

Treball Final de Màster

*Innovació empresarial en el procés de
transmissió de dades mitjançant una
aplicació mòbil*

Josep Codina i Bau

Màster en Tecnologies Aplicades de la Informació

Directora: Alejandra Aramayo

Codirector: Albert Baucells

Vic, Juliol de 2012

Vull agrair el suport rebut per part de totes aquelles persones que m'han fet costat i m'han ajudat a portar a terme aquest projecte.

També vull donar les gràcies de forma especial a l'Alejandra Aramayo i a l'Albert Baucells, que com a directors del projecte han portat a terme una tasca excel·lent.

Moltes gràcies.

Índex

1. Introducció	8
1.1. Justificació.....	8
1.2. Objectius	9
1.3. Marc conceptual general del projecte	10
1.4. Estructura de la memòria.....	12
2. Bases del projecte	13
2.1. Empresa.....	13
2.2. Descripció del projecte	17
2.3. Esquema de funcionament del sistema	18
2.4. Metodologia del projecte.....	19
3. Descripció funcional de la solució	24
3.1. Aplicació per dispositius mòbils	26
3.2. Base de dades.....	34
3.3. Aplicatiu servidor	35
3.4. Verificador	35
4. Descripció tècnica de la solució	37
4.1. Aplicació per dispositius mòbils	39
4.2. Base de dades.....	52
4.3. Aplicatiu servidor	60
4.4. Verificador	68
4.5. Protocol de comunicacions.....	71
5. Resultats i conclusions	85
5.1. Resultats	85
5.2. Possibles ampliacions del sistema.....	88
5.3. Conclusions.....	89
Bibliografia	92

Índex de figures

Figura 1-1 El procés de la innovació tecnològica segons Marquis. Elaborat a partir de les aportacions de Gruber i Marquis (1969), Myers i Marquis (1969), Marquis (1969) i Utterback (1969, 1971 ^a , 1971, b).	11
Figura 2-1 Simulació del muntatge electrònic	13
Figura 2-2 Simulació del procés electromecànic	14
Figura 2-3 Imatge d'un verificador	15
Figura 3-1 Telèfon intel·ligent on cal instal·lar l'aplicació.	24
Figura 3-2 Simulació de la localització de la base de dades.	24
Figura 3-3 Verificador.	25
Figura 3-4 Simulació de la localització de l'aplicatiu servidor.	25
Figura 3-5 Icona de l'aplicació.	26
Figura 3-6 Pantalla inicial.	27
Figura 3-7 Pantalla on donar-se d'alta d'un producte.	28
Figura 3-8 Pantalla on identificar-se i escollir el tipus de cerca.	29
Figura 3-9 Pantalla on escollir producte i dates.	30
Figura 3-10 Pantalla on escollir producte i quantitat.	31
Figura 3-11 Pantalla on es visualitzen els resultats.	32
Figura 3-12 Estructura de pantalles de l'aplicació.	33
Figura 3-13 Diagrama de funcionament d'un verificador.	36
Figura 4-1 Diagrama del sistema	38
Figura 4-2 Gràfica de les versions d' <i>Android</i> que utilitzen el dispositius actualment.	40
Figura 4-3 Cicle de vida d'una <i>Activity</i> .	41
Figura 4-4 Exemple de disseny d'un <i>layout</i> .	43
Figura 4-5 Definició i propietats d'un <i>layout</i> del tipus <i>Relative</i> .	44
Figura 4-6 Definició i propietats d'un <i>widget</i> del tipus <i>TextView</i> .	45
Figura 4-7 Definició i propietats d'un altre <i>widget</i> del tipus <i>TextView</i> .	45
Figura 4-8 Definició i propietats d'un <i>widget</i> del tipus <i>EditText</i> .	45
Figura 4-9 Definició i propietats d'un <i>widget</i> del tipus <i>Spinner</i> .	46
Figura 4-10 Definició i propietats d'un <i>widget</i> del tipus <i>Button</i> .	46
Figura 4-11 Declaracions dels components del <i>layout</i> .	47
Figura 4-12 Mètode <i>onCreate()</i> .	47
Figura 4-13 Declaracions dels components del <i>layout</i> .	48
Figura 4-14 Reescriptura del <i>callback</i> de l'objecte <i>View</i> .	48

Figura 4-15	Llista d' <i>activities</i> al fitxer <i>AndroidManifest</i>	50
Figura 4-16	Permisos del fitxer <i>AndroidManifest</i>	50
Figura 4-17	Primera proposta d'estructura de la base de dades.	53
Figura 4-18	Segona proposta d'estructura de la base de dades.....	55
Figura 4-19	Proposta definitiva de l'estructura de la base de dades.....	56
Figura 4-20	Url del servidor web.	72
Figura 4-21	Com cridar una funció.	73
Figura 4-22	Exemple de <i>URL</i> de la funció <i>DonarAlta</i>	74
Figura 4-23	Exemple de <i>URL</i> de la funció <i>AfegirTest</i>	74
Figura 4-24	Exemple de <i>URL</i> de la funció <i>ProductesUsuari</i>	75
Figura 4-25	Exemple de <i>URL</i> de la funció <i>RecollirResultats</i> per dates.....	75
Figura 4-26	Exemple de <i>URL</i> de la funció <i>RecollirResultats</i> per quantitat.....	76
Figura 4-27	Exemple de <i>URL</i> de la funció <i>TestosProducte</i>	76
Figura 4-28	Exemple de <i>URL</i> de la funció <i>RelacioTestProducte</i>	77
Figura 4-29	Exemple de <i>URL</i> de la funció <i>ProductesSistema</i>	77
Figura 4-30	Exemple d'establiment de connexió.	79
Figura 4-31	Exemple de selecció de base de dades.	80
Figura 4-32	Exemple d'introducció de dades a la base de dades.....	81
Figura 4-33	Exemple de cerca de dades a la base de dades.	82
Figura 4-34	Exemple de tancament de la connexió.....	82
Figura 4-35	Exemple on es pot comprovar el protocol descrit.	83
Figura 5-1	Fitxa tècnica de la prova pilot.....	86

Resum Treball Final de Màster

Màster en Tecnologies Aplicades de la Informació

Títol: Innovació empresarial en el procés de transmissió de dades mitjançant una aplicació mòbil

Paraules clau: Innovació, informació, aplicació per dispositius mòbils, base de dades, control de qualitat, aplicatiu servidor.

Autor: Josep Codina Bau

Direcció: Alejandra Aramayo

Codirecció: Albert Baucells

Data: Juliol de 2012

Resum

Disposar d'informació de qualitat, en el moment oportú i en el volum adequat és el que actualment necessiten les empreses per poder prendre decisions òptimes. L'empresa JCM Technologies no és una excepció i és en la obtenció d'informació del procés productiu de l'empresa on aquest projecte pren part.

L'objectiu d'aquest projecte de final de màster ha consistit en desenvolupar un sistema capaç d'aportar a l'empresa informació sobre els resultats obtinguts durant el control de qualitat dels productes, informació de la qual no disposava. Les dades existien però no es podien convertir en informació valuosa perquè no es disposava d'un mecanisme capaç tractar-les.

Per complir els objectius marcats s'ha creat un sistema format pels següents elements: una aplicació per dispositius mòbils *Android*, una base de dades, un nou programa pel control de la qualitat i aplicatiu servidor. L'aplicació permet obtenir i visualitzar la informació. La base de dades és on es guarda la informació que envia el programa del control de qualitat i finalment, l'aplicatiu servidor habilita la comunicació entre els diferents elements descrits.

El fet que dona valor al sistema és que tots els elements que el formen tenen la capacitat de comunicar-se entre ells mitjançant Internet. Gràcies a això, és possible obtenir informació a temps real de la producció que s'està portant a terme a qualsevol punt del món.

En conclusió, cal dir que els objectius s'han complert ja que s'ha aconseguit un sistema ràpid i segur. D'aquesta manera JCM Technologies pot disposar d'una informació molt valuosa que li permet que certes decisions no siguin preses a l'atzar.

Master Final Project Summary

Applied Information Technologies Master

Title: Business innovation in the transmitting data process through a mobile application

Key words: Innovation, information, mobile phone application, database, quality control, server application.

Author: Josep Codina Bau

Directed by: Alejandra Aramayo

Codirected by: Albert Baucells

Date: July 2012

SUMMARY

Being able to have relevant information at the appropriate moment and in suitable amounts is what companies need nowadays to make optimal decisions. The company JCM Technologies is not an exception and obtaining information from its production process is where this project takes part.

The aim of this master final project is to develop a system capable of providing the company information on the results obtained during the quality control of the products, information which was not available. The data existed but could not become valuable information because a mechanism able to deal with them did not exist.

To meet the goals a system composed of the following elements has been created: an application for *Android* devices, a database, a new quality control program and a server application. The application allows the company to obtain information. The database is where the information sent by the new quality control program is stored. Finally, the server application enables communication between the different elements.

What gives value to the system is that all the elements have the ability to communicate with each other via the Internet. Thanks to this communication, it is possible to obtain, in real time, information about the production that is taking place anywhere in the world.

In conclusion, all goals have been achieved as a quick and secure system has been created. Thus, JCM Technologies has the valuable information to make the right decisions.

1. Introducció

En aquest apartat es fa una breu introducció al projecte, justificant-lo i descrivint aspectes com les motivacions a l'hora de portar-lo a terme. Els objectius també s'hi troben descrits, així com una breu descripció de com es troba estructurada aquesta memòria.

1.1. Justificació

Aquest projecte s'ha portat a terme amb la finalitat de completar el màster en Tecnologies Aplicades de la Informació. Com el nom del màster indica, pràcticament tots els coneixements adquirits han estat aplicats a casos pràctics, fet que també es compleix en aquest projecte.

L'objectiu principal d'aquest projecte consisteix en donar resposta a una necessitat sorgida al lloc de treball que ocupa actualment l'autor. Aquest treballa a una empresa que dissenya i comercialitza productes electrònics i que subcontracta el procés de producció a empreses especialitzades. Per assegurar la bona qualitat dels productes fabricats pels subcontractats, es dissenyen sistemes de test que tots els productes han de superar satisfactòriament. Actualment, l'empresa que fa l'encàrrec de fabricació, no aprofita tota la potencialitat de la informació obtinguda en aquests sistemes de test. Coneixent el seu valor per a la millora de presa de decisions dels processos empresarials, de la gestió de costos i dels propis productes, s'ha cregut convenient donar-hi solució. Poder accedir a aquest tipus d'informació a temps real i tractar-la amb rapidesa dóna la possibilitat de detectar problemes o errors en una fase en la que encara existeix la possibilitat de solucionar-los sense haver de fer una gran inversió.

Per tots és conegut l'important creixement que ha experimentat la utilització de telèfons intel·ligents els últims anys, així com l'increment del nombre de descàrregues de les aplicacions que aquests utilitzen. Cada vegada són més les tasques que es poden portar a terme des d'aquest tipus de terminals i el futur tendeix cap a la substitució de més utensilis de la vida diària.

La possibilitat que les tecnologies de la informació ofereixen de poder-ho controlar tot des de qualsevol lloc i en qualsevol moment, és un dels motius que més força ha fet a l'hora de portar a terme aquest projecte. El fet d'haver vist durant les classes rebudes que, tot i ser una temàtica complexa, crear una aplicació per un dispositiu mòbil no era una idea tant llunyana com podia semblar, va ser capaç de generar una curiositat que ha estat satisfeta amb aquest projecte.

Des de que es va començar a desenvolupar la idea del projecte que aquí es presenta, s'han valorat diferents opcions de millora en el tractament de la informació recollida a peu de fàbrica. Finalment s'ha decidit crear una aplicació per la plataforma *Android*. Mitjançant aquesta aplicació, els usuaris poden conèixer a temps real els

resultats de les diferents verificacions que es porten a terme a cadascun dels productes, sigui en quina sigui la part del món en la que es porta a terme la producció. Perquè tot això funcioni però, ha calgut crear també un aplicatiu servidor que atén totes les peticions de l'aplicació i dels diferents verificadors, així com una base de dades on s'emmagatzema tota la informació.

Escollir la opció de crear una aplicació per un dispositiu mòbil no ha estat a l'atzar, sinó que un dels motius de més pes ha estat el fet que en assignatures com *Eines pel Control i Desenvolupament de Sistemes* s'havia estudiat la plataforma *Android*, entre d'altres.

Un altre aspecte tractat en assignatures del màster i posat a la pràctica en aquest projecte ha estat la manera de portar-lo a terme. El fet que en assignatures com *Gestió de Projectes* i *Organització d'Empreses* s'haguessin estudiat diferents teories i metodologies sobre disseny i gestió de projectes, particularment, en l'àmbit de projectes d'innovació, ha fet que per crear aquest s'hagin seguit unes pautes concretes. En aquest sentit i vist que actualment les empreses tendeixen a la "gestió per objectius" organitzant les tasques i sistemes de treball com a projectes, em vaig interessar en estudiar l'estàndard *PMBok* d'administració de projectes, desenvolupat pel *Project Management Institute (PMI)*. Per aquest motiu, aquest projecte ha estat realitzat seguint aquest estàndard en tot allò que és aplicable, particularment, en les tasques de planificació de projectes.

1.2. Objectius

Els objectius d'aquest projecte es divideixen en dos grups. El principal, en el que cal posar tots els esforços per complir, i els específics, que són de caire més personal i que es compleixen a mesura que es porta a terme el primer.

1.2.1. Principals

L'objectiu principal d'aquest projecte consisteix en donar una resposta innovadora a una necessitat real que té una empresa. Aquesta necessitat consisteix en millorar el sistema de transmissió de dades i el procés de gestió de la informació per assegurar l'eficiència del control de qualitat de la producció externalitzada. Analitzar el context de la necessitat real, estudiar-la i intentar trobar-l'hi la millor solució és el punt de partida del projecte, mentre que dissenyar la opció més òptima i crear-la en són el fonament.

1.2.2. Específics

Aportar una solució concreta al problema estudiat i dissenyar un sistema eficient aplicant els coneixements teòrics i pràctics estudiats durant el màster TAI. Per una part els referents a la gestió i tractament de la informació i les dades, que correspon a la part més tecnològica i per l'altra, la de la gestió i la innovació empresarial.

Ampliar els coneixements d'aquestes dues grans matèries, aprofundir en la creació d'una aplicació per a dispositius mòbils, així com en la gestió de projectes són dos aspectes que personalment m'atrauen tant que fan que aquest projecte sigui més un repte personal que una obligació per aconseguir una titulació superior.

1.3. Marc conceptual general del projecte

A l'acadèmia hi ha el consens que la innovació és un element de creació de riquesa d'un país i de la competitivitat de les empreses (Chesnais, F., 1986), en aquest sentit l'autor M. Porter, creador del concepte de l'estratègia competitiva, considera que "la competitivitat d'una nació depèn de la capacitat de la seva indústria per innovar i millorar. Les empreses aconseguen avantatges competitius mitjançant innovacions " (Porter, 1980)

L'empresa actual està obligada a ser innovadora per sobreviure, la globalització de l'economia i el progrés tecnològic han intensificat la pressió de la competència i han reduït els cicles de vida dels productes i dels processos empresarials. L'empresa que no innova en productes i/o en els processos de la cadena de valor¹ pot ser fàcilment assolida pels competidors i quedar fora del mercat. *La innovació és arriscada, però no innovar ho és encara més.*

La innovació es realitza mitjançant l'aplicació de nous coneixements i de la tecnologia, o una nova combinació dels ja existents. Aquests poden ser desenvolupats internament a l'empresa o externament (en col·laboració externa o adquirits mitjançant serveis d'assessorament o per compra de tecnologia).

A les últimes dècades, diferents autors han estudiat les fases internes que realitzen les empreses per innovar i han plantejat models explicatius per entendre el procés intern d'innovació. El *model lineal* explica el procés des que l'empresa inicia una investigació bàsica², seguint amb la investigació aplicada³ i el desenvolupament

¹ Segons el manual d'Oslo (OCDE, 1997) s'entén per innovació la concepció i implantació de canvis significatius en el producte, el procés, el màrqueting o la organització de l'empresa amb el propòsit de millorar els resultats.

² La investigació bàsica comprèn tots aquells treballs originals que tenen com a objectiu adquirir coneixements científics nous sobre els fonaments dels fenòmens i fets observables amb l'objectiu de formular hipòtesis, teories i lleis.

³ Consisteix en treballs originals que tenen com a objectiu adquirir coneixements científics nous, però orientats a un objectiu pràctic determinat. Els resultats solen ser susceptibles de ser patentats.

tecnològic fins a acabar amb el màrqueting aplicat al llançament de la novetat al mercat (Rossenger, 1980). Aquest model teòric permet explicar la complexitat de l'activitat d'innovar segmentant etapes d'un procés que se suposa seqüencial. A la realitat empresarial però, no necessàriament succeeix d'aquesta forma, "la investigació no és la font directa de les innovacions, moltes innovacions es produeixen partint de molt poca o gens d'investigació. ... La primera referència per als processos innovadors no és la recerca sinó la totalitat del coneixement humà acumulat". A més, les idees per innovar poden originar-se en qualsevol departament de l'empresa, no necessàriament des del de R+D. La figura que es pot veure a continuació mostra un model més real del procés d'innovació (Escorsa, P; Valls, J. 2004)

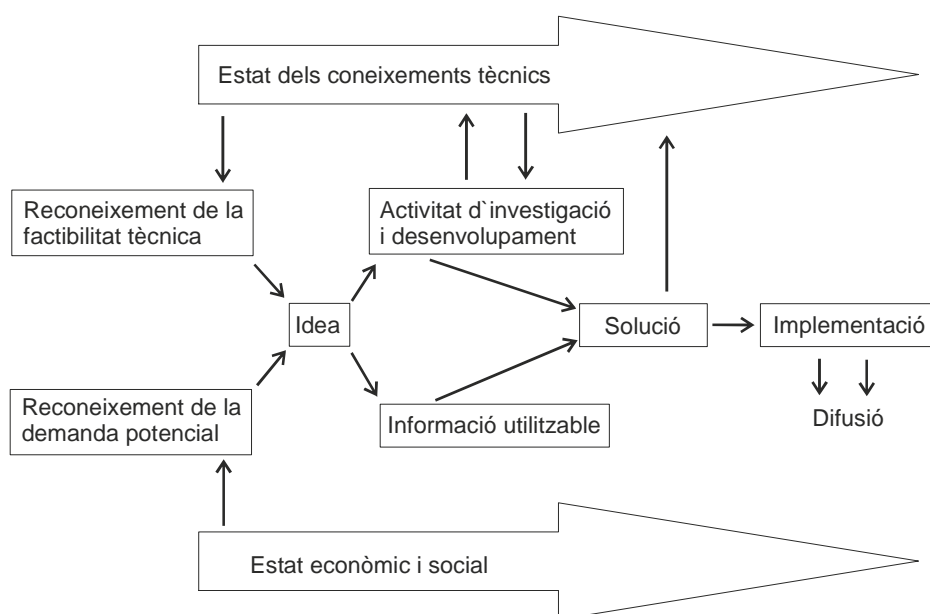


Figura 1-1 El procés de la innovació tecnològica segons Marquis.
Elaborat a partir de les aportacions de Gruber i Marquis (1969), Myers i Marquis (1969), Marquis (1969) i Utterback (1969, 1971^a, 1971, b).

Segons la importància de la innovació, es pot diferenciar entre les innovacions radicals, que suposen un trencament sobtat respecte l'estat anterior, i les innovacions incrementals, constituïdes per millores en els productes o processos ja coneguts. En general, les innovacions radicals tenen l'origen en el progrés de la ciència i la tecnologia "science push o technology push" i produeixen millores espectaculars en els resultats empresarials, mentre que les incrementals són degudes a les necessitats del mercat "demand pull" i tendeixen a una reducció dels costos empresarials.

En qualsevol cas, innovació és sinònim de canvi, evolució, fer coses noves, oferir nous productes i adaptar-los, posar a punt nous processos de la cadena de valor. La innovació és una competència bàsica de qualsevol organització, el seu ús ha de ser

sistemàtic per aprofitar l'oportunitat que ofereixen els canvis en la societat, en l'economia, en la demografia i en la tecnologia (Drucker, P, 1986).

1.4. Estructura de la memòria

Aquesta memòria està dividida en cinc capítols molt diferenciats, els quals es descriuen breument a continuació.

El primer capítol consisteix en la introducció i amb ella s'intenta situar el lector al projecte. També s'hi exposen els objectius, així com el marc conceptual de la importància de la innovació a l'actualitat.

Al segon capítol s'expliquen les bases del projecte descrivint amb profunditat el seu entorn. S'exposen també les necessitats a les que es pretén trobar solució, l'esquema del sistema proposat i el seu funcionament amb una breu descripció dels elements que el formen. Finalment s'exposa la metodologia utilitzada per portar a terme aquest projecte.

El tercer capítol és al que s'exposa el funcionament del sistema que es proposa en aquest projecte. S'explica tot allò que un possible usuari necessita saber per poder-lo utilitzar. Per fer-ho, es descriu detalladament l'element més important del sistema, l'aplicació mòbil i de la resta d'elements se'n descriu breument la seva tasca.

El capítol quatre exposa tota la informació tècnica del sistema. De cadascun dels elements (de l'aplicació per dispositius mòbils, de la base de dades, de l'aplicatiu servidor i dels verificadors) se'n descriu el disseny, el desenvolupament i el funcionament intern. A l'últim apartat d'aquest capítol s'hi pot trobar una descripció molt detallada dels diferents tipus de comunicació que hi ha entre aquests elements.

Finalment, a l'últim capítol es presenten els resultats del projecte i, s'exposa la informació de la prova pilot que s'ha portat a terme. També s'hi descriuen una llista de possibles ampliacions que en un futur es podrien portar a terme al sistema. Per acabar, les conclusions tanquen la memòria.

2. Bases del projecte

En aquest apartat s'expliquen de forma general els aspectes més bàsics d'aquest projecte. Primerament es descriu l'empresa per la que s'ha portat a terme i concretament, els processos en els que es presenten les necessitats i mancances. A continuació s'explica el projecte en tot el seu entorn, descrivint-ne les parts i el funcionament. Finalment s'exposa la metodologia utilitzada per portar a terme el projecte.

2.1. Empresa

Des del 1983 JCM Technologies desenvolupa, crea i comercialitza productes i serveis electrònics per a l'automatització i el control d'accés d'aplicacions residencials, comercials i industrials.

L'objectiu principal de l'empresa és clar, la satisfacció del client, fet que s'aconsegueix realitzant un esforç constant per aportar, tan als seus clients directes com als usuaris finals, solucions fàcils, pràctiques i sobretot, segures.

El desenvolupament d'aquests productes és realitzat íntegrament per l'empresa, fet que unit a la qualitat i a la innovació que els caracteritzen, permet a l'empresa ser competitiva al seu sector.

El procés productiu es divideix en dues fases realment diferenciades.

- La primera fase consisteix en el muntatge dels components de la part electrònica dels diferents productes. Aquesta fase s'anomena muntatge electrònic

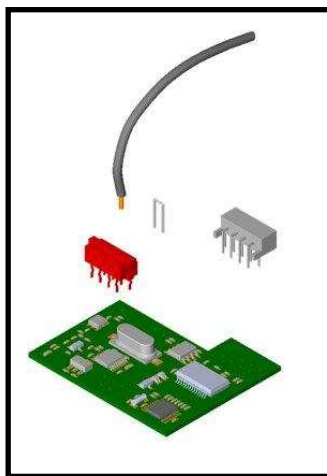


Figura 2-1 Simulació del muntatge electrònic

- La segona fase consisteix en incorporar a les plaques electròniques realitzades a la primera fase els elements de personalització demanats pels clients. Aquests elements poden ser els plàstics, l'embalatge, el programa que porten internament, etc. Aquesta segona fase s'anomena procés electromecànic.

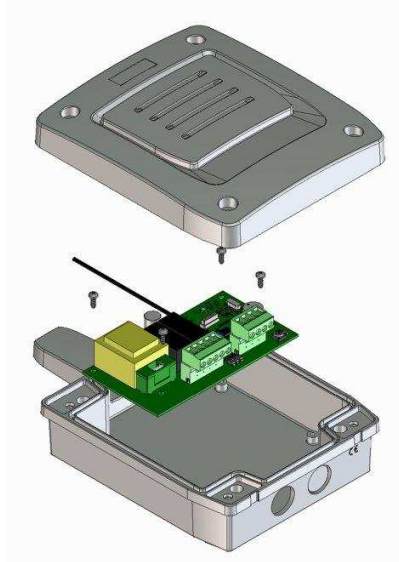


Figura 2-2 Simulació del procés electromecànic

Aquest projecte prèn part en la primera fase del procés productiu, el qual es subcontracta a empreses nacionals i internacionals. Perquè aquestes empreses puguin produir els diferents productes de manera correcta, JCM Technologies els aporta tota la informació necessària, així com la descripció detallada de cadascun dels components que els conformen. Mitjançant aquesta informació, l'empresa subcontractada pot comprar els components i muntar-los. En el moment en el que els equips han estat muntats, aquests són verificats i en el cas de ser aptes, comprats per JCM Technologies.

Actualment la situació dels mercats i l'alta competència fa que el procés productiu sigui un dels punts on més èmfasi es posa per intentar reduir el preu de cost dels productes. El preu de la mà d'obra és un paràmetre que varia força depenent del país i continent al que es porta a terme la producció. Aquest fet fa que empreses com JCM Technologies subcontractin a empreses internacionals la fase de muntatge electrònic del procés productiu dels seus productes.

La segona fase del procés productiu, la del procés electromecànic és portat a terme per la pròpia empresa JCM Technologies. Aquest fet es deu a que es faria molt complicat subcontractar-lo degut a que és un procés molt específic i diferent per cada producte de cada client.

2.1.1. El control de la qualitat en el procés de producció

Com s'ha comentat al punt anterior, l'objectiu principal de l'empresa és la satisfacció del client, per tant, cal comprovar que tots els equips que s'envien als clients funcionen correctament. És clar que quan es compra un producte (ja sigui per part d'una empresa o d'un particular) s'espera que aquest funcioni correctament i a poder ser, que ho faci la major quantitat d'anys possible.

Durant el procés electromecànic, abans d'embalatjar els equips per ser enviats, mitjançant un control de qualitat se'n comprova el bon funcionament. Aquest fet permet que equips defectuosos no arribin al client.

Un control de qualitat unitari i específic per cada tipus de producte és el que JCM Technologies demana a les empreses subcontractades que porten a terme la primera fase de la producció. D'aquesta manera assegura que tots els equips comprats s'han produït de manera correcta i el seu funcionament és l'adequat.

Aquest control de qualitat consisteix en un verificador que s'encarrega de comprovar elèctricament el bon funcionament de l'equip. L'operari ha d'agafar l'equip a verificar, situar-lo a l'emplaçament del verificador i prémer el polsador d'inici de test. En aquest moment el verificador comença a fer les comprovacions pertinents a l'equip i un cop finalitzades en dóna el resultat. Aquest activa un llum verd en el cas que l'equip hagi superat el test satisfactòriament, i un llum vermell en el cas d'haver-li detectat algun error. En aquest últim cas el verificador informa de l'error detectat a l'operari, el qual intentarà resoldre per tornar a provar de verificar l'equip. En el primer cas per la seva part, l'equip es considera apte per ser venut.



Figura 2-3 Imatge d'un verificador

JCM Technologies s'encarrega de dissenyar, crear i posar en marxa a les empreses subcontractades aquests verificadors. Aquest fet li assegura la correcta comprovació de tots els paràmetres necessaris pel bon funcionament dels equips. Cada tipus d'equip electrònic dissenyat per l'empresa disposa del seu verificador ja que tant

els enclatges com les comprovacions a fer, varien. En aquest punt és on l'autor d'aquest projecte pren part ja que té la responsabilitat de dissenyar, desenvolupar i fer el manteniment d'aquests verificadors.

Cal remarcar també que aquesta comprovació o control de qualitat es porta a terme a tots i cadascun dels productes que es compren. D'aquesta manera la possibilitat que productes defectuosos arribin al client final, desapareix.

2.1.2. Aspectes a millorar en el procés de verificació

Actualment els errors detectats durant la verificació de la primera fase de la producció només els coneix la persona encarregada de portar-la a terme. Ella introdueix cadascun dels equips al verificador, controla el resultat que aquest retorna i en llegeix els errors aportats en el cas que l'equip no sigui apte.

Portar a terme tot aquest procés pot semblar senzill i mecànic, però quan cal verificar milers d'equips és pràcticament impossible recordar amb una mínima exactitud la informació obtinguda per part del verificador. Informació com la quantitat d'equips correctes, la quantitat d'incorrectes i el motiu pel que aquests no són correctes és una informació molt valuosa de la que fins al moment no se'n podia disposar.

En el moment en el que es va detectar que hi havia la possibilitat de disposar d'aquesta informació a temps real i que es podia emmagatzemar de manera correcta, JCM Technologies va creure que calia invertir-hi. Actualment disposar d'informació, no d'una gran quantitat, sinó de la òptima i necessària en el moment precís, pot permetre i ajudar a una empresa a marcar la diferència.

A continuació es descriuen aspectes que calia millorar i pels que calia portar a terme aquest projecte.

- Un dels motius més importants era per la ràpida aportació d'informació que podria oferir als enginyers responsables de la producció dels diferents productes. Sovint, les persones encarregades de portar a terme la verificació dels diferents equips no disposa dels coneixements tècnics ni de la informació necessària per entendre cadascun dels imprevistos que pugui sorgir durant aquests dos processos. Aquest fet fa que cada vegada que en algun pas de la verificació quelcom surti de la normalitat, l'enginyer hagi de desplaçar-se fins on es verifica l'equip per buscar-hi una solució.

Mitjançant un sistema de transmissió d'informació eficient, la informació a la que podria accedir l'enginyer responsable seria la que en aquell precís moment s'estigués obtenint a la fàbrica de producció. D'aquesta manera, des de qualsevol lloc del món podria fer un diagnòstic correcte i donar la solució més òptima a la persona que està portant a terme la verificació.

- Un altre motiu important és que actualment, en el cas d'haver de canviar un component electrònic d'un producte (tan sigui perquè el proveïdor deixa de fabricar-lo, perquè el preu incrementa, o perquè es creu que un altre pot funcionar millor), no es coneixen els efectes secundaris del canvi fins que aquest no es troba instal·lat a casa d'un client final.

Es necessitaria que durant la verificació dels equips es poguessin conèixer els resultats i aquests poguessin ser comparats amb els obtinguts en produccions anteriors. D'aquesta manera, es podria detectar si el nou component té algun tipus d'efecte negatiu.

Conèixer els resultats en temps real, és a dir, en el precís moment de la producció té un gran valor ja que en el cas d'obtenir resultats negatius, com abans es pugui parar la producció per solventar el problema, menys despeses innecessàries tindrà l'empresa, ja que com abans es detecta un problema, més ràpid i econòmicament se li dona solució.

- Fins al moment l'empresa contractant no té manera de saber amb certesa si l'empresa subcontractada ha portat a terme o no la verificació a tots els productes entregats.

Es necessitaria conèixer de manera fàcil la quantitat d'equips que s'han verificat entre certes dates, així com els seus resultats obtinguts. D'aquesta manera doncs, comparant la quantitat d'equips verificats amb la quantitat d'equips rebuts es pot saber si han estat tots comprovats o no.

- Disposar d'un sistema amb una base de dades amb una estructura adequada per emmagatzemar un volum considerable d'informació, alta capacitat de processament i sortida gràfica d'informació, permetria obtenir estadístiques sobre els errors més freqüents a la verificació dels productes. Fet que permetria conèixer les debilitats dels productes, poder-los anar millorant.

2.2. Descripció del projecte

Per millorar i/o donar resposta als diferents aspectes esmentats a l'apartat 2.1.2. s'ha optat per crear una aplicació per dispositius mòbils amb sistema operatiu *Android*. Aquesta aplicació permet conèixer de manera fàcil i ràpida els resultats de totes les verificacions portades a terme fins al moment, permetent d'aquesta manera disposar d'informació valuosa en qualsevol lloc i en qualsevol moment.

Perquè tot el sistema pugui ser una realitat ha calgut crear varis elements que proveeixen a l'aplicació d'informació. Aquests elements i la pròpia aplicació es descriuen a continuació.

- **Aplicació per dispositiu mòbil.** Mitjançant l'aplicació pel sistema operatiu *Android* que s'ha creat, l'usuari es pot donar d'alta i pot consultar tota la informació sobre els equips que han estat verificats fins aquell moment.
- **Base de dades.** És l'element en el que es guarda tota la informació referent a l'usuari, als productes, a les diferents comprovacions que cal fer a cada producte i als resultats de cadascuna de les comprovacions que s'han portat a terme.
- **Software del verificador.** És un paquet de software que cal introduir en cadascun dels verificadors perquè aquests, cada vegada que hagin verificat un equip enviïn el resultat perquè aquest sigui guardat a la base de dades.
- **Aplicatiu servidor.** És l'element que permet la comunicació entre els diferents elements que formen el sistema. La seva tasca consisteix en donar resposta a les peticions que li fan l'aplicació i el verificador, buscant o introduint informació a la base de dades.
- **Internet.** Pren una part molt important en el sistema ja que és el mitjà pel que es porten a terme totes les comunicacions entre els diferents elements.

2.3. Esquema de funcionament del sistema

El sistema descrit a l'apartat 2.2, explicat de forma cronològica, és a dir, des de que es genera la informació verificant equips, fins que aquesta és visualitzada per l'aplicació mòbil d'un usuari, passa pels passos que es descriuen a continuació.

1. Per cada equip electrònic muntat per l'empresa subcontractada, el verificador s'encarrega d'enviar el resultat de cadascuna de les comprovacions que ha portat a terme cap a l'aplicatiu servidor mitjançant Internet.
2. En el moment en el que l'aplicatiu servidor rep una petició d'entrada d'informació cap a la base de dades, la recull, la processa extraient-ne la informació que necessita i la guarda a la base de dades.

Aquest procés descrit als punts 1 i 2 és el que més es porta a terme pel sistema ja que si se suposa que el procés de verificació d'un equip té una durada de 20s, en una jornada laboral de 8h, 1.440 vegades el verificador haurà enviat el resultat de la verificació cap a l'aplicatiu servidor. Cal tenir en compte que aquesta quantitat de peticions és referent a un sol producte, en el cas d'estar verificant un altre producte amb el seu verificador, el nombre de peticions es multiplicaria per 2.

En aquest punt es pot considerar que la base de dades ja té informació per poder ser mostrada en cas de ser consultada. Quan un usuari del sistema vol visualitzar els resultats obtinguts durant la verificació d'un cert producte, els passos a seguir són els següents.

3. L'usuari, ja donat d'alta anteriorment, mitjançant l'aplicació mòbil s'identifica i notifica que vol visualitzar els resultats d'un producte.
4. L'aplicació fa una consulta l'aplicatiu servidor demanant-li els productes als que aquest usuari té accés i els mostra a l'usuari perquè en seleccioni un.
5. L'usuari selecciona el producte en el que està interessat i introdueix les dades entre les que desitja obtenir la informació.
6. L'aplicatiu servidor recull la petició de l'aplicació mòbil i busca la informació demanada per l'usuari a la base de dades. Tan bon punt l'ha obtinguda, l'envia cap a l'aplicació.
7. L'aplicació mostra els resultats obtinguts per part de l'aplicatiu servidor en dos formats, numèrica i gràficament.

2.4. Metodologia del projecte

En aquest apartat es descriu el procés que s'ha portat a terme per arribar a fer realitat aquest projecte d'innovació, des dels requeriments i/o necessitats de l'empresa, fins al procés seguit per crear-lo. En tot projecte d'innovació, el punt de partida és comprendre l'entorn del projecte, determinar les necessitats/problemes a resoldre i establir els requeriments de l'empresa; en aquest marc es formulen les propostes d'innovació, s'avaluen les alternatives i es dissenya el procediment més adient per a concretar l'opció escollida.

2.4.1. Detecció de les necessitats

La primera cosa que es va fer una vegada escollit aquest projecte com a treball de final de màster, va ser la identificació de les necessitats que tenia l'empresa, les quals es descriuen a continuació.

- La necessitat més important de l'empresa consistia en poder disposar dels resultats de les verificacions a temps real. Aquest era el fet que li aportava més beneficis, ja que li permetria prendre decisions importants disposant de la informació necessària.
- Una altra necessitat important era que la visualització dels resultats havia de ser creada sobre una plataforma mòbil. Com s'ha comentat al punt 2.1., sovint la subcontractació de la primera fase de la producció es porta a terme a empreses internacionals. Aquest fet fa que degut al canvi horari la majoria de vegades que aquestes empreses tenen problemes, els treballadors de JCM Technologies no es troben al seu lloc de treball. Per aquest motiu, per aconseguir rapidesa a l'hora de detectar l'origen dels problemes o per resoldre'ls, era necessari que des de qualsevol lloc la informació pogués ésser obtinguda.

2.4.2. Requeriments de l'empresa

En el moment en el que es va plantejar portar a terme aquest projecte a l'empresa, aquesta no hi va acabar de detectar l'aportació de valor que podria oferir. Després d'estudiar les necessitats detectades amb una mica més de profunditat però, i de detectar els beneficis, es va decidir tirar-lo endavant. Per aconseguir aquests beneficis calia tenir en compte una sèrie de requeriments, els quals es detallen a continuació.

- Un dels requeriments més importants era que els resultats de les verificacions havien d'estar disponibles a temps real, és a dir, just després d'haver acabat la verificació de cada equip. El fet de tenir un resultat al finalitzar la verificació de tot un lot d'equips, no era vàlid. Aquest fet és degut a que el sistema aporta beneficis importants en el cas de detectar errors abans de la finalització de la producció. Una vegada produït tot un lot de productes, si en aquell moment es detecta que hi ha un problema, ja és massa tard ja que la reparació llavors és molt cara.
- No augmentar el temps de verificació de cap dels diferents productes, era un altre dels requeriments importants. Cal tenir en compte que una

petita variació del temps es transforma directament en un augment del preu de cost del producte.

- El sistema havia de ser ràpid i calia evitar haver de fer molts passos per poder obtenir els resultats. El fet d'obtenir sempre la informació desitjada de manera fàcil i ràpida és un dels aspectes pels que l'empresa lluita dia a dia per aconseguir.
- No tothom havia de poder accedir a la informació, algun tipus de filtre havia de ser aplicat. El sistema hauria de ser utilitzat pels treballadors de l'empresa, possiblement però, s'estudiaria la possibilitat d'oferir-lo als treballadors de les empreses subcontractades. Aquest fet feia que es requerís d'un sistema d'identificació dels usuaris.

2.4.3. Estudi de les possibilitats i elecció de la millor opció

Quan les necessitats i els requeriments estan ben definits el que cal fer és un estudi de les possibles opcions de les que es disposa per poder donar una resposta de la manera més adient possible.

Per portar a terme aquest projecte hi havia una sèrie d'aspectes que es tenien clars, els quals s'enumeren a continuació.

- Les comunicacions s'havien de fer a través d'Internet.
- La informació s'havia de guardar en una base de dades.
- Calia disposar d'un element encarregat de gestionar la base de dades.
- El programa existent dels diferents verificadors caldria modificar-los.

Un dels punts que va centrar més esforços de decisió va ser la visualització de la informació. La primera opció consistia en crear una aplicació per ordinador que permetés gestionar tot el sistema, donar d'alta nous usuaris, accedir a la base de dades i visualitzar la informació que aquesta contingés. El defecte d'aquesta opció però, era que no complia amb el primer punt de les necessitats de l'empresa. Tot i que actualment pràcticament tothom disposa d'un ordinador amb connexió a Internet a casa, la versatilitat que es buscava per aquest sistema era encara més elevada. Va ser en aquest moment quan es va detectar una opció que permetia disposar de la informació des de qualsevol lloc, i era mitjançant un telèfon intel·ligent. Aquesta opció consistia en crear una aplicació per aquest tipus de dispositiu que permetés portar a terme les opcions més importants, com donar-se d'alta al sistema i conèixer els resultats de les verificacions.

Una vegada escollida l'opció de l'aplicació per un dispositiu mòbil només calia escollir el sistema operatiu en el que aquesta havia de funcionar. És per tothom conegut

que hi ha 2 sistemes operatius que actualment estan marcant la diferència, *iOS*, de la reconeguda empresa *Apple*, i *Android*, de la també coneguda *Google*. Prendre la decisió de sobre quina de les dues plataformes desenvolupar l'aplicació no va ser massa complexa pels següents motius. Per disposar de l'entorn de desenvolupament d'aplicacions pel sistema operatiu *iOS* cal pagar, mentre que pel d'*Android* no. En una assignatura del màster s'havia estudiat el sistema operatiu *Android*, fet que el posicionava en avantatge. I per finalitzar, el telèfon mòbil de l'autor d'aquest projecte funciona amb el sistema operatiu *Android*, fet que permetria poder provar l'aplicació durant el desenvolupament d'aquesta i sobretot, utilitzar-la una vegada el sistema estigués implantat.

El fet d'haver de satisfer els requeriments de l'empresa va complicar una mica la decisió sobre com i quan calia enviar els resultats de les verificacions cap a la base de dades. Primerament es va prioritzar el fet de no augmentar el temps de verificació dels diferents productes, i es va pre-dissenyar el sistema perquè una vegada finalitzada la verificació de cada lot d'equips, el verificador enviés el resum dels resultats a la base de dades. Aquest fet feia que la comunicació entre el verificador i l'aplicatiu servidor només es portés a terme una vegada i d'aquesta manera, la durada d'aquesta comunicació no era crítica. Per una altra part però, es necessitava disposar dels resultats tan aviat com fos possible per poder disposar d'informació en temps real. Per aquest motiu calia enviar el resultat individualitzat per cada verificació.

Haver de satisfer aquests dos requeriments alhora era impossible. El que es va decidir desenvolupar va ser un sistema que tingués en compte els dos requeriments. Que el procés d'enviament dels resultats es portés a terme de forma individualitzada per cada equip, però que la durada d'aquest procés fos mínima.

2.4.4. Procés d'elaboració del projecte

En el moment en el que els requeriments, les necessitats van estar ben definides, el que es va fer va ser l'estudi de cadascun dels elements que formen el sistema de forma independent. Els únics elements del sistema que l'autor coneixia eren els verificadors i de manera relativa, el del desenvolupament de l'aplicació mòbil. La resta, el funcionament d'un servidor, la gestió d'una base de dades i les comunicacions entre els diferents programes mitjançant Internet, eren totalment desconeguts.

Aquest fet va fer que primer calgués conèixer cadascun dels diferents elements per separat, per després aconseguir que es comunicessin. A continuació es descriu aquest procés.

1. Disseny i desenvolupament de l'aplicació per dispositius mòbils.
2. Desenvolupament de l'aplicatiu servidor.

3. Disseny i desenvolupament de les comunicacions entre l'aplicació per dispositius mòbils i l'aplicatiu servidor.
4. Disseny i desenvolupament de la base de dades.
5. Disseny i desenvolupament de les comunicacions entre l'aplicatiu servidor i la base de dades.
6. Creació del nou programa pels verificadors.
7. Implementació de les comunicacions realitzades al pas 3 entre els verificadors i l'aplicatiu servidor.

3. Descripció funcional de la solució

En aquest apartat es descriu el funcionament del sistema creat. És a dir, tot el que un possible usuari d'aquest sistema ha de conèixer per poder treballar amb ell i treure'n el màxim rendiment.

Aquesta descripció es troba estructurada en quatre apartats, que són els quatre elements que formen el sistema i s'enumeren a continuació.

- **Aplicació per dispositiu mòbil.** És l'únic element del sistema amb el que l'usuari interactua directament i és des d'on pot visualitzar els resultats de les verificacions que necessiti.



Figura 3-1 Telèfon intel·ligent on cal instal·lar l'aplicació.

- **Base de dades.** És l'element on es guarda de manera estructurada tota la informació que conté el sistema.



Figura 3-2 Simulació de la localització de la base de dades.

- **Verificador.** Són els elements del sistema que es troben a l'empresa subcontractada i que porten a terme les verificacions als diferents productes. Una vegada fetes aquestes verificacions, n'envien el resultat cap al sistema perquè aquest la guardi a la base de dades i pugui ser mostrada per l'aplicació mòbil.



Figura 3-3 Verificador.

- **Aplicatiu servidor.** És l'element del sistema que permet la comunicació entre la base de dades i els altres dos elements del sistema, l'aplicació mòbil i els verificadors.

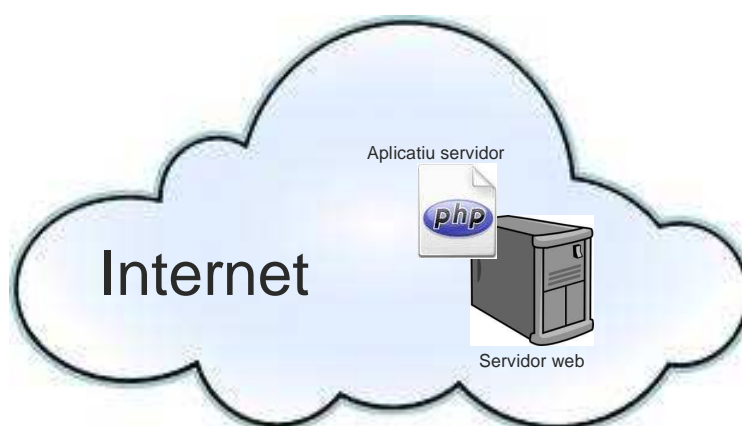


Figura 3-4 Simulació de la localització de l'aplicatiu servidor.

3.1. Aplicació per dispositius mòbils

L'aplicació mòbil és la part més important del sistema ja que n'és la cara visible per l'usuari, és a dir, és des d'on aquest interactua amb el sistema. Tasques com donar-se d'alta o visualitzar la informació que precisa sobre els resultats de la verificació dels diferents productes es porten a terme des d'aquest element.

Per poder portar a terme tota la seva funcionalitat, l'aplicació s'ha estructurat tal com s'explica a continuació.

3.1.1. Icona de l'aplicació

La icona d'una aplicació és l'element que es mostra a la llista d'aplicacions que un dispositiu té instal·lades i és sobre el que cal "clicar" quan es pretén executar-la. El *logo* de l'empresa JCM Technologies té forma de diana i s'ha aprofitat per utilitzar-lo d'icona d'aquesta aplicació. D'aquesta manera, intentant que la icona tingués una relació directa amb l'empresa, s'ha aconseguit que aquesta no passi desapercibuda entre la resta. Aquest fet permet que l'aplicació pugui ser detectada fàcil i ràpidament entre la resta d'aplicacions que hi pugui haver instal·lades al dispositiu.



Figura 3-5 Icona de l'aplicació.

3.1.2. Pantalla inicial

Just després de posar en funcionament l'aplicació, el primer que apareix a la pantalla són dos botons, els quals fan decidir a l'usuari entre dues opcions, donar-se d'alta d'un producte o consultar els resultats d'una verificació.

- Donar-se d'alta només es necessari en el cas que un usuari qualsevol, indiferentment de si ja està donat d'alta d'altres productes o no, es vol donar d'alta d'un producte nou.
- Consultar resultats és la opció més utilitzada ja que és la que permet conèixer de forma ràpida els resultats obtinguts en les verificacions dels diferents productes.

Depenent de les seves necessitats, l'usuari escollirà una opció o l'altra.



Figura 3-6 Pantalla inicial.

3.1.3. Donar-se d'alta d'un nou producte

Donar-se d'alta d'un nou producte significa que aquell usuari disposarà dels permisos per poder visualitzar els resultats de les verificacions del producte sobre el que es dona d'alta. En el cas que l'usuari esculli aquesta opció, la pantalla de la *Figura 3-7* serà la que li apareixerà.

Per poder-se donar d'alta d'un nou producte l'usuari ha d'introduir tota la informació que se li demana, la qual és la següent:

- **Nom.** És el nom de l'usuari. Cal tenir en compte que si ja està registrat al sistema per altres productes i pretén donar-se d'alta d'un de nou, ha d'introduir els mateixos caràcters introduïts anteriorment, amb les mateixes majúscules i minúscules. De no ser així el sistema entendreà que és un usuari nou i el registrarà com a tal.

- **Cognom.** És el cognom de l'usuari. Igual que passa amb el *Nom*, cal que l'usuari entri el cognom de la mateixa manera que ho va fer la primera vegada. Igual que en el cas del *Nom*, només cal tenir-ho en compte en cas d'estar ja registrat per altres productes.
- **Producte.** És el producte sobre el que l'usuari es vol donar d'alta i cal seleccionar-lo d'entre les diferents opcions que proposa l'aplicació. Per evitar errors a l'hora d'entrar el nom del producte, s'ha optat perquè aquest pugui ser escollit d'entre una sèrie d'opcions. Les opcions proposades són tots els productes que hi ha entrats al sistema.
- **Contrasenya:** Són una sèrie de caràcters alfanumèrics diferents per cada producte i que l'empresa que gestiona el sistema dóna només als usuaris que ho creu convenient. D'aquesta manera no tots els usuaris tenen accés a tots els productes que hi ha al sistema. Cada usuari només té accés a la informació dels productes amb els que treballa.

Aquest sistema de protecció de la informació també protegeix el sistema contra aquells individus que per les circumstàncies que fos, poguessin tenir accés a la descàrrega de l'aplicació. Aquests podrien instal·lar-se-la, però no es podrien donar d'alta de cap producte.



The screenshot shows a mobile application interface titled 'TesterDriver'. The main heading is 'TESTER DRIVER'. Below this, the instruction 'Omple els següents camps' is displayed. The form consists of the following elements:

- Nom:** A text input field with an orange border.
- Cognom:** A text input field.
- Producte:** A dropdown menu with 'Emissor' selected and a downward arrow.
- Password:** A text input field.
- Enviar Petició:** A button to submit the registration request.
- Resultat:** A label above a text area for displaying the result.

Figura 3-7 Pantalla on donar-se d'alta d'un producte.

Una vegada portat a terme aquest pas, l'usuari pot accedir a les dades del producte al que s'ha donat d'alta i als que ja estava donat d'alta amb anterioritat, si és el cas.

3.1.4. Consultar resultats

Consultar els resultats de la verificació d'un producte permet a l'usuari conèixer els resultats que s'han obtingut durant la verificació del producte que seleccioni. Per poder obtenir aquesta informació l'usuari ha d'omplir els camps *Nom* i *Cognom* per tal d'identificar-se, tal com es pot veure a la *Figura 3-8*.

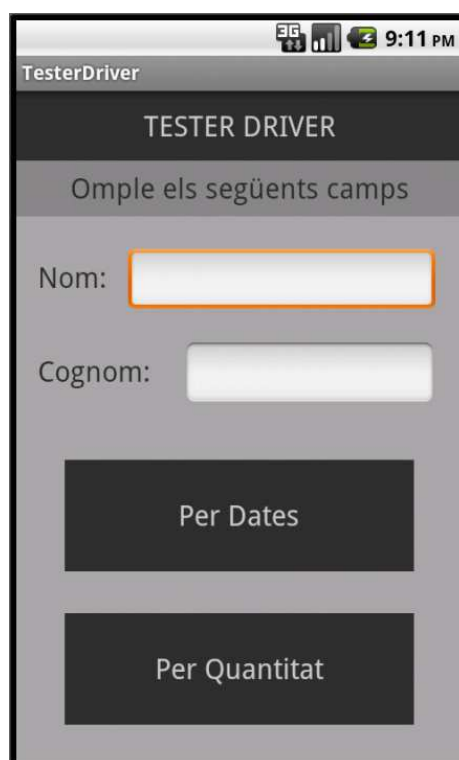


Figura 3-8 Pantalla on identificar-se i escollir el tipus de cerca.

Cal tenir en compte que a l'hora d'entrar els camps *Nom* i *Cognom* aquests han d'estar escrits de la mateixa manera que es va fer quan l'usuari es va donar d'alta del producte. De no ser així l'aplicació informarà que l'usuari entrat no existeix i no permetrà seguir endavant.

Una vegada omplerts els dos camps, cal seleccionar si la cerca la vol fer *Per Dates* o *Per Quantitat* mitjançant els dos botons que l'aplicació proporciona.

3.1.4.1. Per dates

La cerca de resultats per dates consisteix en obtenir els resultats de totes les verificacions portades a terme a un tipus de producte entre dues dates seleccionades. A la *Figura 3-9* es pot veure la pantalla que permet a l'usuari introduir tota aquesta informació a l'aplicació perquè aquesta pugui fer la cerca correctament. Tota la informació a entrar per part de l'usuari es descriu a continuació.

- **Selecció del producte.** Mitjançant un desplegable l'usuari pot seleccionar el producte del que vol conèixer els resultats de les verificacions. Aquesta llista només conté els productes als que l'usuari està donat d'alta.
- **Selecció data inicial.** Consisteix en entrar la data a partir de la que se'n vol conèixer els resultats.
- **Selecció data final.** Consisteix en entrar la data fins la que es volen obtenir els resultats.

Les dues dates entrades a aquestes dues opcions també es tenen en compte en el moment de fer la cerca. D'aquesta manera si es desitja consultar les verificacions portades a terme un dia en concret, caldria posar el mateix dia a la data inicial que a la final.

- **Botó *Veure Resultats*.** Una vegada el producte ha estat escollit i les dues dates han estat entrades de manera correcta, mitjançant el botó *Veure Resultats*, l'aplicació canvia de pantalla per mostrar la informació desitjada.



Figura 3-9 Pantalla on escollir producte i dates.

3.1.4.2. Per quantitat

La cerca de resultats per quantitat serveix per consultar els resultats de les últimes verificacions portades a terme. Si l'usuari entra un 100 al camp *Quantitat*, es visualitzaran els resultats dels últims 100 equips que hagin estat verificats del tipus de producte que hagi seleccionat. En aquest cas els passos a portar a terme per l'usuari són els que es descriuen a continuació.

- **Selecció del producte.** Mitjançant un desplegable l'usuari pot seleccionar el producte del que vol conèixer els resultats de les verificacions. Aquesta llista només conté els productes als que l'usuari està donat d'alta.
- **Introducció de la quantitat.** És el número de verificacions del que es mostraran els resultats.
- **Botó *Veure Resultats*.** Una vegada el producte ha estat escollit i la quantitat ha estat entrada, mitjançant el botó *Veure Resultats*, l'aplicació canvia de pantalla per mostrar la informació desitjada.

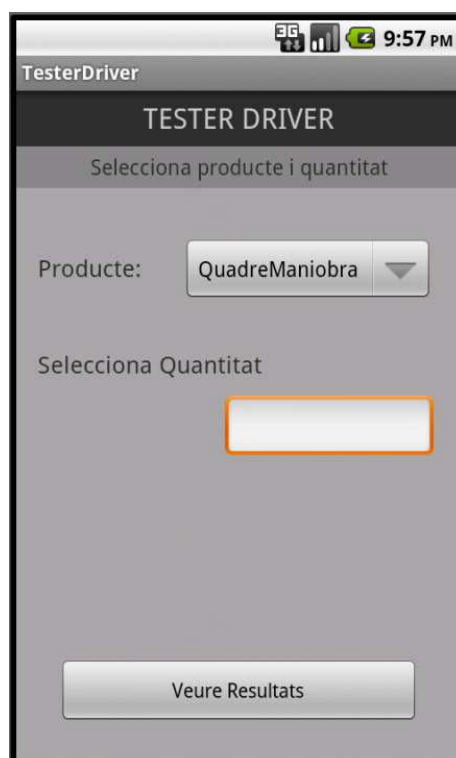


Figura 3-10 Pantalla on escollir producte i quantitat.

3.1.5. Visualització dels resultats

L'última pantalla de l'aplicació serveix per conèixer de manera ràpida i entenedora els resultats que es pretenien obtenir. A la part superior de la pantalla s'hi pot veure el nom de l'equip sobre el que s'estan mostrant els resultats i la quantitat total d'equips sobre la que s'està mostrant l'estadística. Finalment, ocupant la totalitat de la pantalla un gràfic permet visualitzar els resultats de les verificacions cercades.

Com es pot veure a la *Figura 3-11*, el gràfic és de sectors, fet que permet fer la visualització de manera fàcil i ràpida de quin és el resultat més obtingut en les verificacions, així com la tendència que segueixen les altres. Per fer-ho encara més entenedor s'ha optat per no posar llegenda, sinó que s'indiqui al propi gràfic a quina verificació fa referència cadascuna de les seves parts. Finalment, també s'ha afegit entre parèntesis la quantitat d'equips que han obtingut el tipus d'error que s'indica.

Tan si la cerca ha estat per dates com per quantitat, la pantalla de visualització de resultats és la mateixa. Com es pot comprovar en aquest cas s'ha optat per mostrar-la de manera horitzontal.

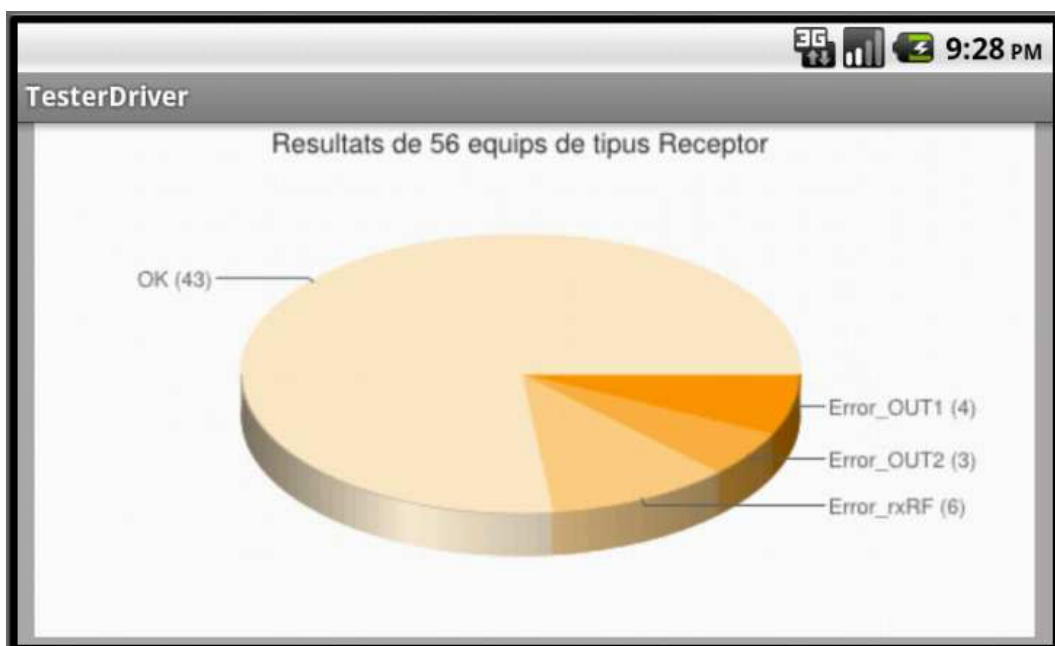


Figura 3-11 Pantalla on es visualitzen els resultats.

3.1.6. Estructura de les pantalles de l'aplicació.

En aquest punt es mostra l'estructura que segueixen les diferents pantalles que formen l'aplicació. Aquest fet permet conèixer de forma ràpida i entenedora els passos que cal seguir per arribar a obtenir els resultats d'un producte o perquè un usuari pugui donar-se d'alta d'un nou producte.

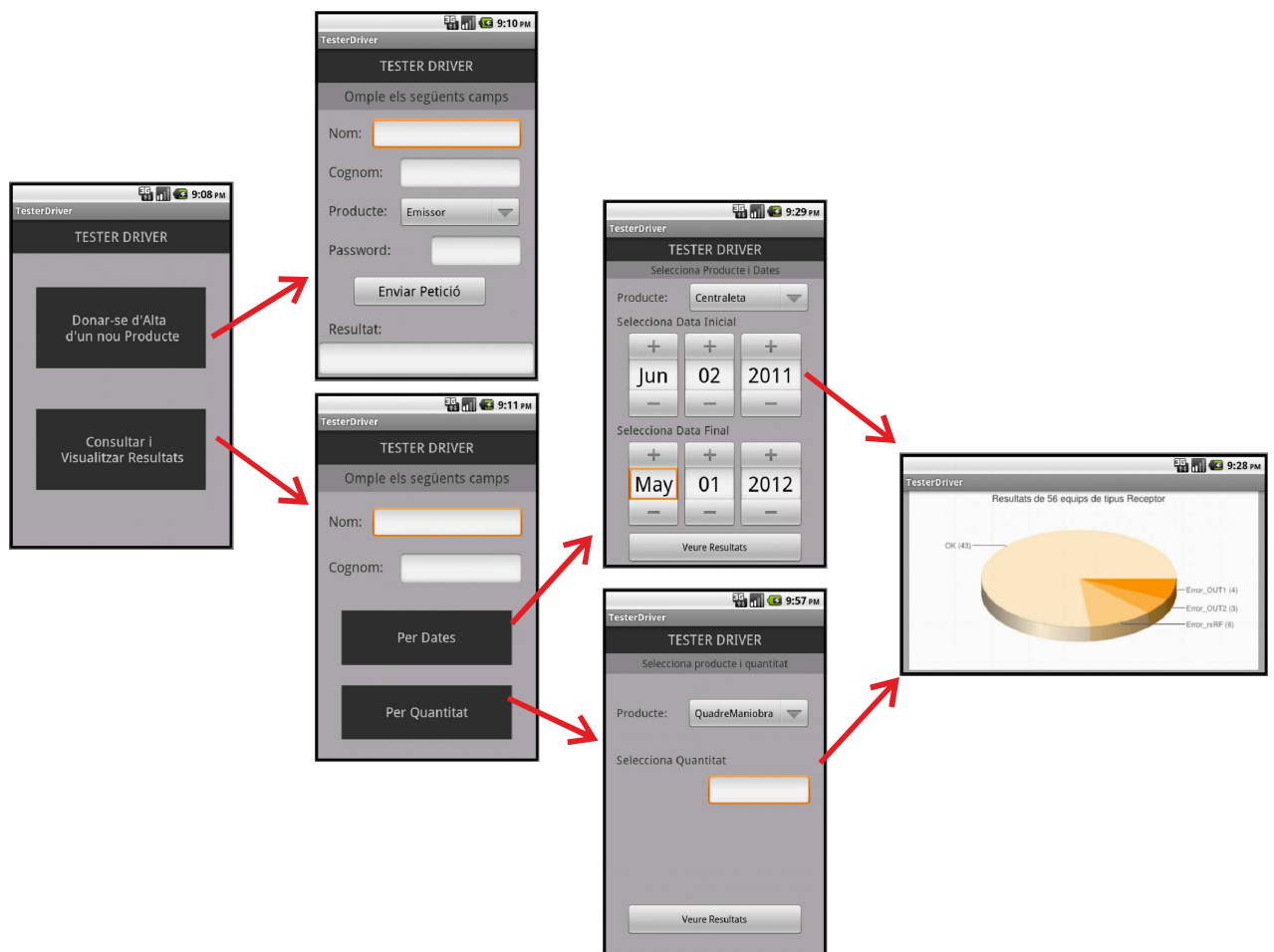


Figura 3-12 Estructura de pantalles de l'aplicació.

3.2. Base de dades

Una base de dades és un conjunt de dades organitzades segons una estructura coherent. Al sistema que conforme aquest projecte se n'ha creat una, la qual permet emmagatzemar tota la informació referent a:

- Diferents tipus de productes
- Diferents tipus de verificacions que es porten a terme als diferents productes
- Relació entre productes i verificacions
- Resultats de les verificacions
- Usuaris donats d'alta

En un sistema d'aquestes característiques és imprescindible fer ús d'una base de dades per tres motius.

- La gran quantitat d'informació que cal tractar.
- La flexibilitat que ofereix a l'hora de gestionar la informació.
- A la capacitat que ofereix per estructurar la informació segons quines siguin les necessitats.

Cal dir que perquè el sistema funcioni segons els requeriments demanats, aquesta base de dades es troba localitzada a un servidor web. Aquest fet és degut a la necessitat d'haver-se de trobar situada a Internet per així poder ser accedida des de qualsevol lloc des d'on es disposi d'aquest tipus de connexió.

Pel que refereix a l'usuari del sistema i a l'estructura funcional d'aquest, la informació descrita és tota la que cal conèixer. El tipus de base de dades, l'estructura d'aquesta i altres aspectes tècnics s'exposen a l'apartat 4.2..

3.3. Aplicatiu servidor

Un aplicatiu servidor és un element que proporciona serveis a altres sistemes informàtics dins d'una xarxa informàtica. Tant pot ser una màquina que conté una pàgina web que es visualitza mitjançant un navegador, com un programa sense interfície d'usuari que permet interactuar amb una base de dades (com és el cas d'aquest projecte).

En aquest sistema l'aplicatiu servidor té una tasca molt clara i concreta, que és la de fer d'enllaç entre la base de dades i els altres dos elements, l'aplicació mòbil i els verificadors. Sense aquest element la base de dades no es podria omplir ni podria ser consultada, però ell és l'encarregat que aquestes tasques siguin possibles.

L'aplicació mòbil per la seva part fa consultes a l'aplicatiu servidor perquè aquest busqui dins la base de dades la informació sol·licitada per l'usuari. Els verificadors, en canvi, envien la informació dels resultats de les diferents verificacions cap a l'aplicatiu servidor, perquè sigui aquest el que els emmagatzemi de manera correcta a la base de dades.

3.4. Verificador

El verificador és l'element creat per JCM Technologies que es troba a l'empresa subcontractada comprovant que els equips electrònics que es produeixen són aptes per ser comprats.

Cada tipus de producte, emissors, receptors, quadres de maniobra, etc. disposen del seu verificador, el qual només pot verificar-los a ells i a cap altre tipus de producte.

Els diferents verificadors estan formats per un *software* i un *hardware* encarregats de comprovar si els dispositius electrònics funcionen correctament o no. Per poder prendre aquesta decisió els verificadors porten a terme una sèrie de comprovacions als equips i una vegada finalitzades informen del resultat obtingut a l'operari que n'està al càrrec. Si totes han estat superades amb èxit, l'equip es dona com a correcte o vàlid. Pel contrari, si alguna de les comprovacions no ha estat superada amb èxit, aquell equip es rebutja i cal reparar-lo.

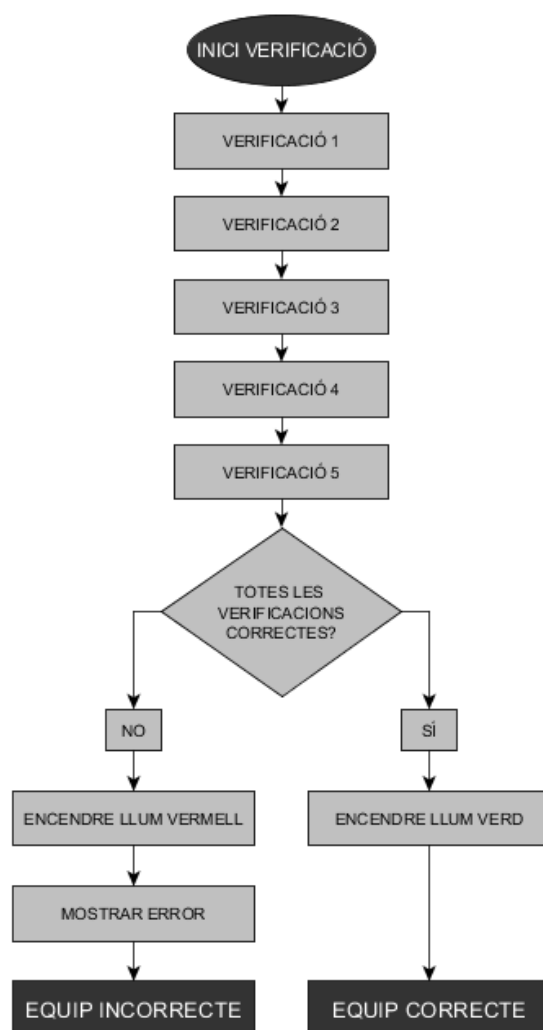


Figura 3-13 Diagrama de funcionament d'un verificador.

El que s'ha fet en aquest projecte és ampliar les tasques dels diferents verificadors perquè tinguin la capacitat d'enviar la informació del resultat de les verificacions a l'aplicatiu servidor. Com s'ha comentat a l'apartat 3.3., una vegada la informació es troba a l'aplicatiu servidor, aquest s'encarrega d'introduir-la a la base de dades.

Un aspecte important que cal tenir en compte perquè el sistema funcioni és que tots els verificadors hauran de tenir connexió a Internet. Si no és així no es podran comunicar amb l'aplicatiu servidor i la informació no podrà ser guardada a la base de dades.

4. Descripció tècnica de la solució

En aquest apartat es descriu tècnicament el sistema creat, és a dir, com s'ha portat a terme el procés de desenvolupament i com està estructurat internament perquè pugui portar a terme totes les tasques demanades.

Aquesta descripció es troba estructurada en els següents cinc apartats:

- Aplicació per dispositius mòbils
- Base de dades
- Aplicatiu servidor
- Verificador
- Protocol de comunicacions

A continuació, a la *Figura 4-1* es pot veure un esquema de tots els elements que formen el sistema, així com les diferents comunicacions que hi ha entre ells.

Com s'ha explicat en apartats anteriors, de verificadors i de dispositius mòbils als que hi hagi instal·lada l'aplicació n'hi ha més d'un al sistema, però per fer aquesta representació s'ha cregut que amb un de cada era suficient.

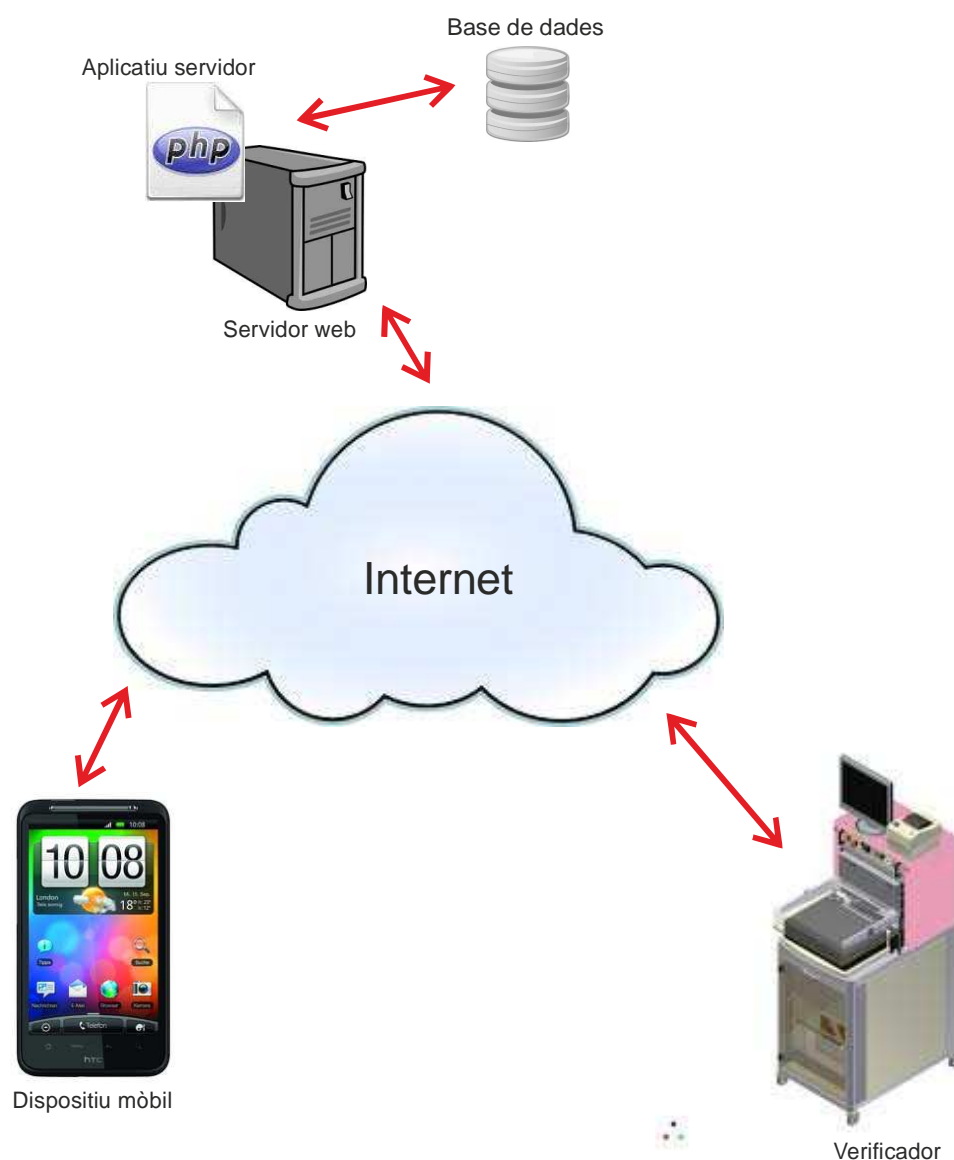


Figura 4-1 Diagrama del sistema

4.1. Aplicació per dispositius mòbils

Com ja s'ha esmentat en varis apartats anteriors, l'aplicació pel dispositiu mòbil que s'ha creat funciona sobre el sistema operatiu *Android*. En aquest apartat se n'explica tot el procés de desenvolupament.

4.1.1. Entorn i eines de desenvolupament

Les eines amb les que s'ha portat a terme l'aplicació mòbil d'aquest projecte són l'entorn de desenvolupament *Eclipse* i la *API 8 d'Android*. L'*Eclipse* és un entorn de desenvolupament que permet treballar amb diferents llenguatges de programació, entre ells el *Java*, que és l'utilitzat per desenvolupar aquesta aplicació. Aquest entorn, a més a més permet treballar amb les diferents *APIs d'Android*, les quals són imprescindibles per poder desenvolupar per aquesta plataforma. Aquestes *APIs* consisteixen en un conjunt de classes que venen incloses en el sistema operatiu i al referenciar-les, poden ser utilitzades des d'un altre programa. D'aquesta manera, des d'*Eclipse* es pot compilar un programa per la plataforma *Android* utilitzant totes les seves funcionalitats.

La versió de la *API* que es fa servir també és un paràmetre important, ja que depenent de la que s'utilitza, l'aplicació funcionarà sobre una sèrie de versions del sistema operatiu o unes altres. Per exemple, aplicacions desenvolupades amb la *API 4* poden funcionar sobre dispositius amb versions de sistema operatiu 1.6 o superiors. En el cas d'utilitzar la *API 8* les aplicacions podran funcionar sobre versions d'*Android 2.2* o superiors i sobre la 14, només poden funcionar en versions 4.0 o superiors.

Els motius pels quals s'ha escollit utilitzar la *API 8* són dos. Un és perquè la gran majoria de dispositius mòbils amb aquest sistema operatiu disposen d'una versió 2.2 o superior, per tant l'aplicació podrà funcionar-hi perfectament. Mentre que la segona és que el dispositiu del que es disposa per desenvolupar l'aplicació i fer-hi totes les proves porta aquesta versió de sistema operatiu, per tant, l'elecció de desenvolupament sobre aquesta *API* va ser molt simple.

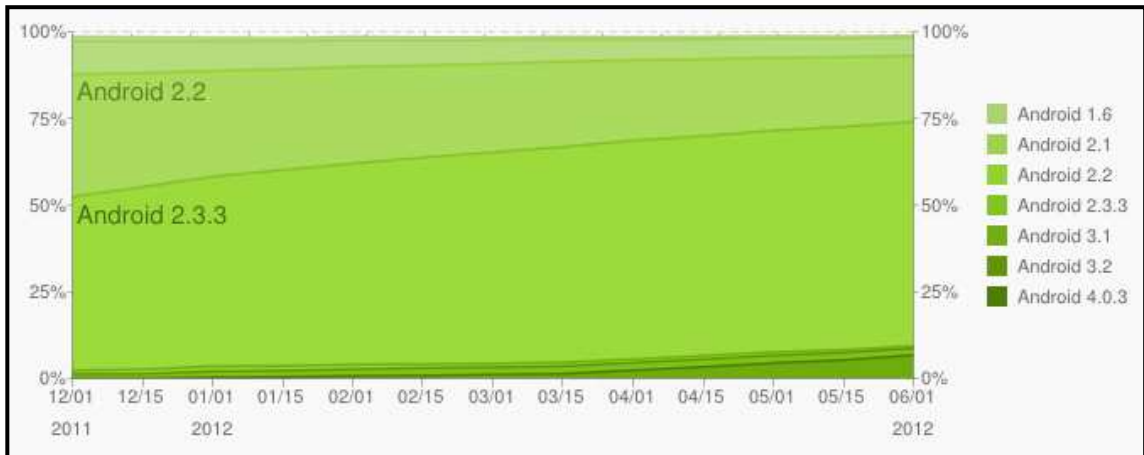


Figura 4-2 Gràfica de les versions d'*Android* que utilitzen els dispositius actualment.

4.1.2. Introducció teòrica a la programació *Android*

Una aplicació *Android* es forma mitjançant la unió de diferents *components*, que es poden "cridar" i executar de manera independent tant des de la pròpia aplicació a la que pertanyen, com des d'altres aplicacions.

Existeixen els següents quatre tipus de *components*:

- **Activity.** Un objecte del tipus *activity* s'encarrega de presentar una interfície visual amb la que l'usuari interacciona amb l'aplicació. Per exemple, una *activity* es pot encarregar de presentar a l'usuari una llista de missatges, una llista d'imatges, un mapa, etc. Una aplicació pot incloure varies *activities*. Els objectes de tipus *activity* es poden mantenir independents entre ells o poden comunicar-se.

El contingut visual d'una finestra activada per una *activity* està definit per una jerarquia de vistes. Una vista és un objecte de tipus *View* que s'encarrega de gestionar una àrea rectangular dins d'una finestra. La manera com es disposen els elements dins d'una *View* es defineix al *layout*. Una *Activity* habilita una *View* que té un *layout* assignat mitjançant la instrucció `Activity setContentView()`.

Els diferents estats en els que es pot trobar una *Activity* es poden veure a la *Figura 4-3*.

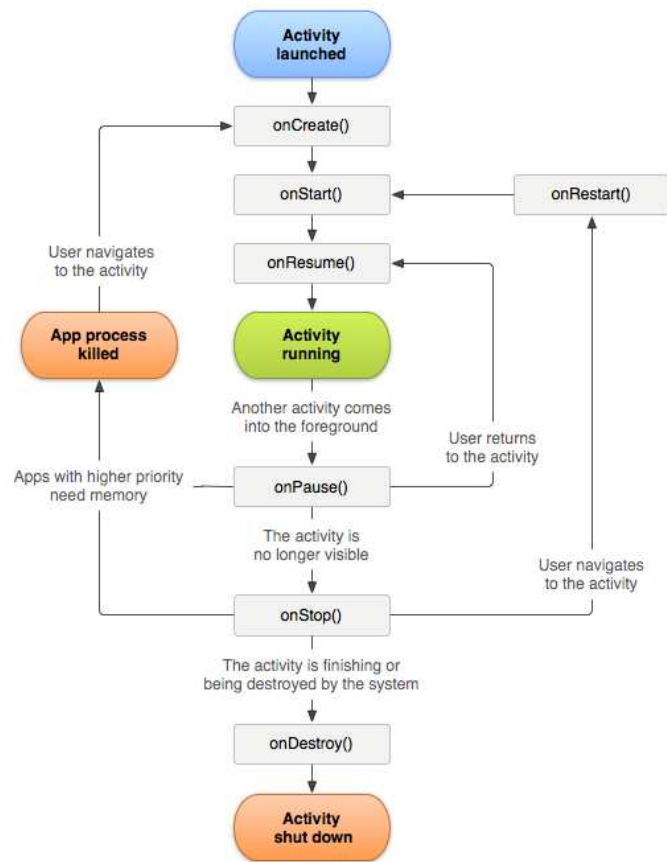


Figura 4-3 Cicle de vida d'una *Activity*.

A l'aplicació mòbil d'aquest projecte d'*activities* se n'ha creat una per cadascuna de les diferents "pantalles" que es poden veure i és el component més utilitzat.

- **Services.** Els *services* també són elements funcionals, però no disposen d'interfície visual d'usuari. Aquests elements són els que permeten que una aplicació continuï funcionant (per sota), encara que l'usuari del dispositiu n'estigui visualitzant una altra.
- **Broadcast receivers.** Els *broadcast receivers* són objectes que es dediquen a respondre els missatges generats per l'aplicació. Estan destinats a detectar i reaccionar davant determinats missatges o events

generats pel sistema (per exemple “bateria baixa”, “SMS rebut”, “SD introduïda”) o per altres aplicacions.

- **Content providers.** Els *content providers* són objectes que permeten que certa informació de l'aplicació estigui disponible per altres aplicacions.

Altres elements importants en una aplicació *Android* són els que es defineixen a continuació.

- **Intents.** Els *intents* són elements que fan possible l'activació d'un objecte de tipus *activity*, *service*, o *broadcast receiver*. Els *intents* gestionen els missatges que demanen que una acció es porti a terme.

En el desenvolupament de l'aplicació d'aquest sistema, els *intents* s'han utilitzat per passar paràmetres d'una *activity* a una altra i també per activar-les.

- **El fitxer *AndroidManifest*.** És el fitxer on es guarda la informació de configuració inicial de l'aplicació. Abans que *Android* executi una aplicació, el sistema llegeix aquest fitxer per saber la quantitat i el nom d'*activities* que té l'aplicació, els permisos dels que disposa, etc.
- **Layout.** Els *layout* són els encarregats de situar els diferents objectes a la pantalla. Un *layout* pot contenir un altre *layout* al seu interior, ja que n'hi ha de diferents tipus depenent del format de la informació que es vulgui mostrar. Alguns exemples són: *LinearLayout*, *RelativeLayout*, *TableLayout* i *GridLayout* entre d'altres.
- **Widgets.** Els *Widgets* són objectes de tipus *View* que serveixen d'interfície per la interacció amb l'usuari. Exemples de *widgets* són: *buttons*, *checkboxes* i diferents tipus de requadres de text entre d'altres.

4.1.3. Programació de la pròpia aplicació

El desenvolupament i programació de l'aplicació es pot dividir en 2 parts que es poden dissenyar per separat però que una vegada acabada funcionen conjuntament. Aquestes dues parts són la interfície d'usuari (creació dels diferents *layouts*) i el propi

funcionament de l'aplicació. En aquest apartat s'hi localitzen aquests dos punts més un tercer que explica com es fa per visualitzar el gràfic de resultats.

4.1.3.1. Interfície d'usuari

La interfície d'usuari és la part de l'aplicació que l'usuari veu i amb la que aquest interactua. En aquesta aplicació la interfície d'usuari consta de les diferents pantalles que s'han pogut veure a les figures que hi ha a l'apartat 3.1..

Cadascuna de les diferents pantalles amb les que l'usuari es pot trobar s'anomena *layout*, el qual es dissenya independentment de la resta però seguint les necessitats de l'aplicació. Els diferents *layouts* es troben en format XML, que és el que s'utilitza per portar a terme el disseny o programació d'aquests.

Per veure el disseny i desenvolupament d'un *layout* es posa com a exemple el que permet als usuaris donar-se d'alta d'un producte, el qual engloba una gran quantitat de *widgets* diferents, permetent així conèixer moltes de les seves possibilitats.



The image shows a screenshot of a mobile application interface titled "TesterDriver". The screen displays a registration form with the following elements:

- Header: "TESTER DRIVER"
- Instruction: "Omple els següents camps"
- Form fields:
 - "Nom:" followed by a text input field with an orange border.
 - "Cognom:" followed by a text input field.
 - "Producte:" followed by a dropdown menu showing "Emissor".
 - "Password:" followed by a text input field.
- Button: "Enviar Petició"
- Label: "Resultat:" followed by a large empty text area.

Figura 4-4 Exemple de disseny d'un *layout*.

L'estructura d'un *layout* és molt simple. Primerament es defineix el tipus de *layout* amb el que es desitja treballar i seguidament, per cada *widget* que es necessita, aquest es defineix i se'n descriuen les propietats necessàries per complir els requeriments.

A continuació es descriu el desenvolupament de la *Figura 4-4*.

1. El primer que es defineix quan es crea un *layout* n'és el tipus. De tipus n'hi ha varis, des del més simple anomenat "LinearLayout", que situa els *widgets* un a sota l'altre en l'ordre que es descriuen, fins al tipus "RelativeLayout", que permet moure en totes direccions i situar els *widgets* com el desenvolupador desitgi.

Per portar a terme aquesta aplicació el tipus "RelativeLayout" ha estat l'escollit, ja que tot i ser més complexa de desenvolupar, l'apreciació de l'usuari final és molt més agradable.

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:background="#AAA"
    android:gravity="center_horizontal"
    android:orientation="vertical" >
```

Figura 4-5 Definició i propietats d'un *layout* del tipus *Relative*.

2. En el moment en el que s'han descrit totes les propietats necessàries del *layout*, es poden introduir tots els *widgets* desitjats.

En el cas d'aquest exemple, el primer *widget* que hi ha permet visualitzar text i que s'anomena *ViewText*. En aquest cas s'utilitza per indicar el nom de l'aplicació. La seva definició i propietats són les que es poden veure a la *Figura 4-6*.

```

<TextView
    android:id="@+id/txtViewTITOL_AltaUsuari"
    android:layout_width="fill_parent"
    android:layout_height="50dp"
    android:background="#333"
    android:gravity="center"
    android:text="@string/txtViewTITOL_AltaUsuari"
    android:textColor="#CCC"
    android:textSize="20dp" />

```

Figura 4-6 Definició i propietats d'un *widget* del tipus *TextView*.

- El text "Nom:" també és un *widget* del tipus *TextView* i la seva configuració es pot veure a la *Figura 4-7*. També es pot veure com al no ser el primer component que es defineix, cal indicar on va situat. En aquest cas es troba 20 píxels per sota el component descrit al punt 2 i a 15 píxels del marge esquerra del *layout*.

```

<TextView
    android:id="@+id/txtViewNomAltaUsuari"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/txtViewInfoAltaUsuari"
    android:layout_marginLeft="15dp"
    android:layout_marginTop="20dp"
    android:background="#AAA"
    android:gravity="left"
    android:text="@string/txtView_NomAltaUsuari"
    android:textColor="#333"
    android:textSize="20dp" />

```

Figura 4-7 Definició i propietats d'un altre *widget* del tipus *TextView*.

- Al costat dret del text "Nom:" s'hi troba un *widget* del tipus *EditText* el qual permet a l'usuari introduir-hi els caràcters desitjats. La seva configuració es pot veure a la *Figura 4-8*.

```

<EditText
    android:id="@+id/editTxtNomAltaUsuari"
    android:layout_width="220dp"
    android:layout_height="wrap_content"
    android:layout_marginLeft="80dp"
    android:layout_alignBaseline="@+id/txtViewNomAltaUsuari"
    android:inputType="text" >
</EditText>

```

Figura 4-8 Definició i propietats d'un *widget* del tipus *EditText*.

- Al costat del text "Producte:" s'hi localitza un *widget* del tipus *Spinner*. Aquest *widget* és un desplegable que permet visualitzar i seleccionar un element d'una llista. Les característiques d'aquest *widget* es poden veure a la *Figura 4-9*.

```
<Spinner
    android:id="@+id/spinner_ProducteOmplerDates"
    android:layout_width="180dp"
    android:layout_height="wrap_content"
    android:layout_alignBaseline="@+id/txtViewProducteOmplerDates"
    android:layout_marginLeft="120dp" />
```

Figura 4-9 Definició i propietats d'un *widget* del tipus *Spinner*.

- Finalment, a la meitat inferior de l'*Activity* s'hi pot veure un *widget* de tipus *Button* amb el text "Envia Petició". Aquest *widget* consisteix en un botó el qual l'usuari pot activar. La definició i les propietats d'aquest *widget* es poden veure a la *Figura 4-10*.

```
<Button
    android:id="@+id/btnEnviaPeticioAltaUsuari"
    android:layout_width="200dp"
    android:layout_height="wrap_content"
    android:layout_below="@+id/txtViewPasswordMaquinaAltaUsuari"
    android:layout_marginLeft="50dp"
    android:layout_marginTop="25dp"
    android:text="@string/btn_EnviaPeticioAltaUsuari"
    android:textSize="20dp" />
```

Figura 4-10 Definició i propietats d'un *widget* del tipus *Button*.

4.1.3.2. Programació de la funcionalitat

A continuació es descriuen les diferents parts que hi ha en la definició de cadascun dels objectes *Activity* que serveixen per mostrar cadascun dels *layouts* creats i donar-los funcionalitat.

1. Creació d'una classe que hereta de la classe *Activity*

La classe *Activity* és una classe que disposa de tots els mètodes i atributs per poder crear una finestra on visualitzar un *layout* i treballar amb tots els tipus de *widgets* que es puguin trobar en aquest *layout*. Crear una classe heretada de la classe *Activity* permet disposar de tots aquests mètodes i variables a més a més dels propis de la classe que es crea.

2. Declaració dels components de l'*Activity*

El que es porta a terme en aquest apartat consisteix en definir cadascun dels components que conformen el *layout* a la classe i es porta a terme de la següent manera.

```
//Declarem BUTTONs, EDITTEXTs i SPINNERs que hi ha en aquesta ACTIVITY
EditText ET_Nom, ET_Cognom, ET_Password, ET_ResultatPeticio;
Button btn_EnviaPeticio;
Spinner Spinner_Productes;
```

Figura 4-11 Declaracions dels components del *layout*.

3. Visualització del *layout*

Un dels mètodes més importants de la classe *Activity* és el mètode *onCreate()*. Aquest mètode és el que s'executa en el moment de la creació de l'*activity* i és el lloc en el que cal definir quin és el *layout* que es desitja mostrar. En el moment en el que s'executi el mètode *onCreate()* es visualitzarà a la pantalla del dispositiu el *layout* definit anteriorment i es podrà començar a treballar sobre ell.

```
/**
 * Called when the activity is first created.
 */
@Override
protected void onCreate(Bundle savedInstanceState)
{
    super.onCreate(savedInstanceState);
    setContentView(R.layout.alta_usuari);
}
```

Figura 4-12 Mètode *onCreate()*.

A la *Figura 4-12* es pot veure com quan es crida el mètode *onCreate()* el primer que es fa és definir quin és el *layout* a mostrar. En l'exemple que es visualitza a la figura, es visualitzarà el *layout* d'alta d'usuari, el qual s'ha descrit a l'apartat *interfície d'usuari* de l'apartat 4.1.3.1..

4. Definició dels diferents *widgets* del *layout* a l'*activity*

Dins el mètode *onCreate()*, just després d'indicar quin serà el *layout* a mostrar, cal definir a quins *widgets* del *layout* corresponen cadascun dels components de l'*activity* declarats al punt 2.

```

//*****
ET_Nom = (EditText)findViewById(R.id.editTxtNomAltaUsuari);
ET_Cognom = (EditText)findViewById(R.id.editTxtCognomAltaUsuari);
Spinner_Productes = (Spinner)findViewById(R.id.spinner_ProductesAltaUsuari);
ET_Password = (EditText)findViewById(R.id.editTxtPasswordMaquinaAltaUsuari);
ET_ResultatPeticio = (EditText)findViewById(R.id.editTxtResultatAltaUsuari);
btn_EnviaPeticio = (Button)findViewById(R.id.btnEnviaPeticioAltaUsuari);
//*****

```

Figura 4-13 Declaracions dels components del *layout*.

Aquest fet permet treballar amb la informació i funcionalitat dels *widgets* del *layout*, des de la classe creada, heretada de la classe *Activity*.

5. Programació de les funcionalitats dels components del *layout*

Hi ha varis mètodes pels quals una aplicació pot controlar els events generats des de la interfície d'usuari. En aquest projecte el mètode escollit ha estat el de reescriure els mètodes *callback* de l'objecte *View*.

```

//*****
//*** Quan premem botó d'enviar petició d'alta ****
//*****
btn_EnviaPeticio.setOnClickListener(new View.OnClickListener()
{
    public void onClick(View v)
    {

```

Figura 4-14 Reescriptura del *callback* de l'objecte *View*.

A la *Figura 4-14* es pot veure la reescriptura del mètode *callback*. Tot seguit, el codi que s'introdueixi després d'aquestes dues línies serà el que s'executarà en el moment en el que l'usuari premi el botó per donar-se d'alta d'un nou producte.

6. *AndroidManifest file*

En el moment de crear un projecte nou per poder desenvolupar una aplicació *Android* en l'entorn de desenvolupament *Eclipse*, aquest s'encarrega de crear el fitxer *AndroidManifest*. Com s'ha comentat a l'apartat *Introducció teòrica a la programació Android* aquest fitxer és de configuració i conté molta informació necessària pel sistema operatiu.

Per poder portar a terme l'aplicació d'aquest projecte, aquest fitxer s'ha hagut de modificar per dos motius.

- Primerament per indicar al sistema cadascuna de les *activities* que conformen l'aplicació.

A la *Figura 4-15* es pot veure com hi ha una *activity* que és la principal (la primera que es veu quan s'executa l'aplicació) i a continuació la resta, les quals només cal que siguin anomenades.

```
<activity
  android:name=".TesterDriverActivity"
  android:label="@string/app_name" >
  <intent-filter>
    <action android:name="android.intent.action.MAIN" />
    <category android:name="android.intent.category.LAUNCHER" />
  </intent-filter>
</activity>
<activity
  android:name=".AltaUsuari" >
</activity>
<activity
  android:name=".TriaDatesQuantitat" >
</activity>
<activity
  android:name=".BusquedaPerDates" >
</activity>
<activity
  android:name=".VeureResultats"
  android:screenOrientation="landscape">
</activity>
<activity
  android:name=".BusquedaPerQuantitat" >
</activity>
```

Figura 4-15 Llista d'activitats al fitxer *AndroidManifest*.

- El segon motiu pel que ha calgut modificar el fitxer ha estat per donar a l'aplicació els permisos que necessita poder funcionar tal com es demanava.
 - El primer que es necessita serveix per obtenir informació del dispositiu mòbil.
 - El segon serveix per permetre a l'aplicació tenir accés a Internet. De no ser així l'aplicació no disposaria de connexió de dades i no podria complir amb el seu requeriment principal.

```
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission android:name="android.permission.INTERNET" />
```

Figura 4-16 Permisos del fitxer *AndroidManifest*.

4.1.3.3. Creació i visualització del gràfic de resultats

En aquest punt es descriu l'eina de *Google* utilitzada per crear el gràfic i mostrar-lo a la pantalla de visualització de resultats. Aquesta eina s'anomena *Google Chart API* i permet la creació de gràfics en format d'imatges.

La seva utilització es porta a terme mitjançant peticions *http* a una determinada *url*, que és la *http://chart.apis.google.com*. A aquesta *url* se li afegixen els paràmetres que es desitgen que el gràfic contingui, que són els que es descriuen a continuació.

- **Dimensions del gràfic.** Mitjançant el paràmetre *chs* (*chart size*) es pot escollir la mida en píxels que es desitja que el gràfic tingui. En el cas del gràfic que es mostra a la pantalla de resultats, aquest té unes mides de 500 per 260 píxels.

Per informar de les dimensions del gràfic, el text que cal afegir a la *url* és "chs=500x260".

- **Tipus de gràfic.** Amb el paràmetre *cht* (*chart type*) es pot escollir el tipus de gràfic que es desitja, pel que fa a les dimensions. Les possibilitats que s'ofereixen són dues, de dues dimensions o de tres. En aquest cas, degut al millor aspecte que ofereix, l'escollit ha estat el de tres dimensions.

Per informar del tipus de gràfic, el text que cal afegir a la *url* és "cht=p3".

- **Conjunt de dades que formen el gràfic.** Mitjançant el paràmetre *chd* (*chart data*) s'envien les dades que el gràfic ha de mostrar. En aquest cas, tant els valors com la quantitat de valors són paràmetres variables, que cal gestionar molt bé a l'hora de col·locar-los.

Per informar de les dades que el gràfic ha de mostrar, un exemple del text que caldria afegir a la *url* és "chd=t:10,20,30".

- **Llegenda del gràfic.** Mitjançant el paràmetre *chl* (*chart label*) s'especifica la llegenda del gràfic. En aquest cas, s'ha escollit que la llegenda es trobés en forma d'etiquetes indicant a què corresponia cada part del gràfic. Com al cas anterior, la quantitat d'etiquetes i el valor d'aquestes és variable, per tant calia tractar-ho amb molta precaució. Per si això fos poc, l'ordre en el que s'entren les etiquetes correspon amb l'ordre en el que s'han entrat les dades al punt anterior. Per aquest motiu, el primer paràmetre que s'entra correspon al primer valor del punt anterior, el segon al segon i així successivament.

Per informar de les dades que el gràfic ha de mostrar, un exemple del text que caldria afegir a la *url* és "chd=t:Verificació1|Verificació2|Verificació3".

La generació i gestió de tots aquests paràmetres i molts d'altres, es porta a terme des de la classe heretada de la classe *Activity*, que permet veure els resultats.

1. Detecta el tipus de cerca que l'usuari ha escollit fer i en recull els paràmetres necessaris.
2. Fa la petició de la informació a l'aplicatiu servidor perquè aquest busqui la informació demanada per l'usuari a la base de dades.
3. Quan rep la resposta de l'aplicatiu servidor agafa les dades, les tracta i genera la *url* per *Google Chart API*.
4. Rep la imatge que *Google Chart API* li proporciona i la mostra a la pantalla mitjançant un *widget* del tipus *WebView*.

4.2. Base de dades

La base de dades d'aquest sistema és de tipus dinàmica, ja que la informació que s'hi emmagatzema és actualitzable i modificable. Ha estat portada a terme mitjançant el sistema de gestió de bases de dades relacionals *MySQL*, el qual ha esdevingut molt popular gràcies a la seva velocitat en l'execució de consultes. Aquest fet ha estat el motiu pel qual va ser escollit per prendre part en aquest sistema.

4.2.1. Disseny de l'estructura

Dissenyar una estructura òptima de la base de dades ha estat un dels processos més complexos del procés de creació d'aquesta. Cal que la informació sigui accessible de manera ràpida, que estigui ben estructurada, que hi hagi tota la que es necessita, que no estigui repetida i que no se'n guardi d'innecessària.

Per aconseguir tots aquests requeriments s'han hagut de fer fins a 3 versions d'estructura ja que les dues primeres no complien amb els objectius establerts.

4.2.1.1. Primera versió de l'estructura

La primera versió d'estructura que es va fer per la base de dades es va portar a terme amb la intenció que fos molt versàtil, per així poder-ne extreure el major rendiment possible. Per aconseguir aquest objectiu es va dissenyar una base de dades amb les següents 7 taules.

- **Taula d'usuaris.** Contenia la informació dels usuaris donats d'alta al sistema.
- **Taula de relació usuari-producte.** Contenia la relació dels diferents productes als que estava donat d'alta cada usuari.
- **Taula de famílies de productes.** Contenia la llista de les diferents famílies de productes que estaven donades d'alta al sistema.
- **Taula de relació producte-família.** Contenia la llista de productes donats d'alta al sistema i la relació amb la família amb la que pertanyien.
- **Taula de verificacions.** Contenia la llista dels diferents tipus de verificacions que es podien portar a terme als diferents productes.
- **Taula de relació família-verificació.** Contenia la relació de verificacions que es portaven a terme en cadascuna de les diferents subfamílies.
- **Taula de resultats.** Contenia els resultats de les diferents verificacions que s'anaven portant a terme als diferents equips.

L'estructura d'aquestes taules quedava tal com es pot veure a la *Figura 4-17*.

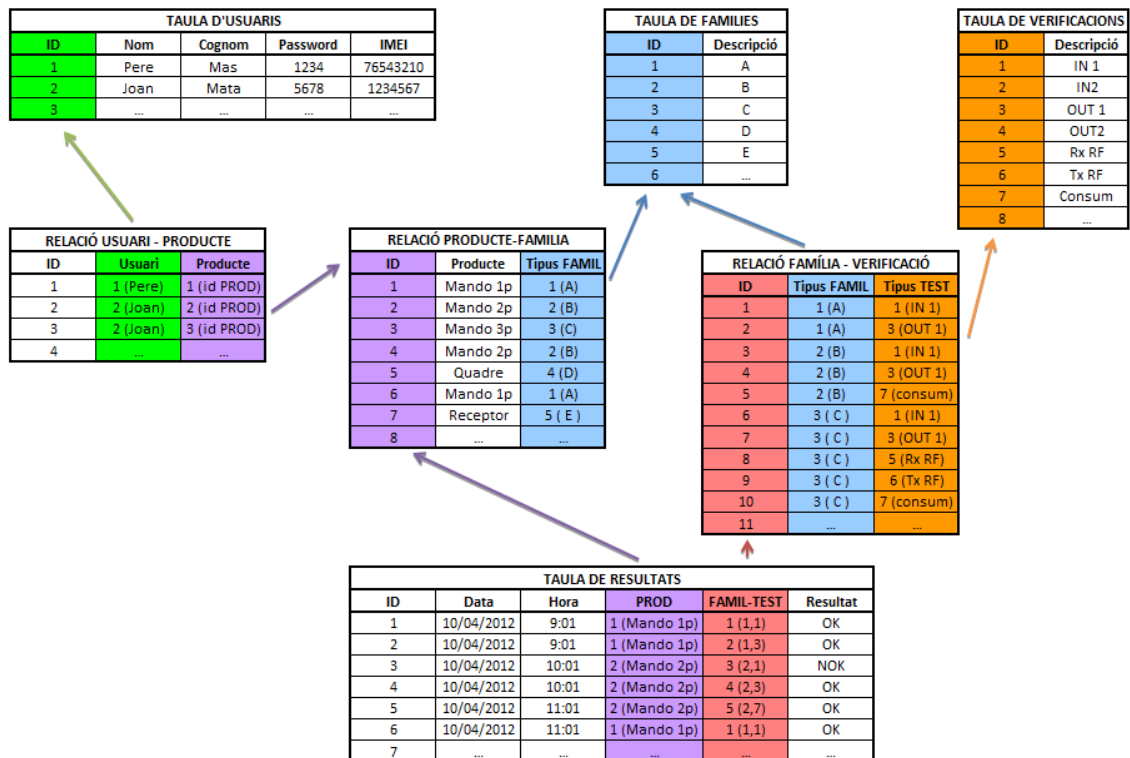


Figura 4-17 Primera proposta d'estructura de la base de dades.

Mitjançant aquesta estructura s'aconseguia que a més a més de poder fer estadístiques dels resultats de les verificacions per productes, també es poguessin obtenir estadístiques per famílies.

Un altre aspecte important que ofería aquesta estructura era que assegurava la impossibilitat que hi hagués incoherència entre les dades. És a dir, la informació estava estructurada de tal manera que feia impossible que s'introduïssin dades errònies o incoherents al sistema.

Una vegada creada l'estructura, a la fase de proves es va poder comprovar com els resultats eren positius, ja que s'obtenia la versatilitat i la seguretat que es buscava. El problema però, apareixia en el moment de fer les cerques i insercions, ja que es tardava més estona de la que es desitjava en obtenir els resultats. Aquest fet va crear la necessitat d'haver de buscar una altra estructura per la base de dades, la qual permetés cerques més ràpides.

4.2.1.2. Segona versió de l'estructura

La segona versió d'estructura que es va dissenyar per la base de dades es va portar a terme amb el clar objectiu de reduir la quantitat de taules i d'informació. D'aquesta manera el temps de cerca es veuria reduït considerablement.

Reduir el nombre de taules i d'informació feia impossible mantenir la versatilitat del sistema. El temps de cerca però, prevalia sobre qualsevol altre requeriment, per tant, la versatilitat es va haver de deixar perdre.

Per aconseguir el nou objectiu es va dissenyar una base de dades amb només les següents 5 taules.

- **Taula d'usuaris.** Contenia la informació dels usuaris donats d'alta al sistema.
- **Taula de relació usuari-producte.** Contenia la relació dels diferents productes als que estava donat d'alta cada usuari de la taula d'usuaris.
- **Taula de productes.** Contenia la llista dels diferents productes que estaven donades d'alta al sistema.
- **Taula de verificacions.** Contenia la llista dels diferents tipus de verificacions que es podien portar a terme als diferents equips i que estaven entrades al sistema.
- **Taula de Resultats.** Contenia els resultats de les diferents verificacions que s'anaven portant a terme als diferents equips.

L'estructura d'aquestes taules quedava tal com es pot veure a la *Figura 4-18*.

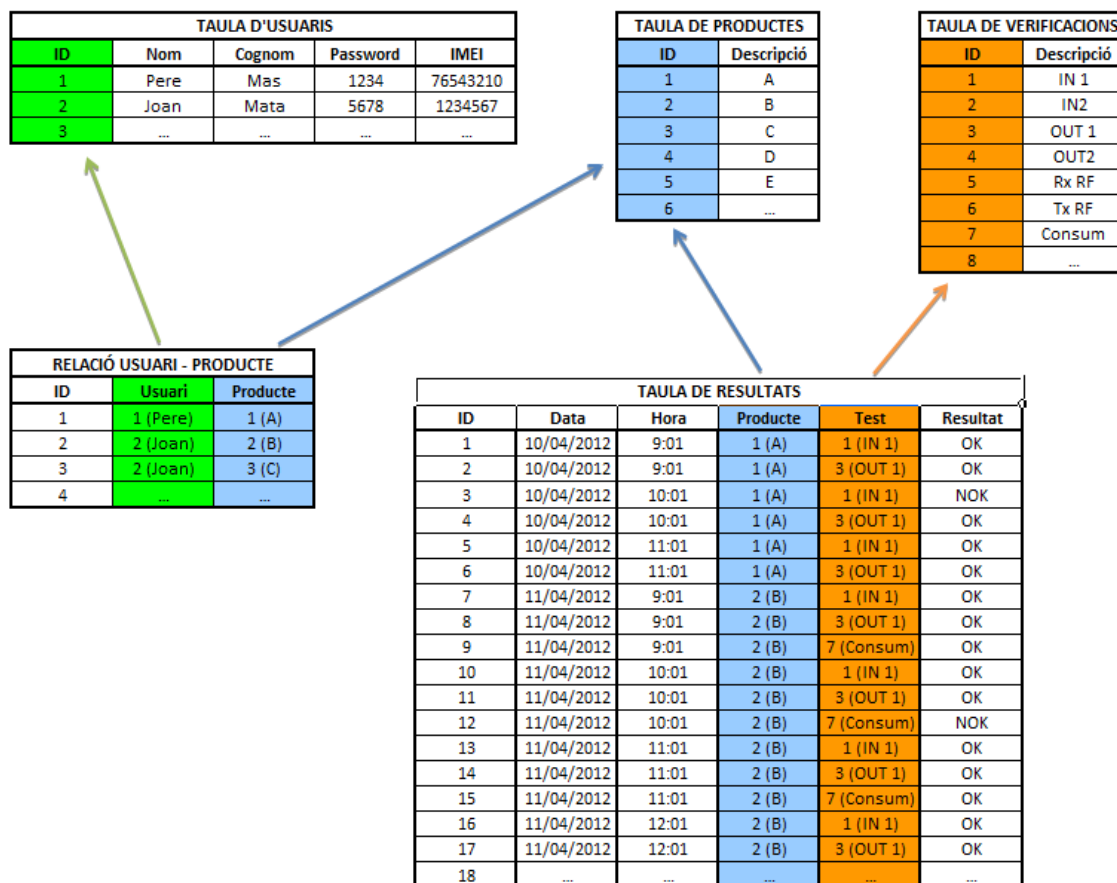


Figura 4-18 Segona proposta d'estructura de la base de dades.

Després de portar a terme les proves pertinents, tot i haver reduït molt els passos a fer durant la cerca i la informació a buscar, el temps que es tardava en obtenir els resultats de la cerca continuava sent massa elevat. Per aquest motiu va caldre portar a terme una tercera estructura.

4.2.1.3. Estructura definitiva

A la tercera versió de l'estructura que es va fer només es buscava reduir el temps de cerca d'informació. En aquell moment, després d'estudiar el tema amb profunditat i llegir bastanta documentació relacionada, es va creure que el problema no estava en la quantitat de taules, sinó en la quantitat d'informació que es contrastava a la taula més important. Trobar la relació de quins són els paràmetres a buscar a la *Taula de Resultats* només es porta a terme una vegada a l'inici de la cerca, però que per cada entrada d'aquesta mateixa taula s'hagin de contrastar més o menys paràmetres fa que el temps sigui molt variable.

Coneixent aquest fet es va crear una taula nova que permetia relacionar cada producte amb les seves verificacions, per d'aquesta manera, a la *Taula de Resultats* poder eliminar una columna. Així doncs, a l'inici de la cerca cal fer una cerca més, però durant la cerca llarga, se'n redueix una en cadascun dels múltiples passos.

Tot i que el disseny de la tercera estructura havia estat creat per disminuir el temps de cerca, un altre canvi va ser-hi introduït. Fins a aquell moment el sistema estava pensat per filtrar els usuaris pel seu nom, és a dir, un usuari tenia accés al sistema o no en tenia. Per aquest motiu cada usuari tenia la seva contrasenya per accedir al sistema.

Va ser en aquest moment però, quan es va creure que el sistema no podia funcionar d'aquesta manera. Cada producte havia de tenir la seva contrasenya i si un usuari volia accedir a la informació de cert producte, n'havia de conèixer la contrasenya. Per aquest motiu, un altre canvi que es va introduir a l'estructura de la base de dades va ser que es va eliminar la columna *Contrasenya* de la *Taula d'usuaris*, i va ser afegida a la *Taula de Productes*.

Portant a terme els passos descrits, l'estructura definitiva de la base de dades consta de 6 taules estructurades tal com es pot veure a la *Figura 4-19*.

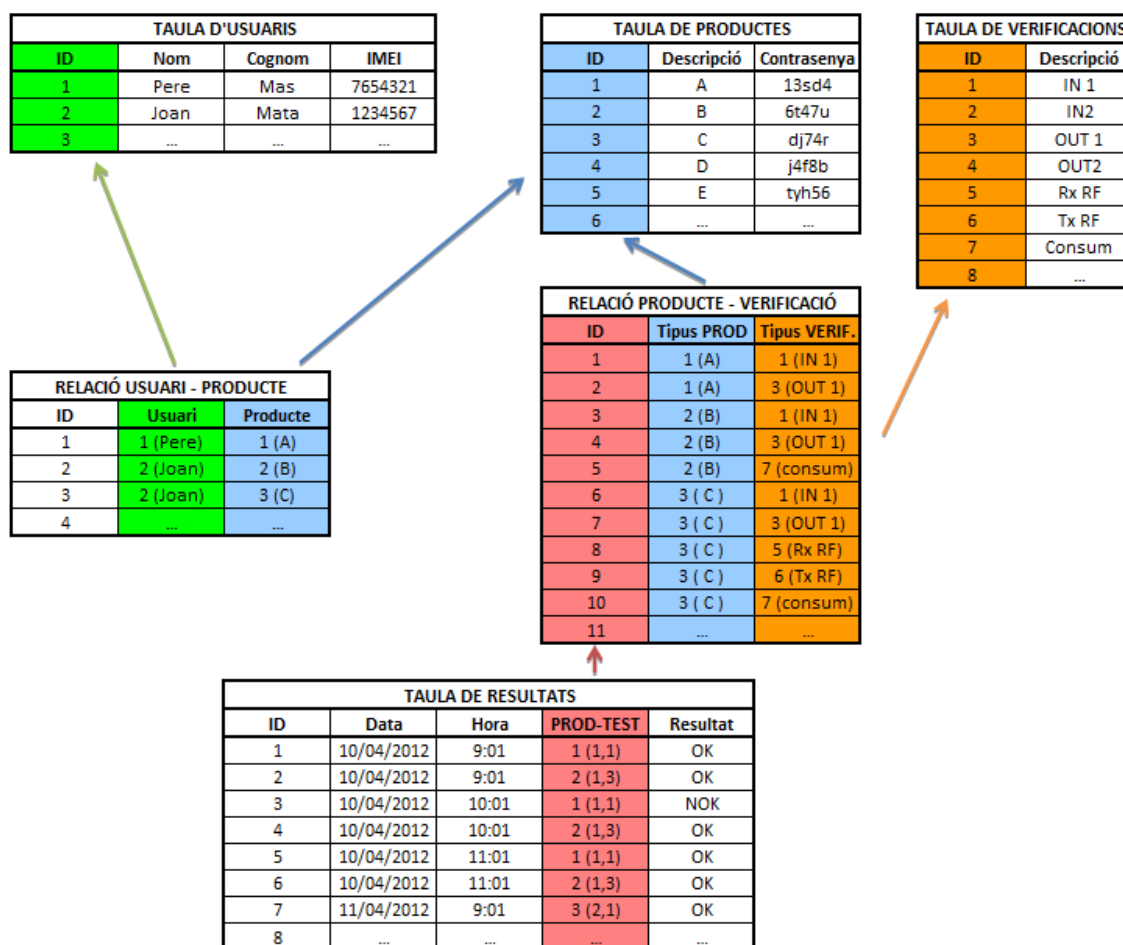


Figura 4-19 Proposta definitiva de l'estructura de la base de dades.

A continuació es descriuen totes les taules i cadascun dels camps que formen l'estructura final de la base de dades.

- **Taula d'usuaris**

Aquesta taula conté la informació de cadascun dels usuaris que té accés al sistema. De cada usuari en té la següent informació:

- **Identificador.** És un valor numèric que serveix per identificar cada usuari de la taula amb un valor únic.
- **Nom** de l'usuari.
- **Cognom** de l'usuari.
- **IMEI.** Paràmetre intern del dispositiu mòbil que quan l'usuari es dona d'alta l'aplicació també envia. Aquest paràmetre permet controlar els dispositius que accedeixen a la informació no permetent que un usuari accedeixi al sistema des de varis dispositius.

- **Taula de productes**

Taula que conté la llista de tots els productes que es troben al sistema. De cada producte en aquesta taula s'hi pot trobar:

- **Identificador.** Valor numèric que serveix per identificar cada producte de la taula amb un valor únic i irrepètible.
- **Descripció.** Camp de caràcters que conté el nom del producte.
- **Contrasenya.** Valor alfanumèric de 5 xifres diferent per cada producte que hi ha a la taula.

- **Taula de verificacions**

Taula que conté totes les possibles verificacions que poden tenir assignades els productes que hi ha entrats al sistema. De cada verificació entrada, a aquesta taula se'n guarda la informació següent:

- **Identificador.** Valor numèric que serveix per identificar cada tipus de verificació de la taula amb un valor únic i irrepètible.
- **Descripció.** Camp de caràcters que conté el nom de la verificació.

- **Taula de relació usuari – producte**

Aquesta taula conté la relació dels diferents productes als que cada usuari té accés. Si per exemple un usuari té accés a 4 productes, doncs a aquesta taula hi haurà 4 entrades d'aquest usuari, una per cada producte. Aquesta taula consta dels següents camps.

- **Identificador.** Valor numèric que serveix per identificar cada relació entre usuari i producte de la taula amb un valor únic i irrepètible.
- **Usuari.** Correspon al camp *Identificador* de la fila d'un usuari de la *Taula d'usuaris* i serveix per designar un usuari.
- **Producte.** Correspon al camp *Identificador* de la fila d'un producte de la *Taula de productes* i serveix per designar un producte.

- **Taula de relació producte – verificació**

Aquesta taula conté la relació de les diferents verificacions que un producte pot tenir. En aquest cas la taula consta dels següents camps.

- **Identificador.** Valor numèric que serveix per identificar cada relació entre un producte i una verificació de la taula amb un valor únic i irrepètible.

- **Producte.** Correspon al camp *Identificador* de la fila d'un producte de la *Taula de Productes* i serveix per designar un producte a la relació.
- **Verificació.** Correspon al camp *Identificador* de la *Taula de Verificacions* i serveix per designar una verificació a la relació.

- **Taula de Resultats**

Taula que conté els resultats de totes les verificacions de tots i cadascun dels productes que hi hagi donats d'alta al sistema. Per poder guardar aquesta informació de manera ordenada els camps de cada entrada són els següents.

- **Identificador.** Valor numèric que serveix per identificar cada entrada de resultat de la taula, amb un valor únic i irrepètible.
- **Data** en la que el resultat és rebut i processat pel sistema.
- **Hora** en la que el resultat és rebut i processat pel sistema.
- **Relació producte-test.** Correspon al camp *Identificador* d'una entrada de la *Taula de relació producte-verificació* i serveix per designar de quin producte i de quina verificació d'aquell producte és aquell resultat.
- **Resultat.** Aquest camp correspon al resultat de la verificació del producte del camp anterior.

4.2.2. Creació de la base de dades

Una vegada es disposa de l'estructura de la base de dades, la creació d'aquesta no és un aspecte massa complexa. El sistema de gestió de bases de dades relacionals *MySQL*, disposa d'una eina per administrar i editar bases de dades anomenada *phpMyAdmin*. Aquesta eina permet crear, modificar i esborrar les diferents taules d'una base de dades, així com els camps que les formen. És una eina molt simple de fer funcionar però que permet portar a terme una gran varietat d'accions, així com comprovacions de cerques i insercions.

Tan difícil com crear la base de dades és el fet de trobar on crear-la. En aquest sistema era imprescindible que es trobés en un servidor web que li oferís accés a

Internet, ja que si no era així el sistema no funcionaria. Per aconseguir-ho hi havia les següents dues opcions:

- Instal·lar tots els elements necessaris a un ordinador propi perquè pogués fer de servidor web i crear allà la base de dades.
- Buscar una pàgina web que oferís allotjament web i crear la base de dades allà.

Després de buscar si alguna pàgina web ho oferia i obtenir un resultat satisfactori, la segona opció va ser l'escollida. D'aquesta manera no calia invertir temps en documentar-se per fer la instal·lació i posta en funcionament d'un servidor web. Tampoc caldria doncs, tenir un ordinador fent de servidor web tot el dia engegat.

Un cop decidit l'emplaçament només va caldre crear la base de dades, les taules i els seus camps en el format correcte. Com ja s'ha comentat, tots aquests passos es van portar a terme mitjançant l'eina *phpMyAdmin*.

4.3. Aplicatiu servidor

L'aplicatiu servidor d'aquest sistema és l'element que dona vida a la base de dades i que s'encarrega de fer la connexió entre els diferents elements que formen el sistema. Per poder portar a terme totes aquestes tasques, aquest servidor s'ha de trobar (igual que la base de dades) en un servidor web, ja que és aquest el que li proporciona la connexió TCP/IP amb la qual es comunica tan amb l'aplicació mòbil com amb els diferents verificadors.

El llenguatge de programació utilitzat per desenvolupar aquest aplicatiu ha estat el *PHP*. Aquest, és un llenguatge de programació dissenyat originalment per la creació de pàgines web dinàmiques, però principalment utilitzat, entre altres tasques, per la interpretació de peticions al costat servidor. Té la capacitat de connexió amb la majoria de motors de bases de dades que més s'utilitzen actualment, entre ells el *MySQL* (utilitzat en aquest projecte). És de lliure ús per tothom i posseeix una àmplia documentació a la seva pàgina web, fet que s'agraeix molt quan els coneixements sobre aquest llenguatge són nuls i cal desenvolupar un aplicatiu de més de 1.000 línies de codi.

A continuació es poden veure les funcions bàsiques que es porten a terme a l'aplicatiu servidor agrupades segons si el servidor ha de fer una consulta a la base de dades o hi ha d'inserir algun tipus d'informació.

4.3.1. Inserció d'informació a la Base de Dades

A continuació es descriuen les 2 funcions que porten a terme insercions d'informació a la base de dades.

- **Alta d'un nou usuari.** Aquesta funció permet donar d'alta sobre un producte, un nou usuari i es crida des de l'aplicació mòbil. El que fa l'aplicatiu servidor és agafar les dades que des de l'aplicació mòbil se li envien i les introdueix a la base de dades seguint la següent seqüència.
 1. Processa tota la informació que rep i es guarda només els paràmetres que necessita, que són els que l'usuari ha entrat a l'aplicació (nom, cognom, producte i contrasenya).
 2. Comprova que la contrasenya del producte sigui correcte.
 - a. En cas de no ser correcte.
 - i. Retorna a l'aplicació un missatge informant que l'alta no s'ha pogut portar a terme correctament.
 - b. En cas de ser correcte.
 - i. Introdueix el nom i el cognom de l'usuari, i l'*IMEI*⁴ del dispositiu mòbil, a la *Taula d'usuaris*.
 - ii. Es guarda el valor *id* de la fila del nou usuari que acaba d'entrar.
 - iii. Busca el nom del producte que l'usuari ha entrat, a la *Taula de productes*, n'agafa el valor de la columna *id* i se'l guarda.
 - iv. Va a la *Taula Relació Usuari-Producte* i mitjançant una nova entrada, guarda l'*id* de l'usuari a la columna *Usuari*, i l'*id* del producte a la columna *Producte*.

⁴ L'*International Mobile Equipment Identity* (IMEI) és un codi preprogramat als telèfons mòbils GSM que els identifica inequívocament a nivell mundial.

- v. Si tot ha pogut ser portat a terme amb èxit, s'envia un missatge a l'aplicació informant-li-ho.
-
- **Afegir resultat d'una verificació.** Aquesta funció permet introduir el resultat d'una verificació portada a terme, a la base de dades. És cridada des del verificador de producció cada vegada que la verificació d'un equip ha finalitzat. Quan l'aplicatiu servidor rep aquesta petició porta a terme els següents passos.
 1. Recull la informació que rep per part del verificador i n'extreu els paràmetres que necessita. Aquests paràmetres consisteixen en la columna *id* de la *Taula Relació Producte-Test* i el resultat de la verificació, que és un "1" si ha estat correcte i un "0" si no ha estat així.
 2. El servidor comprova que la informació rebuda és correcte. Per fer-ho utilitza el procés definit a l'apartat 4.5.4..
 3. Per cada relació producte-test i resultat obtinguts per part del verificador, l'aplicatiu servidor agafa la data i l'hora en la que rep la informació i fa una entrada a la taula *Resultats* de la base de dades.
 4. Comprovació de la correcte inserció a la base de dades.
 - a. Si el procés d'inserció de tota la informació ha estat correcte, l'aplicatiu servidor envia al verificador un missatge conforme la inserció ha estat portada a terme amb èxit.
 - b. Si el procés ha sofert algun imprevist, algun error o el valor de comprovació del punt 2 no correspon amb la informació rebuda, s'envia un missatge al verificador indicant-li l'error detectat i la impossibilitat d'inserció de la informació a la base de dades.

4.3.2. Cerca d'informació de la Base de Dades

A continuació es descriuen les 6 funcions que comporten cerques a la base de dades.

- **Cerca de resultats per dates.** És la funció que permet obtenir els resultats de les verificacions d'un producte que s'han portat a terme entre les dues dates escollides per l'usuari. Es tracta d'una funció cridada per l'aplicació mòbil a l'aplicatiu servidor perquè aquest busqui a la base de dades el que demana.
 1. En rebre aquesta petició, com en tots els casos, el primer que fa l'aplicatiu servidor és descartar la informació que no necessita. En aquest cas, els paràmetres amb els que es queda són:
 - El nom de l'usuari
 - El cognom de l'usuari
 - El producte
 - La data d'inici a partir de la que ha de buscar entrades del producte a la base de dades
 - La data final de la cerca
 2. Amb aquesta informació el servidor comprova que l'usuari tingui accés al producte seleccionat comprovant-ho a la *Taula Usuari-Producte*.
 - a. En cas negatiu.
 - i. Envia un missatge a l'aplicació mòbil indicant que l'usuari no té accés al producte sol·licitat.
 - b. En cas afirmatiu.
 - i. Busca l'*id* del producte demanat a la *Taula Productes*.
 - ii. A la *Taula Relació Producte-Verificació* busca a totes les files on hi hagi l'*id* trobat al pas "i" i se'n guarda els *ids* d'aquesta taula. El que fa amb aquest pas és conèixer les verificacions que es porten a terme a l'equip sol·licitat, per així poder-les buscar a la *Taula Resultats*.

iii. A la *Taula Resultats*, a la columna de relació *producte-verificació*, hi busca tots els *ids* trobats al pas “ii”, i els que estan dins les dates entrades per l’usuari, se’n guarda el valor de la columna *Resultat*.

iv. Una vegada recorreguda tota la *Taula Resultats*, l’aplicatiu servidor contesta la petició informant de la quantitat total d’equips que van ser verificats entre les dates entrades. També informa de la quantitat d’equips amb resultat negatiu que hi van haver per cadascuna de les diferents verificacions del producte.

o **Exemple de resultat**

▪ Total equips verificats entre les dates entrades	50
▪ Equips amb error a la “Verificació 1”	3
▪ Equips amb error a la “Verificació 2”	0
▪ Equips amb error a la “Verificació 3”	1

- **Cerca de resultats per quantitat.** Funció que permet obtenir els resultats de les últimes verificacions que s’han portat a terme d’un producte concret. La quantitat de verificacions que es busquen a la base de dades és també un paràmetre entrat per l’usuari i els resultats que s’obtenen són els últims que han estat verificats. Per portar a terme aquest procés es segueixen els següents passos:

1. Com en el cas anterior, el primer que porta a terme el servidor en el moment en el que li arriba aquesta petició és descartar la informació que no necessita. En aquest cas, els paràmetres amb els que es queda són:

- El nom de l’usuari
- El cognom de l’usuari
- El producte
- Quantitat de verificacions que vol que es busquin a la base de dades

2. Amb aquesta informació el servidor comprova que l'usuari tingui accés al producte seleccionat comprovant-ho a la *Taula Usuari-Producte*.

a. En cas negatiu.

i. Envia un missatge a l'aplicació indicant que l'usuari no té accés al producte sol·licitat.

b. En cas afirmatiu.

i. Busca l'*id* del producte demanat a la *Taula Productes*.

ii. A la *Taula Relació Producte-Verificació* busca a totes les files on hi hagi l'*id* trobat al pas "i" i se'n guarda els *ids* d'aquesta taula.

iii. A la *Taula Resultats*, a la columna de relació producte- verificació, hi busca tots els *ids* trobats al pas "ii" i recull la informació de la columna *Resultat* només de les últimes verificacions entrades a la base de dades. La quantitat de resultats que recull és el paràmetre entrat per l'usuari.

És a dir, si a la *Taula Resultats* hi ha el resultat de 1.000 verificacions del producte "A" i l'usuari ha entrat que vol veure els resultats de les 100 últimes, l'aplicatiu servidor només comptabilitza els 100 últims resultats entrats i en retorna el resultat.

o **Exemple de resultat**

▪ Total equips buscats a la base de dades	100
▪ Equips amb error a la "Verificació 1"	2
▪ Equips amb error a la "Verificació 2"	0
▪ Equips amb error a la "Verificació 3"	1

- **Cerca dels productes que té assignats un usuari.** Aquesta funció permet conèixer els productes als que un usuari donat d'alta al sistema té accés. És cridada des de l'aplicació mòbil en el moment en el que un usuari vol visualitzar

resultats. Just després d'entrar el seu nom i cognom a l'aplicació hi apareix un desplegable al que ha de seleccionar un dels productes llistats. Per poder disposar d'aquesta informació l'aplicatiu servidor segueix els següents passos:

1. En el moment de rebre la petició, l'aplicatiu servidor només es queda amb el nom i el cognom de l'usuari.
2. Amb aquesta informació va a la *Taula Usuaris* i busca la fila en la que el nom i el cognom coincideixen amb els entrats i n'agafa l'*id* de la primera columna.
 - a. En el cas que el nom d'usuari entrat no estigui donat d'alta al sistema.
 - i. Envia un missatge a l'aplicació indicant que l'usuari entrat no està donat d'alta.
 - b. En el cas que el nom de l'usuari entrat sí que estigui donat d'alta.
 - i. Va a la *Taula Relació Usuari-Producte* i per cada fila que coincideixi l'identificador de la columna *Usuari* amb el trobat a la *Taula Usuaris*, agafa l'identificador de la columna *Producte*.
 - ii. Amb tota la llista d'identificadors de productes, va a la *Taula Productes* i canvia cada identificador de la llista extreta de la *Taula Relació Usuari-Producte* pel nom del producte.
 - iii. Una vegada disposa de la llista de productes als que té accés l'usuari, els envia a l'aplicació.
- **Cerca de les verificacions que s'apliquen a un producte.** Funció que permet conèixer les verificacions que s'apliquen a cadascun dels diferents productes entrats al sistema. Aquesta funció es crida des dels verificadors per contrastar la informació que posseeixen. D'aquesta manera, en cas d'haver-hi algun canvi en les verificacions d'algun producte, el verificador informaria de que no disposa de

la mateixa informació que hi ha a la base de dades i esperaria que s'arreglés el problema. Per portar a terme aquesta cerca l'aplicatiu servidor segueix els següents passos.

1. Com a tots els casos anteriors, l'aplicatiu servidor destrua la informació que li interessa i se la guarda. En aquest cas només es queda amb el nom del producte.
 2. Amb el nom del producte va a la *Taula Productes* i agafa l'id del producte.
 3. Mitjançant l'identificador del producte, a la *Taula Relació Producte-Verificació*, busca a totes les files on hi ha l'identificador del producte seleccionat i n'agafa els identificadors del tipus de test.
 4. La llista d'identificadors de tipus de verificacions l'envia als verificadors i són aquests els que contrasten si la informació rebuda és igual a la que disposen.
-
- **Cerca de l'identificador de la relació entre el producte i el test.** Aquesta funció permet conèixer el valor identificador que relaciona cada producte amb cadascuna de les seves verificacions. És una funció que es crida des dels diferents verificadors abans d'enviar els resultats cap a l'aplicatiu servidor, per d'aquesta manera poder cridar la funció *Afegir Resultats d'una Verificació* enviant els paràmetres correctes perquè aquests siguin guardats a la base de dades sense errors. Per aconseguir-ho, l'aplicatiu servidor segueix els següents passos.
 1. En aquest cas els paràmetres amb els que el servidor es queda són dos, el nom del producte i el nom de la verificació.
 2. El primer que gestiona és el nom del producte i el que fa és buscar-ne el seu identificador a la *Taula Productes*.
 3. Seguidament fa el mateix amb el nom de la verificació, però en aquest cas busca l'identificador del test a la *Taula Verificacions*.

4. En el moment en el que disposa dels dos identificadors, a la *Taula Relació Producte-Verificació*, busca la fila on hi apareguin els dos identificadors trobats anteriorment. A la fila on els localitza es guarda el paràmetre *id* que relaciona el producte amb la verificació i l'envia cap al verificador.

- **Cerca de tots els productes que estan donats d'alta al sistema.** Aquesta funció retorna la llista de productes que estan al sistema.

És cridada des de l'aplicació mòbil en el moment en el que un usuari es vol donar d'alta d'un nou producte. La llista rebuda per part de l'aplicatiu servidor, l'aplicació mòbil la col·loca a un desplegable perquè l'usuari pugui seleccionar el producte que més li interessi. Per aconseguir la llista, l'aplicatiu servidor porta a terme els passos descrits a continuació.

1. En el moment de rebre la petició, l'aplicatiu servidor va a la *Taula de Productes* i per cada fila n'agafa el valor que hi ha a la columna *Descripció* (que és el nom del producte).
2. Tan bon punt ha recorregut tota la taula i disposa de la llista l'envia cap a l'aplicació mòbil.

4.4. Verificador

Els verificadors d'aquest sistema són els dispositius que porten a terme les comprovacions elèctriques a les diferents plaques electròniques que es produeixen a les empreses subcontractades. Com ja s'ha comentat a l'apartat 3.4 aquests aparells estan formats per una part hardware i per una part software i mitjançant les dues es porten a terme les verificacions.

Per poder portar a terme aquest projecte la única part dels verificadors que s'ha vist afectada ha estat la part de software ja que ha calgut afegir-hi varis passos. A continuació es descriu el nou procés de verificació dels verificadors i el desenvolupament d'aquests.

4.4.1. Nou procés de verificació

A continuació es pot veure una comparativa entre l'estructura general del procés de verificador actual i la del proposat.

Procés actual	Procés proposat
1. Posar en marxa el verificador	1. Posar en marxa el verificador
2. Iniciar verificació	2. Cerca de les verificacions que s'apliquen al producte
3. Realitzar comprovacions	3. Iniciar verificació
4. Mostrar resultat a l'operari	4. Realitzar comprovacions
	5. Mostrar resultat a l'operari
	6. Cercar identificador de relació Producte-Verificació
	7. Enviar resultat al servidor

Com es pot comprovar a simple vista la quantitat de passos a seguir en un procés i en l'altre varia força. Cal dir però, que el fet d'aconseguir que el temps introduït per portar a terme aquests nous passos sigui mínim, fa que el cost dels equips no variï.

A continuació es descriuen els diferents passos que porta a terme el procés proposat, posant més atenció als nous passos introduïts.

- 1. Posar en marxa el verificador.** Consisteix en engegar-lo i esperar a que tot el hardware i software es posi en funcionament.
- 2. Cerca de les verificacions que s'apliquen al producte.** El programa del verificador sap quin és el producte que ell verifica i també sap quines són les verificacions o comprovacions que li ha d'aplicar. Per estar segur que el que farà serà correcte, abans de començar la verificació de tot el lot d'equips, el verificador demana a l'aplicatiu servidor la llista de verificacions que al sistema està entrat que cal aplicar a aquell producte. D'aquesta manera el verificador pot comparar la llista de la que ell disposa amb la que li arriba de l'aplicatiu servidor, fet que permet conèixer si hi ha hagut canvis o no.

En el cas que les dues llistes no coincideixin, el verificador n'informa a l'operari mostrant un error conforme la informació que ell té no és la mateixa que hi ha al sistema. En el cas que les dues llistes coincideixin, es continua el procés de verificació al següent punt.

Lògicament, aquest pas és un dels que s'introdueix en el cas d'aplicar el procés proposat.

3. **Iniciar verificació.** Una vegada el verificador ha portat a terme les comprovacions de les verificacions a realitzar i l'operari ha situat un equip al seu emplaçament, l'operari prem el botó d'inici de les comprovacions.
4. **Realització de les comprovacions.** Tan bon punt s'engeguen les comprovacions, el verificador comença a realitzar les diferents verificacions que el producte té assignades a la placa que en aquell moment està situada a l'emplaçament. L'operari mentrestant no ha de fer res, només esperar una resposta per part del verificador.
5. **Mostrar resultat a l'operari.** En el moment en el que el verificador ha acabat de fer les comprovacions, mostra a l'usuari el resultat d'aquestes, podent ser "Correcte" si s'han superat totes satisfactòriament, o "Incorrecte" si alguna no ha estat superada. En aquest segon cas el verificador mostra a l'usuari quin ha estat la verificació que no ha estat superada amb èxit.
6. **Cercar identificació de relació Producte-Test.** Un cop el verificador ha acabat de fer totes les verificacions a l'equip i ha mostrat el resultat a l'operari, el verificador fa la consulta de quin és l'identificador que relaciona cada test aplicat amb el producte que està verificant.

Aquest pas s'aprofita per fer en aquest moment perquè és quan el verificador no està fent res (l'operari treu l'equip verificat i en posa un de nou) i d'aquesta manera no es malgasta temps de verificació i el preu de cost del producte no augmenta.
7. **Enviar resultats a l'aplicatiu servidor.** Igual que el pas 6, s'aprofita el moment en que el verificador està en espera per enviar la informació del resultat de la verificació cap a l'aplicatiu servidor. En aquest cas el que es fa és cridar la funció *Afegir Resultat d'una verificació* de l'aplicatiu servidor enviant-li els resultats de les verificacions. Tot seguit, si la inserció a la base de dades s'ha portat a terme amb èxit l'aplicatiu servidor n'informa, i sinó, informa que el procés no ha pogut ser portat a terme correctament.

En el moment en el que s'han portat a terme els 7 passos, de tot el lot de productes a verificar només se n'ha verificat un. El que tocaria fer després del pas 7

(suposant que durant el pas 6 i el 7 l'operari ha canviat l'equip a verificar), seria tornar a començar al pas 3 i portar-los tots a terme fins al 7. D'aquesta manera, caldria anar repetint aquest procés per cada equip fins a tenir tots els que conformen el lot verificats.

4.4.2. Desenvolupament dels nous programes pels verificadors

Portar a terme aquesta part del projecte no ha estat tan complexa com la resta ja que ha estat la única part que no ha calgut estudiar des del primer moment com es podia resoldre cadascun dels petits objectius o reptes que anaven apareixent. Aquest fet ha estat degut a que l'autor coneixia la tecnologia i els diferents llenguatges de programació amb els que es creen els verificadors ja que n'és el dissenyador.

De totes maneres, no ha estat del tot fàcil ja que les comunicacions a través d'Internet no s'havien portat mai a terme i amb llenguatges visuals com el *LabVIEW* de *National Instruments* fan que aquest tipus de comunicació no sigui tan immediata com pot ser en altres llenguatges.

Un altre entorn amb el que es creen la gran majoria de sistemes de verificació és el *Microsoft Visual Studio*, que amb llenguatges com el *C++*, el *Visual Basic* o el *C Sharp (C#)* ofereixen una facilitat important a l'hora de permetre comunicacions mitjançant Internet.

4.5. Protocol de comunicacions

Totes les comunicacions entre els diferents elements que formen el sistema creat en aquest projecte es porten a terme per mitjà d'Internet. Internet es basa en un conjunt de protocols de xarxa que permeten la transmissió de dades entre ordinadors o dispositius de característiques molt semblants a aquests. Sovint, al conjunt de protocols se'ls anomena *protocols TCP/IP* en referència als dos protocols més importants que el formen, el *Protocol de Control de Transmissió (TCP)* i el *Protocol d'Internet (IP)*. De protocols però n'existeixen molts, entre ells el popular *HTTP (Protocol de Transferència d'Hipertext)* que s'utilitza per accedir a pàgines web. Aquest ha estat el protocol escollit per portar a terme les comunicacions d'aquest projecte.

A continuació s'enumeren els diferents tipus de comunicació que es porten a terme al sistema, els quals es descriuen als següents punts.

- A l'apartat 4.5.1. es descriu el tipus de comunicació definida per enviar peticions a l'aplicatiu servidor. Com s'ha comentat en apartats anteriors, aquestes comunicacions les porten a terme l'aplicació mòbil i els verificadors.
- A l'apartat 4.5.2. es descriu la resposta que porta a terme l'aplicatiu servidor a les diferents peticions descrites a l'apartat 4.5.1..

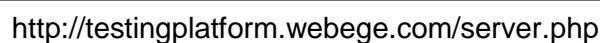
- A l'apartat 4.5.3. es descriu el tipus de comunicació que s'utilitza entre l'aplicatiu servidor i la base de dades.

Aquesta comunicació es porta a terme quan l'aplicatiu servidor rep una petició de les de l'apartat 4.5.1.. Llavors fa la consulta a la base de dades mitjançant les comunicacions de l'apartat 4.5.3. i contesta a l'aplicació mòbil o als verificadors mitjançant el tipus de respostes descrites a l'apartat 4.5.2..

- Finalment, a l'apartat 4.5.4. es descriu el protocol de seguretat creat per protegir el sistema.

4.5.1. Enviament de peticions cap a l'aplicatiu servidor

L'aplicatiu servidor creat per aquest projecte es troba situat a un servidor web, el qual li proporciona la connexió a Internet. Es pot dir doncs, que es troba en el mateix emplaçament que una pàgina web, per tant, la comunicació cap a ell serà la mateixa que s'utilitza per accedir a una pàgina web, que és mitjançant la *URL (Uniform Resource Locator)*.



```
http://testingplatform.webege.com/server.php
```

Figura 4-20 Url del servidor web.

La *URL* de la *Figura 4-20* és la que porta fins al servidor web del sistema. Les parts que la formen es descriuen a continuació:

- “**http://**”. És la part que defineix quin és el protocol utilitzat en la comunicació.
- “**testingplatform.webege.com**”. És la part que conforma l'adreça del servidor web on es troba el fitxer aplicatiu servidor.

Quan es va decidir que l'aplicatiu servidor del sistema es trobaria en un servidor web i que aquest no seria propi, va caldre triar un domini. La pàgina web que es va seleccionar per guardar-hi el fitxer *PHP* que conté el codi font de l'aplicatiu servidor ofereix la possibilitat de triar un tros del domini. En aquest cas “testingplatform” va ser la part escollida, mentre que “webege.com” la va assignar la pròpia pàgina web.

- “**/server.php**”. És el nom del fitxer que conté el codi font de l'aplicatiu servidor que s'ha creat per poder portar a terme aquest projecte. Com s'ha comentat en

apartats anteriors aquest s'ha creat amb llenguatge *PHP*, per això de l'extensió que es pot veure.

En el cas de ser una pàgina web, el fitxer que la contingues estaria localitzat en aquest mateix emplaçament però tindria un altre nom, *Index* o *Default* per exemple. Aquest fet feria que s'agafés automàticament i se'n visualitzés la interfície d'usuari definida internament, mitjançant el navegador.

En el cas de l'aplicatiu servidor d'aquest sistema, com que no es necessita interfície d'usuari, no se n'ha creat, i el nom del fitxer s'ha preferit que no fos un dels que s'agafa automàticament.

Tan bon punt s'ha arribat fins a l'aplicatiu servidor, el que cal és informar-lo de quina és la funció que se li pretén cridar.



```
http://testingplatform.webege.com/server.php?command=NomFunció
```

Figura 4-21 Com cridar una funció.

Com es pot veure a la *Figura 4-21*, la manera que s'ha creat per cridar una funció és mitjançant el text “*?command=*” seguit del nom de la funció que l'aplicatiu servidor té definida al seu codi font.

Seguidament, després del nom de la funció cal passar-li també els paràmetres necessaris perquè el resultat que s'obtingui sigui el desitjat. A continuació s'especifica per cada funció, com es crida i se li passen els paràmetres.

Només hi ha dos elements del sistema que envien peticions cap a l'aplicatiu servidor, que són l'aplicació mòbil i els verificadors. La següent llista de funcions són les que aquests dos elements poden enviar i coincideixen amb les que als apartats 4.3.1 i 4.3.2 es descriuen. Allà però es defineix com les tracta l'aplicatiu servidor, aquí en canvi, es descriu com cal fer-les-hi arribar.

4.5.1.1. Afegir alta d'un nou usuari

Aquesta és la funció que es crida des de l'aplicació mòbil quan un usuari desitja donar-se d'alta al sistema.

- **Nom de la funció:** DonarAlta
- **Paràmetre 1.** Nom de l'usuari.
- **Paràmetre 2.** Cognom de l'usuari.

- **Paràmetre 3.** *IMEI* del dispositiu des del que pretén donar-se d'alta.
- **Paràmetre 4.** Producte del que es vol donar d'alta.
- **Paràmetre 5.** Contrasenya.

```
http://testingplatform.webege.com/server.php?command=DonarAlta.josep.codina.IMEI.producte.contrasenya
```

Figura 4-22 Exemple de *URL* de la funció *DonarAlta*.

4.5.1.2. Afegir resultat d'una nova verificació

Aquesta funció és la que es crida des dels diferents servidors quan aquests han realitzat la verificació d'un equip. En aquest cas la quantitat de paràmetres a enviar és variable ja que depenent de les verificacions que es porten a terme a l'equip en qüestió, s'envien més paràmetres o menys.

- **Nom de la funció:** AfegirTest
- **Paràmetre 1.** Identificador de la *Relació Producte-Verificació*.
- **Paràmetre 2.** Resultat que l'identificador del paràmetre anterior indica. En aquest cas és un "1" quan el resultat ha estat correcte, i un "0" quan aquest ha estat incorrecte.

Els paràmetres 1 i 2 es repeteixen conjuntament tantes vegades com verificacions tingui assignades l'equip del que s'envien els resultats.

- **Penúltim paràmetre.** El penúltim paràmetre sempre és "end" i serveix per indicar que a partir d'aquí no hi ha més grups de paràmetres 1 i 2.
- **Últim paràmetre.** És un paràmetre que permet fer l'entrada d'informació més segura.

```
http://testingplatform.webege.com/server.php?command=AfegirTest.3.1.4.1.5.1.end.27
```

Figura 4-23 Exemple de *URL* de la funció *AfegirTest*.

4.5.1.3. Cerca dels productes que té assignats un usuari

Aquesta funció es crida des de l'aplicació mòbil en el moment en el que un usuari donat d'alta al sistema s'identifica (entra el seu nom i cognom) amb la intenció de

conèixer els resultats de les verificacions d'un producte. Els paràmetres que cal enviar-li són els següents.

- **Nom de la funció:** ProductesUsuari
- **Paràmetre 1.** Nom de l'usuari.
- **Paràmetre 2.** Cognom de l'usuari.

```
http://testingplatform.webege.com/server.php?command=ProductesUsuari.josep.codina
```

Figura 4-24 Exemple de URL de la funció *ProductesUsuari*.

4.5.1.4. Cerca de resultats per dates

Aquesta funció es crida des de l'aplicació mòbil en el moment en el que un usuari donat d'alta desitja conèixer els resultats de les verificacions d'un producte en concret. En aquest cas, la consulta es porta a terme sobre les verificacions que han estat entrades al sistema entre dues dates escollides per l'usuari.

- **Nom de la funció:** RecollirResultats
- **Paràmetre 1.** Nom de l'usuari.
- **Paràmetre 2.** Cognom de l'usuari.
- **Paràmetre 3.** Indicador que permet definir si la cerca s'ha de fer "PerDates" o "PerQuantitat". En aquest cas és "PerDates".
- **Paràmetre 4.** Data d'inici sobre la que s'ha de buscar els resultats del producte del paràmetre 6.
- **Paràmetre 5.** Data final sobre la que s'ha de buscar els resultats del producte del paràmetre 6.
- **Paràmetre 6.** Producte del que cal fer la cerca de resultats.

```
http://testingplatform.webege.com/server.php?command=RecollirResultats.josep.codina.PerDates.25/04/2012.27/04/2012.QuadreManiobra
```

Figura 4-25 Exemple de URL de la funció *RecollirResultats* per dates.

4.5.1.5. Cerca de resultats per quantitat

Aquesta funció es crida des de l'aplicació mòbil en el moment en el que un usuari desitja conèixer els resultats de les verificacions d'un producte en concret. En aquest cas la consulta es porta a terme sobre la quantitat de verificacions escollida per l'usuari.

- **Nom de la funció:** RecollirResultats
- **Paràmetre 1.** Nom de l'usuari.
- **Paràmetre 2.** Cognom de l'usuari.
- **Paràmetre 3.** Indicador que permet definir si la cerca s'ha de fer "PerDates" o "PerQuantitat". En aquest cas és "PerQuantitat".
- **Paràmetre 4.** Quantitat d'equips del tipus que indica el paràmetre 5, sobre els que se'n vol conèixer el resultat de la verificació.
- **Paràmetre 5.** Producte del que cal fer la cerca de resultats.

```
http://testingplatform.webege.com/server.php?command=RecollirResultats.josep.codina.PerQuantitat.100.Emissor
```

Figura 4-26 Exemple de URL de la funció *RecollirResultats* per quantitat.

4.5.1.6. Cerca de les verificacions que s'apliquen a un producte

Aquesta funció es crida des dels diferents verificadors en el moment en el que s'han de posar a verificar un lot d'equips per així assegurar que la informació que tenen guardada coincideix amb la que hi ha a la base de dades.

- **Nom de la funció:** TestosProducte
- **Paràmetre 1.** Producte sobre el que es volen conèixer les verificacions que se li apliquen.

```
http://testingplatform.webege.com/server.php?command=TestosProducte.Emissor
```

Figura 4-27 Exemple de URL de la funció *TestosProducte*.

4.5.1.7. Cerca de l'identificador de la relació entre el producte i el test

Aquesta funció es crida des dels diferents verificadors en el moment en el que han verificat un equip i necessiten enviar al servidor els resultats obtinguts.

- **Nom de la funció:** RelacioTestProducte
- **Paràmetre 1.** Nom del producte que s'està verificant.
- **Paràmetre 2.** Identificador del test que s'ha aplicat al producte i que se'n vol conèixer la relació.

```
http://testingplatform.webege.com/server.php?command=RelacioTestProducte.Emissor.5
```

Figura 4-28 Exemple de URL de la funció *RelacioTestProducte*.

4.5.1.8. Cerca de la llista de productes donats d'alta al sistema

Aquesta funció es crida des de l'aplicació en el moment en el que un usuari es vol donar d'alta. Aquesta funció no necessita passar-li paràmetres, senzillament cridant-la n'hi ha suficient.

- **Nom de la funció:** ProductesSistema

```
http://testingplatform.webege.com/server.php?command=ProductesSistema
```

Figura 4-29 Exemple de URL de la funció *ProductesSistema*.

4.5.2. Recepció d'informació procedent de l'aplicatiu servidor

En el moment en el que un usuari accedeix a una pàgina web mitjançant un navegador, aquesta el que fa és mostrar informació, podent ser en format text, imatges, taules, etc. L'aplicatiu servidor, per la seva part, utilitza el mateix mecanisme per enviar informació. En el moment en el que s'accedeix a ell, aquest mostra un text o un altre, depenent de la funció que se li crida, els paràmetres que se li passen i les dades obtingudes de la cerca a la base de dades.

Cadascun dels elements receptors de la informació que ofereix l'aplicatiu servidor, té el seu propi mecanisme per obtenir-la. Una vegada obtinguda però, cal que

cadascun dels elements sigui capaç de destriar-ne només la interessant per poder-la mostrar o tractar-la depenent de les necessitats de cada cas.

La recepció o lectura de la resposta es porta a terme des de l'aplicació mòbil o des dels diferents verificadors. D'aquesta manera, llegint la resposta de l'aplicatiu servidor a cada petició que se li fa, es porta a terme la comunicació.

- **Recepció per part de l'aplicació mòbil**

A continuació s'expliquen els passos que es porten a terme per rebre i llegir la informació enviada des de l'aplicatiu servidor cap a l'aplicatiu mòbil.

1. Per llegir la informació contestada per part de l'aplicatiu servidor, s'utilitza el mètode *OpenStream()* de la classe *Url*. Aquest mètode permet obrir un canal de comunicacions entre l'aplicació mòbil i la resposta de l'aplicatiu servidor.
2. Una vegada obert el canal de comunicacions, mitjançant les classes *InputStreamReader* i *BufferedReader* es llegeix tot el contingut contestat per l'aplicatiu servidor i es guarda en format "cadena de caràcters".
3. En el moment en el que es disposa de la informació, el que es fa és tractar-la, quedant-se només amb la que és interessant i necessària. Per fer-ho, com que el format en el que l'aplicatiu servidor l'ha enviat és conegut, mitjançant mètodes varis de la classe *string* (cadena de caràcters) es guarda només la informació desitjada.

- **Recepció per part dels verificadors**

Com s'explica a l'apartat 4.4.2., no tots els verificadors estan programats en el mateix llenguatge de programació, fet que fa que per cada llenguatge el procés de lectura de les respostes sigui diferent. En aquest cas s'explica com es porta a terme aquesta lectura en un verificador programat en *Visual C Sharp (C#)*.

1. Mitjançant la classe *HttpWebRequest* es crea la sol·licitud d'informació de la pàgina web que es desitja. En aquest cas, depenent de quina sigui la petició a fer a l'aplicatiu servidor, se li passa la *Url* de l'apartat 4.5.1. que més convingui.
2. Seguidament, amb el mètode *GetResponse()* de la classe *HttpWebResponse* es llegeix tota la informació enviada per l'aplicatiu servidor.
3. Una vegada es disposa de tota la informació, aquesta es converteix a "cadena de caràcters". D'aquesta manera, el tractament d'aquesta informació en passos posteriors és molt més senzill.

4. Finalment, en el format "cadena de caràcters" es tracta la informació rebuda per poder ser gestionada correctament.

Com s'ha pogut comprovar en la descripció de la recollida d'informació per part dels dos elements, tot i fer-se de manera diferent, l'essència és la mateixa. En els dos casos cal obrir un canal de comunicacions, recollir la informació, guardar-la, posar-la en el format que més convingui i tractar-la per poder disposar dels paràmetres desitjats.

4.5.3. Comunicació entre l'aplicatiu servidor i la base de dades

Totes les comunicacions que es porten a terme entre l'aplicatiu servidor i la base de dades es porten a terme mitjançant consultes *MySQL* (*MySQL queries*). En el cas d'aquest sistema, es porten a terme dos tipus de consultes, una per introduir dades a la base de dades i l'altra per recollir-ne la informació. Al punt 3 se'n descriuen les diferències.

A continuació s'explica com es porten a terme aquestes consultes, les quals es troben a la part aplicatiu servidor. En aquest cas la base de dades no ha de fer res, és el motor de consultes de *MySQL* el que s'encarrega d'introduir les dades o de recollir-les, depenent de la petició feta per l'aplicatiu servidor.

1. **Establiment de connexió.** El primer que cal fer abans de fer la consulta consisteix en obrir una connexió entre l'aplicatiu servidor i el servidor en el que es troba la base de dades.

```
/** Establiment de connexió amb la Base de Dades *****  
$conn=mysql_connect($server,$user,$passwd) or die (mysql_error());
```

Figura 4-30 Exemple d'establiment de connexió.

- Com es pot comprovar a la figura anterior, per portar a terme la connexió es fa mitjançant el mètode *mysql_connect*.
- Els paràmetres que cal passar a aquest mètode són 3.

- L'adreça del servidor web on hi ha la base de dades. En aquest cas s'ha creat la base de dades al mateix servidor on es guarda l'aplicatiu servidor.
 - El nom d'usuari que fa la connexió. En aquest cas només hi ha un usuari que és el creador d'aquest projecte.
 - Contrasenya de l'usuari. Per verificar que l'usuari que pretén connectar-se a la base de dades és el verdader, és imprescindible passar-li la contrasenya.
- Finalment, perquè si durant la connexió hi ha algun error, aquest pugui ser mostrat, cal afegir els caràcters “*or die (mysql_error())*”.
2. **Selecció de la base de dades.** Una vegada la connexió ha estat feta amb èxit, cal seleccionar quina és la base de dades sobre la que es vol fer la consulta. En aquest cas, com que al sistema només hi ha una base de dades, sempre es selecciona la mateixa.

```
/** Selecció de la base de dades sobre la que es vol fer la consulta */  
mysql_select_db($bd,$conn) or die (mysql_error());
```

Figura 4-31 Exemple de selecció de base de dades.

- La selecció de la base de dades es porta a terme mitjançant el mètode *mysql_select_db*.
- Els paràmetres que cal passar a aquest mètode són 2.
 - El nom de la base de dades.
 - Nom de la connexió que s'utilitzarà per fer el traspàs d'informació. En aquest cas la connexió és la creada al pas “1”.
- Perquè en cas d'error durant la selecció de la base de dades, aquest sigui mostrat, s'hi afegeixen els caràcters “*or die (mysql_error())*”.

3. **Consulta a la base de dades.** En el moment en el que la connexió ha estat feta de manera correcta i la selecció de la base de dades també, cal fer la consulta. De consultes utilitzades en aquest sistema n'hi ha dues, una per recollir dades de la base de dades i una altra per introduir-n'hi. Les dues però es porten a terme mitjançant el mètode *mysql_query*.
- Introducció de dades a la base de dades.

```
/** Afegim una fila (nou usuari) a la taula 'info_usuaris' **/  
mysql_query("INSERT INTO info_usuaris (ID, nom, cognom, imei, producte, password)  
VALUES (' ', '$Info[1]', '$Info[2]', '$Info[3]', '$Info[4]', '$Info[5]') ") or die (mysql_error());
```

Figura 4-32 Exemple d'introducció de dades a la base de dades.

- Per poder portar a terme una inserció de dades amb el mètode *mysql_query* cal tenir en compte certs aspectes.
 - Cal indicar mitjançant la paraula “*INSERT*” que el que es pretén portar a terme és una inserció.
 - Cal indicar el nom de la taula a la que es vol fer la inserció després de la paraula “*INTO*”. En aquest cas *info_usuaris* és el nom de la taula a la que es pretén inserir una fila.
 - Dins de parèntesis i separats per comes cal afegir els noms dels camps que formen la taula i que es pretenen omplir.
 - Cal indicar amb la paraula “*VALUES*” que a continuació s'entren els valors a inserir als diferents camps de la taula.
 - També dins de parèntesis i separats per comes cal introduir els valors que es volen inserir. En aquest cas, els valors a inserir són els paràmetres passats al moment de cridar la funció *DonarAlta* de l'apartat 4.5.1.1..
 - Finalment, per si hi ha algun error durant la inserció, s'afegeixen els caràcters “*or die (mysql_error())*” perquè així l'error sigui mostrat.

- Consulta de dades de la base de dades

```
/** Busquem si un usuari està donat d'alta i quin és el seu ID ****  
$Consulta=mysql_query("SELECT * FROM info_usuaris", $conn) or die (mysql_error());
```

Figura 4-33 Exemple de cerca de dades a la base de dades.

- Per poder portar a terme una cerca de dades amb el mètode *mysql_query* cal tenir en compte els següents aspectes.
 - Cal indicar mitjançant la paraula “*SELECT*” que el que es pretén portar a terme és una cerca.
 - Seguidament cal indicar sobre quina columna es vol portar a terme la cerca. En aquest cas la cerca es vol fer sobre totes les columnes de la taula, per tant, cal posar-hi un “***”.
 - A continuació cal indicar el nom de la taula a la que es vol fer la cerca després de la paraula “*FROM*”. En aquest cas *info_usuaris* és el nom de la taula a la que es pretén buscar un usuari.
 - Després d'una coma cal indicar quina és la connexió que s'utilitza per portar a terme la cerca. En aquest cas és la primera que s'ha fet.
 - Finalment, per si hi ha algun error durant la cerca, s'afegeixen els caràcters “*or die (mysql_error())*” perquè així l'error pugui ser mostrat.
4. **Finalització de la connexió.** En el moment en el que la cerca o la inserció han estat finalitzades amb èxit, el que cal fer és finalitzar la connexió oberta al pas “1” tancant-la.

```
/** Finalitzem connexió **  
mysql_close($conn);
```

Figura 4-34 Exemple de finalització de la connexió.

- Per poder finalitzar la connexió cal utilitzar el mètode *mysql_close*, sobre el que només cal tenir en compte un paràmetre.
 - Cal indicar el nom de la connexió que es vol finalitzar. En aquest cas és la que s'havia establert per fer la cerca o inserció.

4.5.4. Protocol de seguretat contra agents externs

Per evitar que l'aplicatiu servidor processés peticions d'entrada d'informació no generades pels verificadors, sinó per programes externs al sistema, s'ha creat un mecanisme que minimitza les possibilitats de vulnerar l'entrada d'informació.

Aquest mecanisme consisteix en afegir un paràmetre al final de la funció *AfegirTest* de l'apartat 4.5.1.2., el qual se n'obtingui el valor fent diverses operacions matemàtiques conegudes als paràmetres que realment es volen entrar.

Un exemple d'aquest mecanisme seria el següent.

<http://testingplatform.webege.com/server.php?command=AfegirTest.3.1.4.1.5.1.end.27>

Figura 4-35 Exemple on es pot comprovar el protocol descrit.

- Els paràmetres en format número que es passen a la funció *AfegirTest* cal:
 - Sumar-los
 - Multiplicar la suma per 2.
 - Restar 3 a la suma.
- Una vegada portats a terme aquests càlculs, cal afegir el valor obtingut després del paràmetre "end".
- En el moment en el que l'aplicatiu servidor rep una petició d'*AfegirTest*, el primer que fa és agafar tots els paràmetres en format número que hi ha

abans del paràmetre “*end*” i els aplica els mateixos càlculs que anteriorment.

- Els suma.
 - Multiplicar la suma per 2.
 - Restar 3 a la suma.
-
- Si el resultat d'aquesta operació coincideix amb el valor del paràmetre que hi ha després de “*end*”, fa la inserció a la base de dades. Per altra banda, si no coincideix no fa la inserció.

D'aquesta manera, mitjançant aquest mecanisme, la intrusió per part d'elements externs al sistema es serà molt més complexa, quedant així el sistema molt més protegit.

5. Resultats i conclusions

En aquest apartat es descriuen els resultats obtinguts del sistema creat després d'aplicar-li una prova pilot. Seguidament s'enumeren una sèrie de modificacions o ampliacions que es podrien portar a terme al sistema perquè aquest pogués aportar encara més valor a l'empresa, i finalment es descriuen les conclusions del projecte.

5.1. Resultats

Per poder obtenir uns resultats realistes i poder avaluar si el sistema creat era vàlid per complir els objectius marcats, aquest va ser posat a prova mitjançant una prova pilot.

En el moment en el que el sistema va estar totalment desenvolupat, es van portar a terme les 3 accions necessàries perquè la prova pilot es pogués realitzar. Aquestes 3 accions són les que es descriuen a continuació.

1. Fer un nou programa específic per aquell verificador perquè portés a terme les noves tasques (explicades a l'apartat 4.4.1.).
2. Donar al verificador connexió a Internet.
3. Posar el nou programa al verificador.

5.1.1. Objectius de la prova pilot

- Comprovar el bon funcionament del sistema en un entorn real.
- Detectar possibles errors de funcionament.
- Comprovar que amb el sistema implantat, el temps de verificació dels productes no augmenta.

5.1.2. Fitxa tècnica de la prova pilot

A continuació es detalla la fitxa tècnica de la prova pilot.

FITXA TÈCNICA	
Data	14/06/2012
Empresa subcontractada	Nacional
Tipus de producte	Receptor
Número de lot a verificar	4687
Quantitat d'equips que formen el lot	1000
Durada estimada de la verificació de cada equip	20s
Durada estimada de la verificació de tot el lot	5.5h
Durada real de la verificació del lot	6h

Figura 5-1 Fitxa tècnica de la prova pilot.

5.1.3. Observacions

Cal dir que degut a la facilitat a l'hora de portar-la a terme, la prova pilot va ser realitzada a l'empresa subcontractada que es troba més aprop físicament de JCM Technologies. Aquest fet oferia la possibilitat que en cas d'error o problema, el desplaçament era curt i les pèrdues mínimes.

Un altre factor important a l'hora d'avaluar el resultat de la prova pilot és el cost que ha tingut la seva implementació. Tal com s'ha descrit a l'inici d'aquest apartat, ha calgut portar a terme 3 accions, en les que ha calgut dedicar unes hores de feina.

- Les hores de desenvolupament del nou programa pel verificador van ser 4. Si aquestes hores es comptabilitzen a 30€/hora, el cost econòmic de portar a terme el nou programa és de 120€.
- A donar accés a Internet al verificador, gravar-li el nou programa, fer les últimes comprovacions al sistema i observar l'inici de la verificació, s'hi va

tardar 1 hora. Si també es comptabilitza a 30€/hora, posar en marxa el verificador costa 30€.

Així doncs, el cost de posar en marxa el verificador per poder portar a terme la prova pilot va ser de 150€.

5.1.4. Resultats obtinguts a la prova pilot

Els resultats obtinguts a la prova pilot van ser realment positius ja que el nombre d'incidències generades pel sistema van ser mínimes. L'única incidència que hi va haver va ser que durant les primeres verificacions la connexió a Internet que s'acabava de posar al verificador, no funcionava bé. Aquest fet va fer que durant mitja hora el sistema no funcionés correctament. Al cap de mitja hora però es va detectar el problema, es va solventar i es va poder seguir amb la verificació. A partir de llavors tot va funcionar de manera correcta.

A la fitxa tècnica de la prova pilot es pot veure com hi ha una desviació de mitja hora entre la durada estimada per verificar el lot de productes, i la que realment es va tardar. Aquest fet, tal com s'explica al paràgraf anterior, no va ser degut a un error del sistema, sinó a un problema amb la connexió a Internet.

Pel que fa al temps, també cal dir que el fet d'introduir el sistema a la verificació dels productes, el temps de verificació d'aquests no es veu afectat. Aquest fet pot ser afirmat perquè després del problema amb la connexió del verificador a Internet, el pas de treure un equip del verificador i posar-n'hi un altre, es va estar repetint cada 20segons. Fet que es veu reflexat amb la diferència entre la durada real i l'estimada del procés de verificació d'un lot de 1000 equips.

Durant les 6 hores que va durar la prova pilot, aproximadament cada mitja hora i mitjançant l'aplicació mòbil, es comprovava que no hi haguessin incidències. Primerament en el sistema que s'estava posant a prova i segonament en la pròpia verificació dels diferents equips. En cap cas es va detectar algun problema, i els resultats obtinguts a l'aplicació informaven que en el procés de verificació tampoc n'hi havia.

5.1.5. Conclusions de la prova pilot

Tal com s'exposa a l'apartat anterior, els resultats de la prova pilot van ser molt favorables. Aquest fet permet tenir la certesa que s'ha creat un sistema que compleix amb els seus requeriments i que si es desitja, permetrà disposar d'una informació que fins al moment era impossible i impensable d'obtenir.

Un aspecte important que s'ha aconseguit gràcies a la prova pilot ha estat el fet de conèixer el que costa, quantitativament, implantar aquest sistema a cada tipus de producte. Aquest fet permet a l'empresa tenir la informació suficient per decidir si implanta el sistema a tots els productes, només a alguns, o a cap.

Cal dir però, en referència al cost d'implementació, que el codi font creat pel primer verificador es podrà aprofitar (en un percentatge molt elevat) en altres verificadors. Aquest fet permetrà reduir les 4hores de programació a una o dues, fet que reduiria molt el cost d'implementació d'aquest sistema a cada producte.

5.2. Possibles ampliacions del sistema

En aquest punt es descriuen una sèrie d'ampliacions que es podrien portar a terme al sistema. Aquest fet però, no significa que el sistema creat tingui aquestes mancances.

Els punts descrits a continuació són aspectes que amb els coneixements adquirits desenvolupant el sistema es podrien resoldre fàcilment. No s'han afegit a aquest projecte però, perquè no es disposava del temps necessari per fer-ho i en algun punt calia fixar-ne els límits. En el moment en el que aquest sistema s'implementi a gran escala, algun d'ells molt possiblement s'hi introduirà.

- Una ampliació molt interessant que es podria portar a terme a aquest sistema és la de poder modificar paràmetres interns dels verificadors mitjançant l'aplicació mòbil.

És clar que per poder fer aquesta ampliació caldria modificar el programa de tots els elements que formen el sistema. Aquest fet però, no seria de gran complexitat ja que actualment es disposa d'uns coneixements elevats de cadascun dels elements. A més a més, les comunicacions ja estan establertes i només caldria que els paràmetres fossin uns altres.

Aquesta ampliació permetria a l'enginyer responsable de la producció solucionar moltes més incidències a distància.

- Un altre aspecte a "millorar" seria el fet de guardar els resultats de les verificacions en format valor. Actualment el verificador indica si les diferents verificacions han estat superades o no i aquest resultat és el que es guarda a la base de dades. El que es podria fer seria que el verificador continués donant el mateix resultat, però que la informació guardada a la base de dades fos el número obtingut. Per exemple, el resultat d'una verificació ha d'estar entre 10 i 15, i el verificador obté un 12. Com que el resultat es troba entre els límits, el verificador dona la verificació com a bona i informa d'aquest resultat a l'operari. En el moment d'enviar la informació a la base de dades però, en comptes d'enviar "correcte" o "no correcte", enviaria el 12.

Aquest fet feria que de les estadístiques que es poguessin extreure amb les dades de la base de dades se'n pogués extreure molta més informació. Es podria saber si els diferents equips passen les verificacions molt aprop del llindar o si les passen sobradament. Fet que permetria millorar els equips perquè aquests treballessin sempre en nivells òptims.

Portar a terme aquesta modificació consistiria en modificar bàsicament l'aplicatiu servidor, ja que seria l'encarregat de gestionar totes aquestes dades. Amb els coneixements adquirits del llenguatge *PHP* però, no seria molt complicat.

- Finalment, per millorar la traçabilitat dels productes, si cadascun que passa pel verificador fos identificat amb un codi únic, se li podrien guardar els resultats de les verificacions amb la seva "matrícula".

D'aquesta manera, si mai un client retornés l'equip avariats, es podrien conèixer els resultats que es van obtenir durant la verificació. D'aquesta manera es podrien saber dues coses. La primera, si l'equip havia estat verificat per l'empresa subcontractada. I la segona, si l'averia estava causada perquè el resultat d'una verificació es va obtenir massa aprop del llindar.

Afegir aquesta informació al sistema seria molt senzill ja que només caldria afegir una columna a la *Taula de Resultats* de la base de dades amb "matrícula" de l'equip.

5.3. Conclusions

En el moment en el que es va iniciar aquest projecte, uns objectius molt clars amb la intensió de ser assolits, van ser establerts. Com a objectiu principal es pretenia donar resposta a una necessitat real d'una empresa que, per poder assolir-lo caldria aplicar coneixements teòrics i pràctics adquirits durant el màster TAI. Per últim, degut a l'atracció que m'havia creat l'estudi d'aquestes matèries, també volia ampliar-los i poder crear, d'aquesta manera, una solució que pogués donar resposta a la totalitat de les necessitats de l'empresa.

Una vegada definits els objectius, calia estudiar les necessitats de l'empresa i els requeriments que aquesta tenia. Seguidament fer un estudi de les possibles solucions i finalment, dissenyar i desenvolupar la solució escollida.

Desenvolupar un sistema format per varis elements té una complexitat elevada, sobretot si de la gran majoria d'aquests no en coneixes ni el funcionament ni la seva programació. Aquest fet és el que em va passar a mi, ja que de servidors web, d'aplicatius servidor i de bases de dades pràcticament només en coneixia el nom. Pel que fa al les aplicacions per dispositius mòbils, aquestes s'havien estudiat durant el màster i amb els diferents verificadors hi treballava diàriament.

El concepte augmenta de complexitat quan els diferents elements han de ser capaços de comunicar-se entre ells. Per si això fos poc, la comunicació ha de ser possible des de qualsevol part del món, per tant, Internet és l'única possibilitat que existeix. D'experiència en aquest camp jo no en tenia gens i va caldre, igual que amb els servidors i la base de dades, començar a estudiar des de zero el seu funcionament i desenvolupament.

El fet d'haver de començar a desenvolupar sobre plataformes que desconeixes totalment fa que avançar sigui molt complicat. Davant de qualsevol objectiu, per petit que sigui, cal dedicar moltes hores a la recerca d'informació i a l'aprenentatge de la seva configuració o desenvolupament. Sovint però, aquesta recerca s'allarga molt més de lo desitjat i la planificació de les tasques feta prèviament, cal refer-la de nou. Alguns exemples que hagin forçat una replanificació total són els següents:

- Establir una comunicació entre l'aplicació mòbil i l'aplicatiu servidor. Degut al fet de desconèixer totalment el funcionament d'un aplicatiu servidor i el llenguatge escollit per programar-lo, aconseguir la primera comunicació entre aquests dos elements es va convertir en una tasca molt complexa.
- Establir la primera comunicació entre l'aplicatiu servidor i el motor de gestió de la base de dades. Igual que en el cas anterior, el fet de desconèixer totalment el tema va fer que calgués destinar-hi moltes més hores de les previstes.
- L'últim cas no és com els altres dos, però també va acabar convertint-se en un punt crític. El tema del que em refereixo és el de crear una estructura de la base de dades que permetés cerques ràpides.

Quan ja n'havia creat dues que no oferien el que es necessitava el sistema i estava dissenyant la tercera, ja no sabia si el problema era degut a l'estructura de la base de dades, al tipus de base de dades o al tipus de cerca que hi feia. En aquest punt ho vaig passar malament perquè no sabia on era el problema, si anava pel camí correcte o si calia buscar a un altre lloc. Finalment però, després de molta documentació i estudi, amb la tercera estructura de base de dades es va aconseguir l'objectiu.

Una vegada acabat el projecte i comprovat el seu bon funcionament amb una prova pilot, es pot dir que els resultats obtinguts són molt positius. Els objectius marcats a l'inici han estat complerts, tan a nivell de requeriments del sistema per part de l'empresa, com a nivell personal.

L'objectiu principal consistia en donar resposta a una necessitat real d'una empresa. Els requeriments que aquesta posava però eren que no es podia augmentar el temps de verificació, els resultats s'havien de poder obtenir durant la pròpia

verificació, no podia ser massa complicat obtenir els resultats de les verificacions i finalment, l'accés a la informació havia de ser restringit.

A la prova pilot es va poder comprovar que el temps de verificació no augmenta, per tant, el cost del producte tampoc, significant així que el requeriment s'ha complert. Durant la prova pilot també es va poder comprovar que mitjançant l'aplicació mòbil es pot disposar dels resultats en el moment de la verificació, confirmant així que el segon requeriment també s'ha complert. Que amb 10 o 15 segons un usuari s'hagi identificat i estigui veient els resultats a l'aplicació mòbil, significa que el tercer requeriment també s'ha assolit. Finalment, que només els usuaris donats d'alta puguin obtenir els resultats de les verificacions fa que el quart requeriment també hagi estat assolit. Val a dir doncs que havent aconseguit els quatre requeriments de l'empresa permet afirmar que l'objectiu principal d'aquest projecte ha estat assolit.

A títol personal els objectius marcats han estat superats molt per sobre de les expectatives, sobretot els referents a aspectes tècnics. Els coneixements d'aquesta àrea adquirits al màster es van poder aplicar perfectament i van ser de gran ajuda per disposar d'una base sòlida. Pel que fa referència a ampliar-los, cal dir que potser ha calgut fer-ho massa i tot. La necessitat d'haver de dedicar tantes hores en ampliar els coneixements tècnics per poder donar solució al projecte, ha fet que aspectes de la gestió d'aquest no s'hagin pogut treballar tant com s'hagués desitjat.

La meva valoració sobre la creació d'aquest projecte però, és clarament positiva. El fet d'haver aconseguit portar a terme un projecte d'aquestes característiques on prenen part tants elements diferents i tenen la capacitat de comunicar-se mitjançant Internet, és per mi una fita molt important. Una fita que si el dia que vaig començar el projecte hagués sabut lo gran que seria i la quantitat d'informació que hauria d'adquirir, no sé si m'hagués vist en cor de posar-m'hi.

Bibliografia

○ **Llibres:**

- L. MURPHY, Mark. *The Busy Coder's Guide to Android Development*. Estats Units: CommonsWare, 2008.
- BAUCCELLS, Albert. *Android. Conceptos Básicos para desarrolladores*. Mòdul teòric de l'assignatura *Eines pel Control i Desenvolupament de Sistemes del màster TAI*.
- JASON GILMORE, W.. *Beginning PHP and MySQL: From Novice to Professional*. Estats Units: Apress, 2008.
- COBO, Angel; GÓMEZ, Patricia; PÉREZ Daniel, ROCHA Rocío. *PHP y MySQL Tecnologías para el desarrollo de aplicaciones web*. Espanya: Ediciones Díaz de Santos, 2005.
- DUNCAN William R. (Director of Standards). *A Guide to the Project Management Body of Knowledge*. PMBOK Guide, Estats Units, Tercera Edició, 2004.
- ESCORSA, Pere; VALLS, Jaume. *Tecnología e innovación en la empresa*. Barcelona: Edicions UPC, 2004.
- DRUCKER, Peter. *La disciplina de la innovación*. Harvard Deusto Business Review, 1986.
- ROSSEGER, Gerhard. *The Economics of Production and Innovation*. Oxford: Pergamon Press, 1987.
- OCDE. *Manuel d'Oslo. Principes, directeurs proposes par L'OCDE pour le recueil et l'interprétation des données sur l'innovation technologique*. París, 1992.
- PORTER, Michael. *Competitive Strategy*. The Free Press, 1980. Traducció espanyola, *Estrategia competitiva*. Mèxic: CECSA, 1982.

○ **Pàgines web referents a l'aplicació mòbil:**

- Consultades el dia 09/02/2012:
 - <http://developer.android.com/sdk/installing.html>
 - <http://www.android.es/foro/viewtopic.php?f=2&t=14&sid=2b578cf9acbeabffd89be487e6976f7d>
- Consultades el dia 15/02/2012:

- <http://developer.android.com/guide/topics/fundamentals.html>
- Consultades el dia 18/02/2012:
 - <http://developer.android.com/design/index.html>
 - <http://www.javacodegeeks.com/2010/10/android-full-application-tutorial.html>
- Consultades el dia 20/02/2012:
 - <http://www.connorgarvey.com/blog/?p=97>
 - <http://developerlife.com/tutorials/?p=288>
 - <http://developer.android.com/guide/components/activities.html>
 - <http://developer.android.com/guide/components/intents-filters.html>
- Consultades el dia 18/03/2012:
 - <http://developer.android.com/guide/topics/manifest/manifest-intro.html>
 - <http://developer.android.com/guide/topics/ui/declaring-layout.html>
- Consultades el dia 05/05/2012:
 - <http://androideity.com/2011/07/06/ciclo-de-vida-de-una-actividad/>
- Consultades el dia 18/06/2012:
 - <http://developer.android.com/about/dashboards/index.html>
- **Pàgines web referents a l'aplicatiu servidor:**
 - Consultades el dia 01/03/2012:
 - <http://www.phpcode.net/category/php/>
 - <http://php.net/manual/es/index.php>
 - <http://www.000webhost.com/>
 - http://www.w3schools.com/php/php_syntax.asp
 - Consultades el dia 08/03/2012:
 - <http://php.net/manual/es/ref.array.php>

- <http://www.nociondigital.com/webmasters/php-tutorial-funciones-para-el-tratamiento-de-cadenas-de-caracteres-detalle-96.html>

- **Pàgines web referents a la comunicació entre l'aplicació mòbil i l'aplicatiu servidor:**
 - Consultades el dia 14/04/2012:
 - <http://www.coderzheaven.com/2011/04/08/android-phpmysql-connection/>
 - <http://stackoverflow.com/questions/2696190/how-to-connect-android-to-server>

 - Consultades el dia 15/22/2012:
 - <http://www.chuidiang.com/java/sockets/socket.php>
 - <http://thinkandroid.wordpress.com/2009/12/30/getting-response-body-of-httpresponse/>
 - <http://docs.oracle.com/javase/7/docs/api/java/net/URLConnection.html>
 - <http://stackoverflow.com/questions/6188379/how-to-read-pdf-content-from-server-and-save-in-file>

 - Consultades el dia 22/04/2012:
 - <http://www.sgoliver.net/foro/viewtopic.php?f=5&t=48>
 - <http://www.ibm.com/developerworks/opensource/library/os-android-networking/>
 - <http://docs.oracle.com/javase/tutorial/networking/sockets/index.html>

- **Pàgines web referents a la base de dades:**
 - Consultades el dia 05/05/2012:
 - <http://es.wikipedia.org/wiki/MySQL>
 - http://ca.wikipedia.org/wiki/Base_de_dades

 - Consultades el dia 06/05/2012:
 - http://www.programacion.com/articulo/tutorial_basico_de_mysql_189

- <http://www.000webhost.com/>
- <http://www.slideshare.net/juanmazalazar/bases-de-datos-mysql-y-phpmyadmin>
- Consultades el dia 12/05/2012:
 - http://www.wikilearning.com/tutorial/introduccion_a_las_bases_de_datos-clave_principal_o_primary_key/10147-4
 - <http://tecnoclara.wikispaces.com/file/view/Bases+de+Datos+Relacionales+con+MySQL.pdf>
- **Pàgines web referents a la comunicació entre l'aplicatiu servidor i la base de dades:**
 - Consultades el dia 13/05/2012:
 - <http://www.tizag.com/mysqlTutorial/mysqlinsert.php>
 - <http://php.net/manual/es/index.php>
 - Consultades el dia 19/05/2012:
 - <http://php.net/manual/es/function.mysql-query.php>
 - <http://php.net/manual/es/function.mysql-select-db.php>
- **Altres pàgines web visitades:**
 - Consultades el dia 16/06/2012:
 - www.termcat.cat/

