	unitats

On s'ha generat una nova taula ProdCom que té com a clau primària l'identificador de la comanda i l'identificador del producte, on a més, l'identificador de la comanda és clau forana (Comandes (\_id)) i l'identificador del producte també és clau forana (Productes(\_id)). D'aquesta manera podem saber quins són els productes seleccionats per una comanda i en quina quantitat.

A més tindrem una taula per mantenir la versió actual del catàleg i serà la data que utilitzarem per saber quan és necessari la sincronització i qual no:

VersioCatalog
dVersio

La base de dades del servidor només sap del catàleg de productes però no de comandes, per tant, contindrà les taules de Productes i de VersioCatalog.

## Llicències

El sistema operatiu de la màquina de desenvolupament és Linux sota llicència GNU General Public License.

El sistema operatiu per a la màquina d'execució (terminal mòbil) és Android, que està sota llicència Apache 2.0 i GNU GPL.

En fer servir el SDK (Software Development Kit) així com el plugin per Eclipse cal acceptar la llicència Apache 2.0. La resta de la documentació disponible a la web d'Android és pot fer servir sota llicència Creative Commons Attribution 2.5.

El llenguatge de desenvolupament és Java d'Oracle que està subjecte a la Java Binary Code License Agreement.

L'entorn de desenvolupament triat és Eclipse que gaudeix d'un plugin especialment dissenyat per a la programació sobre Android, anomenat ADT (<https://dl-ssl.google.com/android/eclipse/>). Tot projecte desenvolupat amb Eclipse cau sota llicència EPL (Eclipse Public License).

Tot i això, en instal·lar l'entorn de desenvolupament, algunes parts d'aquest cauen sota altres tipus de llicències. Aquest és el cas de jcommon-1.0.12.jar que està sota llicència BSD o jfreechart-1.0.9.jar i jfreechart-1.0.9-swt.jar que estan sota llicència LGPL.

Caldrà treballar també sobre un SGBD (Sistema de Gestió de Bases de Dades) al terminal mòbil de l'Android. L'SGBD que ofereix aquest dispositiu és SQLite i es distribueix sota Public Domain.

Com a SGBD del servidor he triat MySQL, els projectes que en fan servir aquest programari es llicenciaran diferent segons el seu propòsit: <http://mysql.com/about/legal/licensing/index.html> .

Com a servidor de web i de Java Servlets instal·laré Apache + Tomcat, que cauen sobre llicència Apache 2.0.

## DESENVOLUPAMENT

---

Un cop exposats els objectius i característiques principals de l'aplicació, a continuació es detallen alguns dels aspectes més rellevants relacionats amb el desenvolupament.

L'ordre seguit en el desenvolupament ha estat treballar primer les funcionalitats només del client, amb una base de dades local al terminal inicialitzada directament per codi.

Un cop desenvolupades totes aquestes funcionalitats la següent tasca ha estat la inicialització del catàleg la primera vegada que s'executa l'aplicació (eliminant la inicialització per codi que s'havia creat a la etapa anterior) i la sincronització sota demanda. Per tant, s'han creat els fitxers php: un per sol·licitar la data de la versió del catàleg del servidor i un altre per obtenir el catàleg de productes.

L'última etapa del desenvolupament l'he dedicat a intentar millorar l'aparença de les vistes.

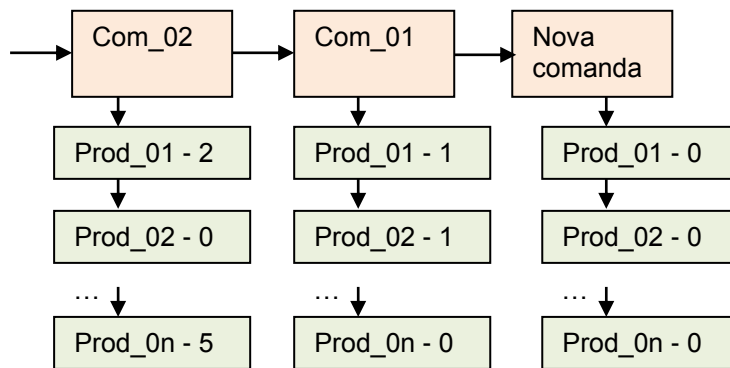
### Decisions

A continuació es detallen algunes de les decisions que s'han pres en aquesta etapa de desenvolupament.

S'ha pres com a premissa del desenvolupament que la llista de comandes locals al dispositiu no serà molt gran. Una comanda local és una comanda que ha estat creada però encara no ha estat enviada (quan s'envia s'elimina del sistema). L'utilitat de l'aplicació aLlum és poder enviar el pressupost al moment però no té cap sentit anar desant les comandes i no enviar-les, per tant, aquests casos de generar comandes locals només es donaran quan amb el lloc on s'està fent la localització i s'està generant la comanda no hi ha cobertura i no es pot fer aquest enviament, o bé, cal alguna consulta posterior sobre algun dels productes i per tant el director de fotografia decideix no enviar-la fins a resoldre els dubtes.

Amb aquesta premissa, l'estructura de dades escollida és una llista amb totes les comandes disponibles que es carrega a l'inici de l'aplicació (al accedir al llistat de les comandes), de manera que cada element d'aquesta llista és un objecte Comanda. Aquest objecte Comanda conté una llista d'objectes Producte amb tots els productes del catàleg. Aquest objecte producte a més de tenir el nom i la descripció, té també un camp 'unitats' que conté les unitats d'aquest producte seleccionats per la comanda actual.

S'ha pres aquesta decisió de tenir tota la llista de productes del catàleg a cada comanda i no només els productes seleccionats perquè així després a l'hora de la visualització és molt més ràpid i no cal anar cercant dintre de la llista de productes el identificador de cadascun dels productes de la comanda per actualitzar les unitats sinó que tenim directament la llista construïda amb les unitats de cada producte:



Imatge 18: Estructura de dades de la llista de comandes

Com a primer element d'aquesta llista de comandes sempre està la “Nova comanda”, és a dir, una comanda sense nom i amb tota la llista de productes inicialitzada amb el número d'unitats a 0.

L'arquitectura específica d'una aplicació Android en dificultava l'accés a l'objecte Controler des de cadascuna de les vistes, per solucionar el problema vaig decidir fer servir el patró *Factory* per tenir un mètode que, sense ser el constructor de la classe Controler, em retornés la instància a l'objecte:

```

public class Controler {

    private static Controler controler = null;
    ...

    private Controler () {...}

    public static Controler getInstance(){
        if (controler == null) {
            controler = new Controler();
        }
        return controler;
    }
    ...
}
  
```



Tal i com descriu aquest patró els objectes Factoria [7] són útils quan s'han de fer servir des de molts llocs diferents de l'aplicació. El fet de declarar l'atribut `Controler` com a *estatic* i assegurar que la construcció es produeix en el mètode `getInstance()`, tal i com es veu en el codi anterior, ens proporciona aquesta funcionalitat.

## Disseny final de pantalles

Menú inicial:



**Imatge 19: Pantalla inicial**

Llistat de comandes:



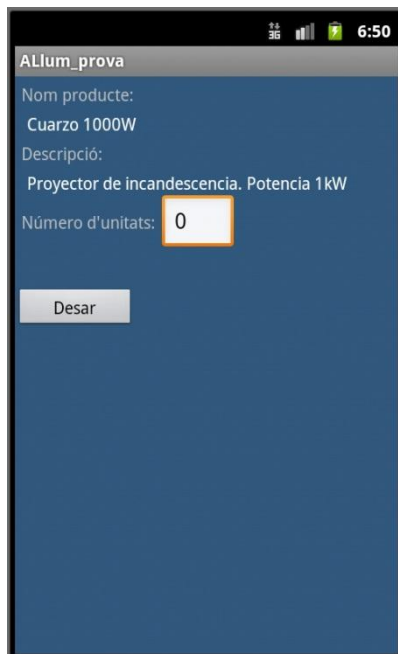
**Imatge 20: Pantalla del llistat de comandes**

Comanda:



**Imatge 21: Pantalla comanda**

Producte:



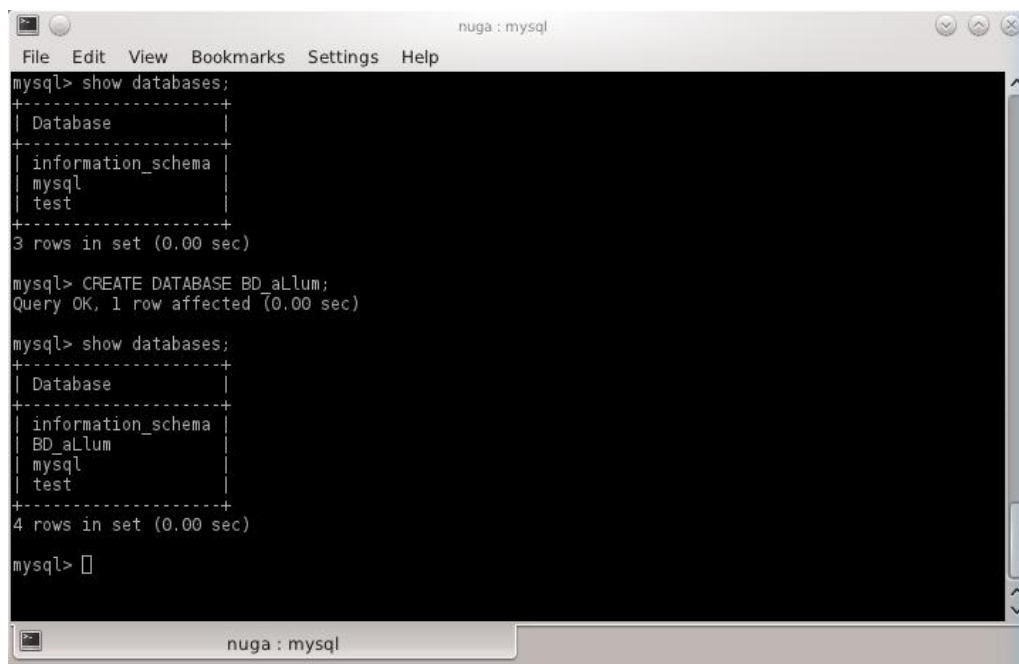
**Imatge 22: Pantalla producte**

## IMPLANTACIÓ

Un cop finalitzat el desenvolupament i superades les proves de validació l'aplicació està preparada per ser implantada en l'entorn de producció.

Per fer aquesta implantació cal primerament inicialitzar el servidor amb la creació de la base de dades que contindrà el catàleg inicial dels productes.

Des de mysql creo la nova base de dades BD\_aLlum amb la sentència de CREATE DATABASE:



```

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| test |
+-----+
3 rows in set (0.00 sec)

mysql> CREATE DATABASE BD_aLlum;
Query OK, 1 row affected (0.00 sec)

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| BD_aLlum |
| mysql |
| test |
+-----+
4 rows in set (0.00 sec)

mysql>

```

Imatge 23: Creació de la base de dades

Tot seguit cal crear les taules. Podem executar directament un fitxer script amb les comandes de la creació de la taula fer-ho directament des de comanda, la forma tant és, però el contingut és la creació de les dues taules que l'aplicació utilitza al servidor, la taula de productes i la taula amb la data de l'última versió del catàleg:

```

CREATE TABLE Productes (
  _id          INT          PRIMARY KEY AUTO_INCREMENT,
  nom          VARCHAR(30)  NOT NULL UNIQUE,
  descripcio  VARCHAR(150),
  vigent      BOOLEAN      DEFAULT '1');

CREATE TABLE VersioCataleg (dVersion TIMESTAMP);

```

```

mysql> use BD_aLlum;
Database changed
mysql> CREATE TABLE Productes (
->   id INT PRIMARY KEY AUTO_INCREMENT,
->   nom VARCHAR(30) NOT NULL UNIQUE,
->   descripcio VARCHAR(150),
->   vigent BOOLEAN DEFAULT '1');
Query OK, 0 rows affected (0.01 sec)

mysql> CREATE TABLE VersioCataleg (dVersion TIMESTAMP);
Query OK, 0 rows affected (0.01 sec)

mysql> 

```

Imatge 24: Creació de les taules

```

mysql> desc Productes;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| id         | int(11)       | NO   | PRI | NULL    | auto_increment |
| nom        | varchar(30)   | NO   | UNI | NULL    |                |
| descripcio | varchar(150)  | YES  |     | NULL    |                |
| vigent     | tinyint(1)    | YES  |     | 1       |                |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

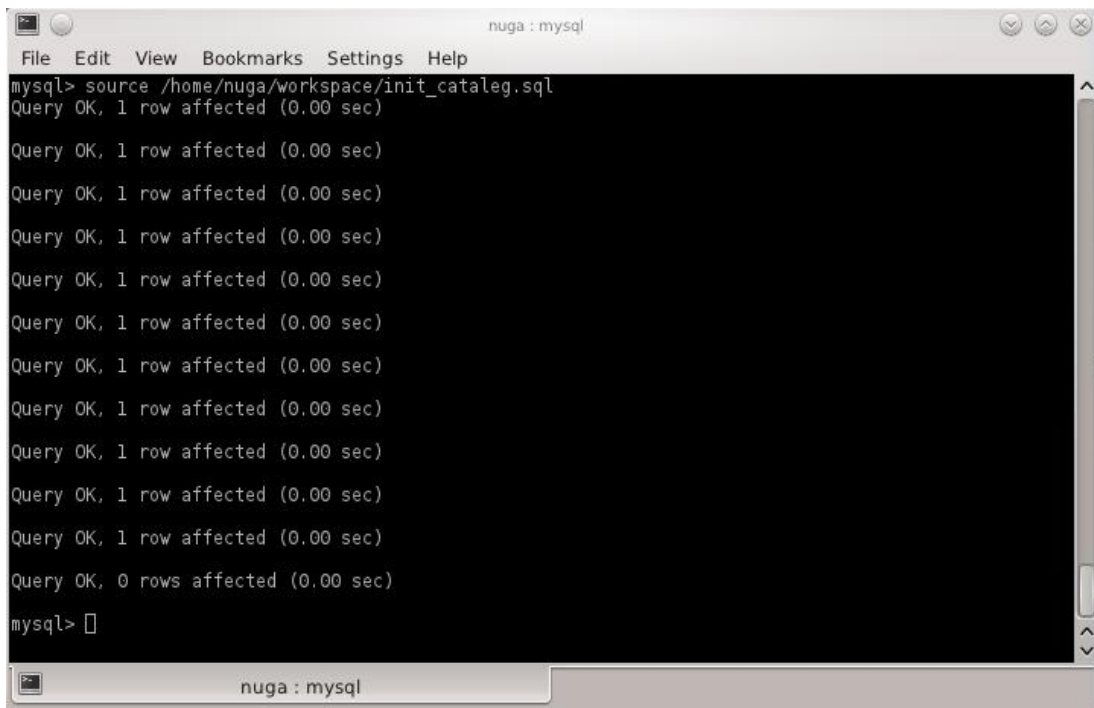
mysql> desc VersioCataleg;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default          | Extra          |
+-----+-----+-----+-----+-----+-----+
| dVersion   | timestamp     | NO   |     | CURRENT_TIMESTAMP | on update CURRENT_TIMESTAMP |
+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> 

```

Imatge 25: Visualització de les taules

Un cop creades les taules cal inicialitzar-les, i aquest cop, com el número de sentències és més llarg, ho he fet cridant a un fitxer de configuració que conté tots els inserts del elements que formaran el catàleg inicial ([Annex1](#)):



```

mysql> source /home/nuga/workspace/init_cataleg.sql
Query OK, 1 row affected (0.00 sec)

Query OK, 1 row affected (0.00 sec)

Query OK, 1 row affected (0.00 sec)

Query OK, 1 row affected (0.00 sec)

Query OK, 1 row affected (0.00 sec)

Query OK, 1 row affected (0.00 sec)

Query OK, 1 row affected (0.00 sec)

Query OK, 1 row affected (0.00 sec)

Query OK, 1 row affected (0.00 sec)

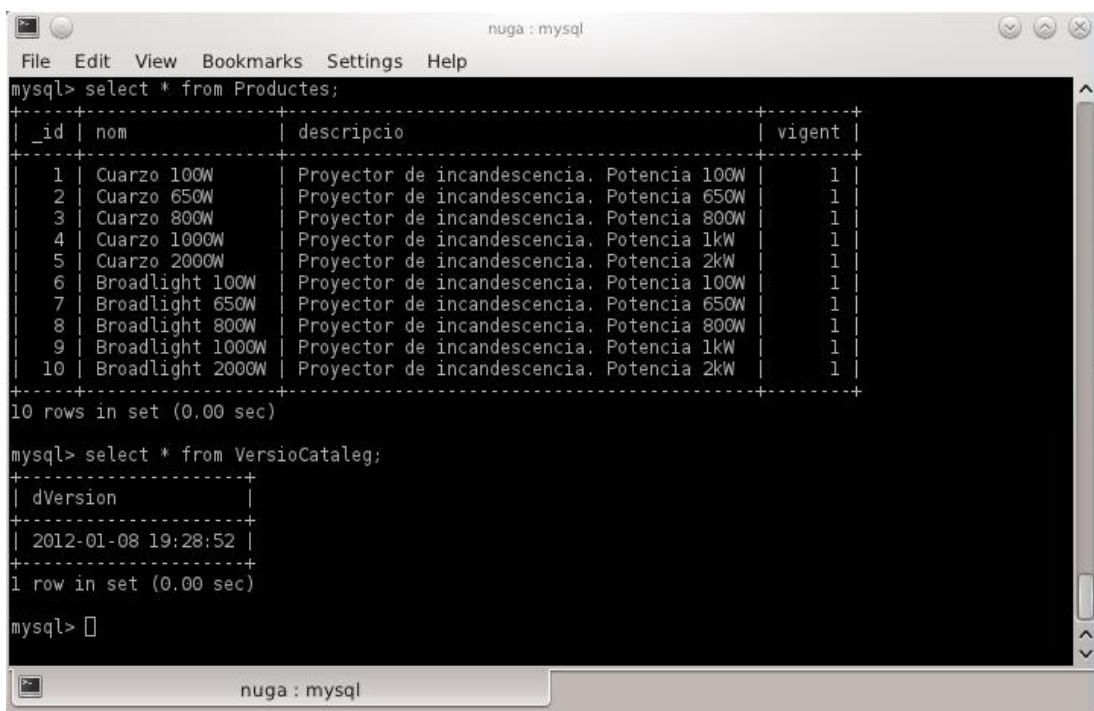
Query OK, 1 row affected (0.00 sec)

Query OK, 1 row affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

mysql>
    
```

**Imatge 26: Execució de l'script d'inicialització**



```

mysql> select * from Productes;
+----+-----+-----+-----+
|_id| nom          | descripcio                                     | vigent |
+----+-----+-----+-----+
| 1 | Cuarzo 100W | Proyector de incandescencia. Potencia 100W | 1      |
| 2 | Cuarzo 650W | Proyector de incandescencia. Potencia 650W | 1      |
| 3 | Cuarzo 800W | Proyector de incandescencia. Potencia 800W | 1      |
| 4 | Cuarzo 1000W | Proyector de incandescencia. Potencia 1kW  | 1      |
| 5 | Cuarzo 2000W | Proyector de incandescencia. Potencia 2kW  | 1      |
| 6 | Broadlight 100W | Proyector de incandescencia. Potencia 100W | 1      |
| 7 | Broadlight 650W | Proyector de incandescencia. Potencia 650W | 1      |
| 8 | Broadlight 800W | Proyector de incandescencia. Potencia 800W | 1      |
| 9 | Broadlight 1000W | Proyector de incandescencia. Potencia 1kW  | 1      |
| 10 | Broadlight 2000W | Proyector de incandescencia. Potencia 2kW  | 1      |
+----+-----+-----+-----+
10 rows in set (0.00 sec)

mysql> select * from VersioCataleg;
+-----+
| dVersion |
+-----+
| 2012-01-08 19:28:52 |
+-----+
1 row in set (0.00 sec)

mysql>
    
```

**Imatge 27: Resultat de la inicialització**

Per últim cal copiar el fitxers php a /var/www/htdocs.

El fitxer `versio.php` retorna el contingut de la taula `VersioCataleg`, és a dir, la data de l'última actualització del catàleg al servidor.

El fitxer `productes.php` retorna tots els productes vigents de la taula de `Productes`, per tant, el catàleg actual del servidor.

L'intercanvi d'informació entre el client i el servidor, i per tant, resultat de tots dos php és en format JSON [10].

## CONCLUSIONS

---

La realització d'aquest projecte m'ha permès ampliar els coneixements de desenvolupament d'aplicacions per Android. La gran quantitat de documentació que hi ha disponible i el fet d'utilitzar un llenguatge de programació tant conegut com Java m'ha facilitat la seva realització. També m'ha permès aprendre el format de declaració d'objectes JSON.

En aquest projecte he pogut tornar a posar en pràctica i actualitzar amb tecnologies més recents l'enginyeria del programari i el desenvolupament d'un projecte, així com aplicar una gran quantitat dels coneixements adquirits al llarg d'aquest màster.

Amb els resultats obtinguts crec que s'han complert els objectius inicials: l'aplicació aLlum, programada des de zero pensant en Android, cobreix no només totes les funcionalitats de l'aplicació iLuz sinó que n'afegeix i les millora.

L'aplicació disposa d'una interfície intuïtiva que permet que un perfil d'usuari no tècnic amb una mínima descripció de l'aplicació sigui capaç de fer-la servir sense dificultats. L'usuari pot realitzar una comanda dels productes disponibles a la base de dades seleccionant quantes unitats de cada producte vol. També pot desar a la base de dades local del dispositiu la comanda generada o bé enviar-la per correu electrònic detallant automàticament si es tracta d'un pressupost de venda o lloguer i especificant tot el detall de la comanda. Tot això tenint el catàleg local al dispositiu, sense que sigui estàtic ni calgui una actualització de programari per actualitzar-lo. El client pot demanar en qualsevol moment la sincronització amb la base de dades del servidor i disposar automàticament de tots els productes nous.

Finalment, però no menys important, aquesta aplicació amplia el catàleg de programari lliure i està a disposició de la comunitat. El projecte crec que està ben documentat i fortament desacoblats el model de dades i la vista de la interfície d'usuari; la qual cosa permetrà la reusabilitat del codi amb facilitat. El projecte, per tant, podrà ser millorat o particularitzat per qualssevol necessitats.

La limitació de temps m'ha fet deixar abandonades pel camí algunes característiques de l'aplicació que inicialment estaven plantejades i que ara passaran a millores de futur.

Entre aquestes funcionalitats està la creació d'una nova Activity (pantalla) que mostri el contingut del missatge abans de ser enviat per correu electrònic i que permeti a l'usuari la seva modificació. El text que s'envia com a cos del correu és fix: una demanda de pressupost de lloguer o venda (segons l'opció seleccionada) més un detall dels productes

que tenen alguna unitat seleccionada. De vegades, per ser molt útil poder afegir qualsevol comentari a aquest correu i aquesta era la funcionalitat d'aquesta nova pantalla, poder afegir una capçalera o un peu a aquest text automàtic del correu. També estava previst poder seleccionar el destinatari d'aquest correu de l'agenda de contactes del dispositiu, però en aquesta primera versió cal introduir l'adreça manualment.

Ja des del començament, aquest projecte era només una primera versió d'un sistema que té moltes possibilitats de millora, perquè només canviant el catàleg de productes a, per exemple, el productes d'una empresa distribuïdora de begudes, els comercials que van pels locals (bars o restaurants) per fer la comanda que serviran en dies posteriors ja podrien enviar la comanda en el mateix moment de fer-la i per tant la comanda seria servida molt més ràpidament.

Una de les possibles millores de futur seria poder mostrar imatges per a cada producte: inicialment una imatge de resolució petita però amb la possibilitat de mostrar una imatge de millor qualitat al obrir la fitxa del producte.

També es podria millorar la interfície per poder seleccionar les unitats de cada producte directament des de la llista de productes de la comanda, i així només caldria accedir a la pantalla del producte quan es vulgui llegir la seva descripció detallada o veure la seva imatge d'alta resolució (si la millora anterior ha estat implementada).

Si es força a una sincronització automàtica amb el servidor abans de realitzar una comanda, el catàleg de productes també podria tenir informat el número d'unitats de producte disponible i així l'usuari pot escollir un producte alternatiu si veu que no disposa de totes les unitats d'un producte desitjades. Aquesta millora també podria continuar amb l'enviament de la comanda no només per correu electrònic sinó també actualitzar directament la base de dades del servidor amb la reserva d'aquests productes.



## BIBLIOGRAFIA

---

- [1] [En línea]. Available: [www.apple.com/](http://www.apple.com/). [Último acceso: 27 12 2010].
- [2] [En línea]. Available: <http://developer.apple.com/technologies/ios/>. [Último acceso: 27 12 2010].
- [3] [En línea]. Available: <http://developer.android.com/guide/basics/what-is-android.html>.
- [4] [En línea]. Available: <http://www.eclipse.org/org/>.
- [5] [En línea]. Available: [http://es.wikipedia.org/wiki/Eclipse\\_%28software%29](http://es.wikipedia.org/wiki/Eclipse_%28software%29).
- [6] [En línea]. Available: <http://developer.android.com/sdk/index.html>.
- [7] J. Hunt, Guide to the unified process featuring UML, java and desing patterns, London: Springer, 2003.
- [8] R. C. Martin, UML for JAVA programmers, Upper Saddle River: Prentice Hall, 2003.
- [9] J. G. Barbara Liskov, Program Development in Java, Upper Saddle River: Addison-Wesley, 2001.
- [10] <http://www.json.org/>.
- [11] F. J. Ceballos Sierra, Java2: Curso de programación, Madrid: Ra-ma, 2002.
- [12] C. Haseman, Android essentials, California: Apress, 2008.
- [13] S. Y. K. S. M. D. Hashimi, Pro android 2, New York: Apress, 2010.
- [14] K. Knoernschild, Java design, Indianapolis: Addison Wesley, 2002.
- [15] B. P. G. & W. D. McLaughlin, Head first object-oriented analysis and design, Sebastopol, California: O'Reilly, 2006.
- [16] J. H. F. A. Nino, An introduction to programming and object-oriented design using java (3rd ed.), Hoboken: J. Wiley, 2008.

## ANNEX1. SCRIPT DE CREACIÓ DEL CATÀLEG

---

```
insert into Productes (nom, descripcio) values
    ("Cuarzo 100W", "Proyector de incandescencia. Potencia 100W");
insert into Productes (nom, descripcio) values
    ("Cuarzo 650W", "Proyector de incandescencia. Potencia 650W");
insert into Productes (nom, descripcio) values
    ("Cuarzo 800W", "Proyector de incandescencia. Potencia 800W");
insert into Productes (nom, descripcio) values
    ("Cuarzo 1000W", "Proyector de incandescencia. Potencia 1kW");
insert into Productes (nom, descripcio) values
    ("Cuarzo 2000W", "Proyector de incandescencia. Potencia 2kW");
insert into Productes (nom, descripcio) values
    ("Broadlight 100W", "Proyector de incandescencia. Potencia 100W");
insert into Productes (nom, descripcio) values
    ("Broadlight 650W", "Proyector de incandescencia. Potencia 650W");
insert into Productes (nom, descripcio) values
    ("Broadlight 800W", "Proyector de incandescencia. Potencia 800W");
insert into Productes (nom, descripcio) values
    ("Broadlight 1000W", "Proyector de incandescencia. Potencia 1kW");
insert into Productes (nom, descripcio) values
    ("Broadlight 2000W", "Proyector de incandescencia. Potencia 2kW");

insert into VersioCatalog value (CURRENT_TIMESTAMP);

commit;
```

## ANNEX2. VERSIO.PHP

---

```
<?php
    $db_user    = "root";
    $db_pass    = "";
    $db_host    = "localhost";
    $db_schema  = "BD_aLlum";
    $db_query   = "SELECT * FROM VersioCataleg";

    // connexió amb la base de dades
    $link = mysql_connect($db_host, $db_user, $db_pass) or die('No es pot con-
nectar amb la BD');
    mysql_select_db($db_schema, $link) or die('ERROR : No es pot seleccionar la
BD!');

    // execució de la consulta
    $result = mysql_query($db_query, $link) or die('Error en la consulta: ' .
$query);

    // construir un array amb el resultat de la consulta
    if(mysql_num_rows($result)) {
        $data = mysql_result($result, 0);
    }

    // formatejar la sortida amb JSON
    header('Content-type: application/json');
    echo json_encode(array('versio'=>$data));

    // desconexió de la base de dades
    @mysql_close($link);
?>
```

## ANNEX3. PRODUCTES.PHP

---

```
<?php
    $db_user    = "root";
    $db_pass    = "";
    $db_host    = "localhost";
    $db_schema  = "BD_aLlum";
    $db_query   = "SELECT * FROM Productes where vigent=1";

    // connexió amb la base de dades
    $link = mysql_connect($db_host, $db_user, $db_pass) or die('No es pot con-
nectar amb la BD');
    mysql_select_db($db_schema, $link) or die('ERROR : No es pot seleccionar la
BD!');

    // execució de la consulta
    $result = mysql_query($db_query, $link) or die('Error en la consulta: ' .
$query);

    while($row=mysql_fetch_assoc($result))
        $output[]=$row;

    // formatejar la sortida amb JSON
    header('Content-type: application/json');
    echo json_encode(array('prods'=>$output));

    // desconexió de la base de dades
    @mysql_close($link);

?>
```