



Universitat
Autònoma
de Barcelona



escola
d'enginyeria

**5649-1 : DESENVOLUPAMENT D'INTERFÍCIES D'USUARI I
VISUALITZACIÓ D'IMATGE MÈDICA EN DISPOSITIUS MÒBILS
ANDROID**

Memòria del Projecte Fi de Carrera
d'Enginyeria en Informàtica
realitzat per
Sergi Llamas Guerrero
i dirigit per
Ramón Grau Sala
Bellaterra, 12 de setembre de 2014

Agraeixo aquest projecte al centre mèdic UDIAT del parc Taulí de Sabadell, per la paciència i la oportunitat que ens han donat per a poder realitzar el treball.

També agraeixo la feina i el suport rebut pel director que l'ha fet possible.

Finalment, no em puc oblidar de la meua parella per estar recolzant-me en tot moment i fer que no decaigués en els moments més foscos i durs.

Moltes gràcies.

ÍNDEX

1. Introducció.....	5
1.1. Motivació	5
1.2. Objectiu i abast	5
1.3. Organització del projecte.....	6
2. Interacció Persona-Ordinador.....	9
2.1. Història.....	9
2.2. Observació participant.....	11
2.2.1. Entrevista.....	12
2.2.2. Personas.....	13
2.3. Mapeig de processos.....	14
2.4. Prototipatge ràpid.....	16
2.4.1. Storyboarding.....	16
2.4.2. Prototips de paper.....	17
2.5. Avaluació heurística.....	18
3. Programació Android.....	25
3.1. Inici.....	25
3.2. AndroidManifest.xml.....	26
3.3. Activities.....	27
3.4. Fragments.....	27
3.5. Layouts.....	28
3.6. Action Bar.....	28
3.7. Pop-ups.....	28
3.7.1. Toasts.....	29
3.7.2. Dialogs.....	29
3.8. Adapters.....	30

3.9. AsyncTask.....	32
3.10. ViewPager.....	33
3.11. Traduccions.....	35
4. Aplicació final.....	37
4.1. Action Bar i navegabilitat.....	37
4.2. Carregar estudi.....	40
4.3. Carregar sèrie.....	43
4.4. Mostrar informació de les imatges.....	48
4.5. Canviar color.....	48
4.6. Comentaris.....	52
4.7. Compartir.....	55
4.8. Informació del pacient.....	55
4.9. Títol de l'aplicació.....	59
5. Planificació.....	61
6. Conclusions i futur del projecte.....	65
6.1. Conclusions.....	65
6.2. Futur del projecte.....	66
7. Índex d'imatges.....	67
8. Bibliografia.....	69
9. Annexos.....	71
9.1. Storyboards.....	71
9.2. Prototip de paper.....	73
9.3. Entrevista.....	75
9.4. Personas.....	81
9.5. Avaluació heurística.....	91
9.6. Full d'avaluació heurística.....	93
10. Abstract.....	96

CAPÍTOL 1

INTRODUCCIÓ

Durant els darrers anys hem estat experimentant un augment exponencial de l'ús dels dispositius anomenats *smartphones* i de les tauletes. Aquests aparells són l'evolució dels telèfons mòbils que han incorporat, gràcies a l'avanç de la tecnològica, funcionalitats molt més complexes. Degut a això, les empreses, cada vegada més, s'estan preocupant pel desenvolupament d'aplicacions i n'estan llençant al mercat per a què els usuaris se les descarreguin als seus mòbils i tauletes, d'aquesta manera es pot interactuar amb el client d'una manera més directa i personal. Aquest fet també ha estat possible pel servei de transmissió de dades que ofereixen les companyies telefòniques, que permeten l'accés a Internet amb aquests dispositius.

Dins dels *smartphones* s'han imposat els dispositius d'Apple que incorporen el sistema operatiu iOS i els mòbils amb Android.

1.1. Motivació

Amb aquest treball es pretén millorar els diagnòstics dels metges mitjançant la visualització d'imatges. Actualment els metges disposen d'equips molt potents que permeten un diagnòstic molt precís gràcies a la qualitat d'imatge i la manipulació d'aquestes que poden oferir, de manera que no és aquest l'objectiu. En canvi, als metges que hi ha de guàrdia els permetria tenir mobilitat i no haver de passar-se totes les hores de guàrdia a casa per poder atendre les peticions que els puguin arribar, si disposen d'un dispositiu mòbil que els hi permeti validar una conclusió d'un altre o donar una segona opinió amb certes garanties.

1.2. Objectiu i abast

L'objectiu d'aquest projecte és dissenyar i programar una interfície d'usuari d'una aplicació per a la visualització d'imatges mèdiques per a dispositius mòbils amb sistema operatiu Android.

La finalitat d'aquest treball és contactar amb un centre mèdic real, per saber les necessitats no cobertes dels metges i com podem ajudar a millorar la feina de diagnòstics amb una aplicació per a tauletes (o dispositius mòbils en general). En aquest cas hem contactat amb la UDIAT Centre Diagnòstic del parc Taulí de Sabadell.

La decisió de implementar el projecte per a Android és deguda a la poca explotació d'aquest mercat en l'àmbit mèdic.

1.3. Organització del projecte

Durant el capítol 2.1 es fa una introducció històrica i es repassa el context històric en que es pot situar el naixement de la Interacció Persona-Ordinador.

Sovint no es sap els problemes, requeriments, o necessitats que tenen els potencials usuaris, per tant, s'ha de començar per fer una anàlisi de requeriments, fer una cerca de quin perfil de persones pot fer servir el nostre projecte, per a què el faran servir, en què els beneficiarà... Aquest estudi s'anomena observació participant, i es descriurà a l'apartat 2.2. Dins l'observació participant hi ha diverses maneres d'obtenir informació dels possibles usuaris. Una d'elles és fer enquestes estàndards per a molta gent, una altra és fer entrevistes... Per a aquest procés hem seguit el que s'explica a [2], un llibre de Mike Kuniavsky.

A la secció 2.3 es detalla el procés que ha de seguir un usuari de l'aplicació a partir de les observacions de l'apartat anterior. A partir d'un cas d'ús es procedeix a explicar pas per pas les accions que ha de fer per arribar al seu objectiu.

Una vegada finalitzat el procés d'observació participant continuarem amb els prototips de paper, que detallarem a l'apartat 2.4. En comptes de començar a programar, és bona idea fer dibuixos, representar la idea en paper per a que l'avaluació es faci sobre un prototip. Això permet que els canvis que s'hagin de fer siguin molt més fàcils i comportin menys cost en tots els sentits que si s'hagués de modificar el programa real. Quan s'hagi donat per finalitzada l'avaluació i el prototip hagi estat refinat fins a un punt acceptable aleshores ja es podrà procedir a la implementació en codi.

Quan s'ha fet un primer disseny, després s'ha d'avaluar. La tècnica d'avaluació que es farà servir al projecte és l'avaluació heurística, que s'explicarà a l'apartat 2.5. Possibles usuaris o gent interessada en el programa són els que han d'avaluar-lo. Han d'identificar les mancances i els errors de l'aplicació entre altres funcions. En general, com s'ha insinuat anteriorment, l'avaluació es fa sobre el prototip de l'aplicació i no sobre l'aplicació real. Quan s'ha finalitzat aquest pas, s'ha de modificar l'aplicació per adaptar-la millor als requeriments dels usuaris.

Al capítol 3 s'expliquen els punts més importants de la programació en Android. Cada secció introdueix un aspecte important que s'ha hagut de fer servir per a la programació de l'aplicació.

Al capítol 4 es desglossa el funcionament de l'aplicació final, es veu com ha quedat finalment i es relaciona amb els capítols anteriors, de manera que queda reflectit el que s'ha après tant en interacció Persona-Ordinador com en programació Android.

Al capítol 5 es mostra la planificació del projecte.

Al capítol 6 s'exposen les conclusions, el que hem après, els errors que hem comès i les millores que queden pendents per a una possible ampliació del projecte en un futur.

El capítol 7 és l'índex de les imatges que apareixen a la memòria.

El capítol 8 està compostat per la bibliografia.

El capítol 9 són els annexos.

Per acabar, el capítol 10 és l'*abstract*, traduït al castellà i l'anglès.

CAPÍTOL 2

INTERACCIÓ PERSONA-ORDINADOR

Per dur a terme el procés de disseny, hem seguit les lliçons impartides pel professor Scott Klemmer mitjançant [1] i les hem pres com a base. Als vídeos que trobem a la pàgina web s'aprèn que és molt important fer un bon disseny perquè els mals dissenys comporten un cost massa alt. Si no se li dóna la importància suficient a aquest procés, es pot arribar a implementar una aplicació on l'usuari ha de perdre molt de temps cercant com pot fer una determinada operació, o fins i tot pot creure que està fent una acció i l'aplicació n'està fent una altra totalment oposada. Això pot tenir un impacte econòmic i de temps molt elevat o fins i tot pot costar vides humanes.

Hi ha tres elements importants a tenir en compte; el primer i més important és la persona, que és a qui ha d'anar enfocada la feina, el segon és la màquina, i el tercer és la interfície, que és el mitjà entre l'usuari i la màquina. La Interacció Persona-Ordinador (a partir d'ara l'anomenarem HCI, ja que en anglès és Human-Computer Interaction) és el procés iteratiu de disseny, implementació i avaluació de les interfícies d'usuari que relaciona els tres elements que comentàvem unes línies més amunt. Explicarem detalladament les fases que hem seguit al nostre projecte i les desglossarem en aquest capítol, però abans en farem un resum històric.

2.1. Història

Per fer notar que l'HCI no ha sorgit els últims anys, sinó que ha estat present des de molt temps enrere, repassarem alguns exemples que ens han semblat interessants i representatius.

Scott Klemmer explica a la quarta classe de [1] que el naixement de l'HCI es podria situar al juliol de l'any 1945 quan Vannevar Bush va escriure l'article "As we May Think" per a l'Atlantic Monthly, que es pot consultar a [3].

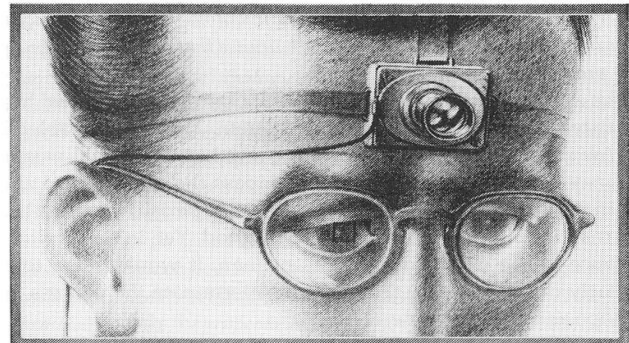
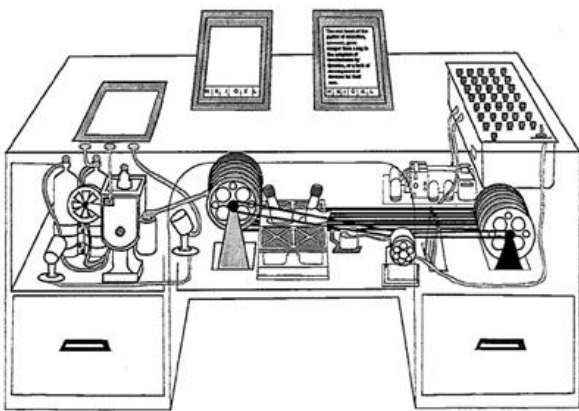
Repassarem algunes de les primeres idees que van donar lloc a l'HCI i les primeres interfícies d'usuari, que finalment és el que més ens interessa.

Vannevar Bush va estar sempre relacionat amb el camp de la investigació informàtica i la tecnologia. Entre altres funcions, va ser vicepresident i degà de l'escola d'enginyeria de l'MIT (Institut de Massachussets de Tecnologia) i més tard va ser nomenat director de la Office of Scientific Research and Development, sent així cap de la comunitat de científics que van crear la bomba atòmica als inicis de la segona guerra mundial. Dues de les seves aportacions més influents van ser

construir la primera computadora analògica que va anomenar analitzador diferencial i sobretot el Memex.

A l'article "As we May Think" es preguntava com es podria millorar la vida de les persones en temps de pau, ja que al 1945 ja estava acabant la guerra, amb mitjans tecnològics i científics des d'un punt de vista centrat en la persona. Va ser un visionari plantejant una màquina que va anomenar Memex capaç d'emmagatzemar i transmetre informació per tot el món, ja que segurament la World Wide Web, creada molts anys més tard d'aquest article, s'apropa molt a la idea que ell va tenir.

El problema principal que veia Bush és que cada vegada es disposava de més informació i li preocupava que no s'emmagatzemés de manera adequada per poder-la aprofitar i accedir-hi quan convingués. Deia que hi ha maneres d'ordenar la informació com l'alfabètica o la numèrica, però la ment humana no funciona d'aquesta manera, sinó que ho fa per associacions segons els camins neuronals que s'han format al cervell. Aleshores és quan sorgeix la idea del Memex (Figura 1 – esquerra), que seria un dispositiu que constaria d'una taula amb un teclat i palanques que permetria la consulta de dades emmagatzemades en microfilms que serien posteriorment projectats en unes palanques translúcides, podent realitzar anotacions sobre dites consultes. A més, també va pensar en que la gent no es passa tot el dia davant d'un escriptori, i que la majoria d'informació no és escrita, sinó que és visual, aleshores es va imaginar que en un futur la gent duria una càmera al cap que es faria servir per capturar informació (Figura 1 – dreta).



A scientist of the future records experiments with a tiny camera fitted with universal-focus lens. The small square in the eyeglass at the left sights the object (*LIFE* 19(11), p. 112).

Figura 1: Memex, càmera

El Memex no va ser mai creat però si que ha influït notablement per a que altres investigadors poguessin aprofitar aquestes idees.

A l'actualitat la immensa majoria de persones porten un telèfon mòbil que els permet fer una fotografia i hi ha aplicacions que permeten obtenir informació a partir d'imatges del món real, per tant també es va apropar a la realitat en descriure el futur amb aquesta idea de càmera, i més si tenim en compte un objecte tan actual com les Google Glass.

Poc després es van inventar els primers ordinadors programables, però com es pot veure a la fotografia de la Figura 2, considerada la primera computadora digital, la interfície d'usuari és molt millorable.

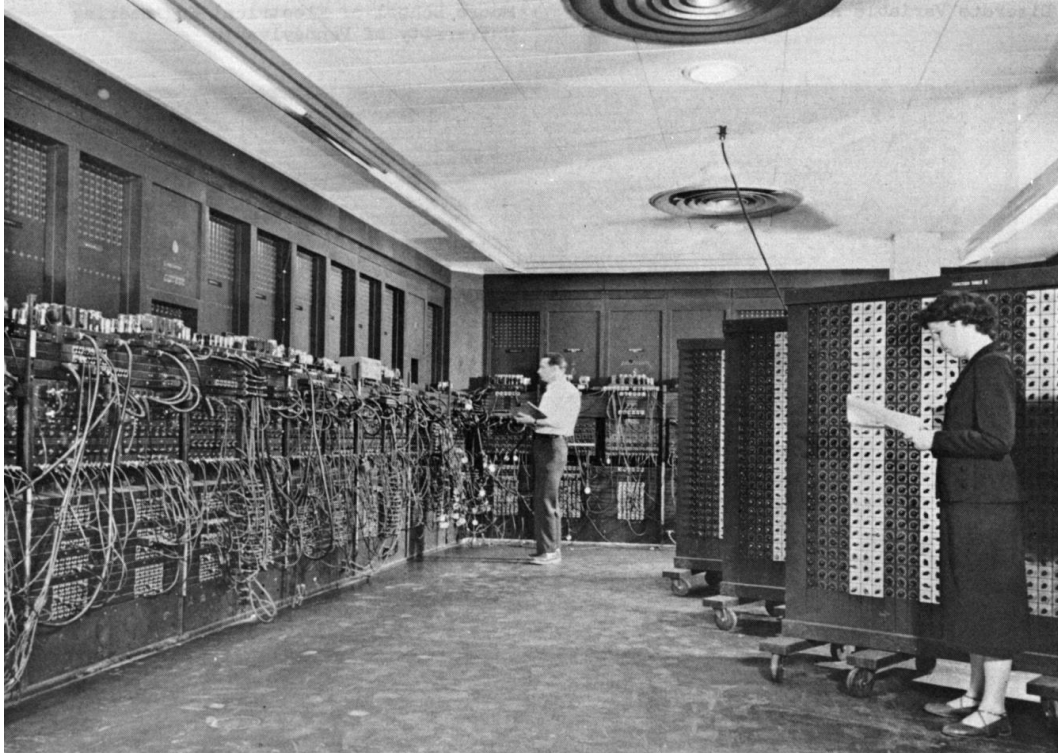


Figura 2: ENIAC

L'experiència de la interacció de l'usuari amb l'ordinador va començar a millorar a partir de que Grace Hopper s'inventés el primer compilador, i de la seva idea de millorar les eines per poder donar accés a la programació de computadors a molta més gent. Això últim va contribuir a que una onada de nous programadors possessin les bases per a que s'estengués l'ús ordinadors personals.

Aquests són només els primers exemples d'un procés que encara dura en el temps on encara podríem aprofundir-hi fins i tot en més detall.

2.2. Observació participant

Quan es vol començar a crear un producte nou, és necessari fer una cerca de necessitats per assegurar-se de que es cobreixen les mancances que té la gent a l'hora de dur a terme una activitat.

Per exemple, en el nostre cas, cal veure quines aplicacions mòbils existeixen al mercat, quin producte fan servir actualment els metges i amb quines dificultats es troben diàriament per poder oferir solucions amb el projecte que es vol fer.

D'aquesta manera es descobreix quins aspectes s'han de potenciar i quins estan ja explotats i no val la pena millorar.

Hi ha varies tècniques que permeten obtenir una idea de les tasques i escenaris als que es vol donar suport abans de començar a treballar en el producte. Les que hem fet servir s'expliquen amb més detall a les següents seccions.

2.2.1. Entrevista

L'entrevista és una eina molt valuosa per a la cerca d'una millor experiència d'usuari i per dur a terme aquest procés hem seguit el que es mostra a [2].

El primer pas a l'hora de preparar una entrevista és escollir els participants. Aquests poden ser representació d'usuaris finals, usuaris reals d'altres aplicacions similars, o fins i tot pot ser que no siguin usuaris potencials. El nostre cas és el tercer, ja que vam concertar l'entrevista amb representants de l'equip informàtic del parc Taulí.

Encara que sembli una tasca trivial i senzilla, la realitat és que és importat ja que pot aportar informació molt rellevant. S'ha de tenir en compte molts detalls i seguir una sèrie de premisses per a que l'entrevista aporti informació de profit.

Un primer punt a tenir en compte és tenir clar a qui s'està entrevistant per saber quin tipus de preguntes poden aportar-nos més informació.

Si no podem obtenir precisió exacta en la resposta és millor obtenir una aproximació. Tota informació pot ser valuosa.

És bo poder entrevistar a diversos interessats (*stakeholders*), usuaris finals, usuaris d'aplicacions semblants, persones que no tenen res a veure, etc. En el nostre cas només ha estat possible entrevistar-nos amb l'equip informàtic.

Encara que una pregunta no estigui preparada, és una bona pràctica ser curios i preguntar a l'adonar-se que una pregunta nova pot aportar informació valuosa.

Hi ha un tipus de preguntes que cal evitar realitzar perquè ens poden dur a respostes contraproduents o en el millor dels casos perdre el temps ja que no ens aportaran informació:

- Preguntes tendencioses. Per exemple: "Li agradaria que l'aplicació anés més ràpid?".
- Preguntes sobre la freqüència amb que es realitzen accions. Poden dur a mentir a l'entrevistat, encara que no se n'adoni. Una pregunta com "Quantes hores dedica a la setmana a anar al gimnàs?" pot fer que la resposta sigui massa optimista i no s'ajusti a la realitat.
- Valoracions numèriques de quant li agrada una cosa. Cada persona té la seva escala de valors i cada puntuació té un significat diferent per a cadascú.

- Preguntes de resposta binària o tancada. Poden fer que la resposta sigui massa pobre i es quedin idees i detalls importants sense explicar.

En canvi unes pràctiques interessants que poden aportar-nos respostes de molt valor són:

- Fer preguntes obertes. D'aquesta manera la resposta serà més llarga i detallada i per tant amb més possibilitat d'extreure'n dades valuoses.
- Esperar uns segons encara que hi hagi silenci perquè l'entrevistat es deixi anar i aportï més contingut a la resposta.
- Ser pacient i no interrompre.
- Ajustar les següents preguntes a les respostes anteriors si cal.
- Fer servir un llenguatge familiar per a l'entrevistat, ja que d'una altra manera pot ser que no entengui la pregunta o no aportï tot el contingut que podria a la resposta.
- Insistir i aprofundir en les preguntes. Demanar exemples.
- Ser flexible i així causar sensació de confiança. En un ambient agradable és més fàcil que l'entrevista sigui més fluida, amena i amb més contingut important.

Enregistrar l'entrevista pot ajudar a recuperar detalls que s'han pogut escapar i a conduir millor l'entrevista ja que no és necessari estar apuntant les respostes i no cal estar per més d'una tasca alhora. Però en aquest cas, s'ha de preguntar sempre si no és cap inconvenient que sigui gravat i tenir en compte que segons el tipus d'entrevistat això pot comportar que es contingui més al respondre i això és contraproductiu.

És bo assajar i simular l'entrevista per trobar errors.

A l'annex 9.3 s'adjunta la preparació de l'entrevista amb els seus resultats.

2.2.2. Personas

Un mètode útil per a tenir sempre present els objectius dels potencials usuaris és crear persones fictícies que s'anomenen *personas*.

Cada *persona* ha de representar a un tipus qualsevol d'usuari potencial de l'aplicació. Tant pot ser una metgessa, com un infermer, com una persona que només hagi de fer servir l'aplicació en comptades ocasions.

A cada *persona* se li ha de donar un nom, cognoms, una personalitat, una descripció de la seva vida privada, de la laboral, s'ha de descriure les inquietuds que té... I fins i tot és convenient assignar-li una fotografia.

Aquesta tècnica ajuda a comunicar-se amb l'equip de treball per parlar de casos d'ús i quin tipus d'usuari pot trobar-se amb segons quins problemes en diferents ocasions.

Per a aquest projecte hem creat cinc *personas* amb diferents rols i ocupacions, que s'adjunten a l'annex 9.4.

2.3. Mapeig de processos

El mapeig de processos (o process mapping en anglès) consisteix a descriure un cas d'ús pas a pas amb el màxim de detall possible per detectar errors de disseny (o altres) amb antelació i fer que corregir-los sigui menys costós.

Un exemple per a la nostra aplicació és el següent:

- Arriba una notificació a la *tauleta*.
- La desbloquegem i observem que és un correu electrònic que han enviat al Gmail.
- Cliquem la icona del Gmail i llegim el missatge: és del cardiòleg i ens demana una segona opinió sobre la imatge de l'enllaç que ens envia adjunt al correu.
- Polsem l'enllaç i ens obre directament l'aplicació demanant-nos una contrasenya d'autenticació.
- Introduïm la contrasenya i ens mostra la imatge.
- Estem visualitzant una imatge.
- Si deixem el dit pitjant la pantalla surten totes les sèries de la imatge rodejant un quadre al centre, si les arrosseguem al quadre es mostrarà després a la pantalla, si les traiem del quadre no, fins a un màxim de 4, si en seleccionem més, surt un missatge d'error suggerint que es seleccionin menys imatges. A baix a la dreta hi ha el botó d'OK. Al pitjar-lo es torna a la imatge amb tantes imatges com s'hagin triat.
- Per passar a la següent sèrie passem la mà d'una banda a l'altra de la pantalla.
- Observem una petita irregularitat al centre de la imatge, col·loquem dos dits a la pantalla i els separem per fer zoom.
- Observem amb més claredat una irregularitat. Però encara no estem segurs per a fer un diagnòstic.
- A la *action bar* tenim les opcions de marcar, color, contrast, llum, compartir i sortir.
 - Premem la opció de color i se'ns obre un quadre de diàleg amb scrolling on es permet triar diferents paletes de colors, també hi ha la opció de cancel·lar. Triem la verda:

- Al quadre de diàleg hi apareixen dues barres verticals amb dos indicadors: un situat a la primera barra, i l'altre a la segona. La primera barra va del blanc al negre i la segona comença pel blanc i passa per les diferents tonalitats del verd fins arribar al negre. Cliquem el botó d'OK que apareix a la part inferior dreta del quadre de diàleg. Veiem com la imatge ha canviat de colors.
- Com encara no acabem d'apreciar clarament la irregularitat, anem a la opció de contrast:
 - Apareix un quadre de diàleg amb una *seek bar* que permet modificar el contrast. L'ajustem amb el dit movent-lo fins a la meitat de la barra. Premem el botó d'OK que apareix a la part inferior dreta del quadre de diàleg. Ha canviat el contrast de la imatge.
- Ara es pot apreciar com hi ha una taca fosca a la zona de la imatge que havíem vist. Per assegurar-nos polsem la opció de llum:
 - Apareix un quadre de diàleg amb una *seek bar* que permet modificar la il·luminació. L'ajustem amb el dit movent-lo una mica cap a la dreta per fer-la més fosca. Premem el botó d'OK que apareix a la part inferior dreta del quadre de diàleg. Ha canviat la llum de la imatge.
- Finalment volem marcar la zona que hem detectat conflictiva, per tant anem a l'opció de marcar:
 - Apareix una paleta de colors en un quadre de diàleg en una barra vertical. Amb el dit triem quin color volem. Aprem l'OK de la part inferior dreta.
 - A la *action bar* canvien els botons, ara hi apareixen: Marcar, esborrar, mida, guardar i cancel·lar.
 - Marcar està per defecte i permet dibuixar sobre la imatge amb el color i la mida que s'ha triat.
 - Esborrar permet esborrar les marques que s'han realitzat.
 - Al polsar mida apareix un quadre de diàleg amb *radio buttons* que permeten tirar com de gruixuda serà la línia de la marca.
 - Si polsem cancel·lar sortim sense desar cap marca i tornem a la vista de la imatge que teníem abans.
 - Si polsem guardar sortim amb la marca que hem fet.
- Després de marcar la imatge, volem enviar-li al radiòleg de guàrdia per a que ens doni una segona opinió. Anem a compartir.
 - L'aplicació obre el Gmail, i hi adjunta un enllaç a la base de dades d'imatges, on es troba la imatge que hem marcat.
 - Escrivim l'adreça electrònica del metge de guàrdia i li enviem un missatge dient-li que volem el seu diagnòstic sobre la imatge que li enviem.

- Tanquem el Gmail.
- Anem a la opció sortir i ens torna a la pantalla del menú principal de l'aplicació.

Com podem veure, aquest procés és força diferent del resultat final de l'aplicació. Això és degut a que el procés de disseny és iteratiu. Aquest tema s'abordarà a la secció 2.5.

2.4. Prototipatge ràpid

Una bona pràctica a l'hora de dissenyar una aplicació és no anar directament a programar, sinó fer una exploració de les necessitats, fer proves, veure diferents possibilitats... Per fer-ho, és útil fer simulacions en paper de la idea que es té. D'aquesta manera s'estalvia temps ja que no han de ser representacions precises i refer una part, ja sigui gran o petita, és molt menys costós que si fos un canvi al producte final.

Produir molts prototips diferents permet obtenir molta informació i una probabilitat més alta d'aconseguir un bon disseny.

2.4.1. Storyboarding

L'storyboard són una sèrie de vinyetes que han d'explicar una situació representativa i paradigmàtica de la idea que té el dissenyador de com ha de funcionar l'aplicació. No ha de ser molt elaborada ja que no pretén ser molt realista, sinó plasmar en un paper el que es vol dur a terme i com s'està pensant que es faci servir. És a dir, ha de representar un cas d'ús (o més d'un).

És una tècnica molt útil per a ensenyar-la als futurs usuaris per a que puguin avaluar-la, per a aquest projecte se n'han fet dos que s'adjunten a l'Annex 9.1.

L'storyboard ha de transmetre qui hi està involucrat, l'entorn on es farà servir el producte, quina tasca ha de dur a terme, la seqüència (quins passos es fan?, què fa que algú faci servir la aplicació?, quina tasca es representa?), i la satisfacció (què motiva a una persona a fer servir aquest sistema i no un altre?, què fa que les necessitats siguin satisfetes?, quines mancances ha de cobrir?).

2.4.2. Prototips de paper

Els prototips de paper són maquetes de l'aplicació final construïdes generalment amb paper o cartró que pretenen simular el funcionament d'aquesta estalviant temps i cost econòmic.

El prototip ha de ser el màxim de fidel al producte final, però sense afinar 100% al detall. També és interessant fer diferents maquetes per poder obtenir diferents experiències d'usuaris i triar la millor o la més convenient.

Aquesta tècnica permet corregir errors de disseny i detectar mala usabilitat i sobretot una correcció ràpida i barata. No és el mateix haver de reprogramar un botó en una aplicació o canviar totalment un menú que fer-ho en un dibuix de paper.

El professor Scott Klemmer ens proporciona als seus materials del curs "Human-Computer Interaction" [1] una llista de consells i trucs per a construir els prototips de paper d'una manera més eficient:

- Mantenir tot el material a un mateix lloc, ja que els ginys petits com els botons es poden perdre o fer malbé fàcilment.
- Treballar ràpid i fer elements reusables, com botons o pestanyes.
- Si hi ha algun component difícil de simular, millor no representar-lo i esperar a que l'usuari a qui s'ensenyi el prototip pregunti per ell i explicar-li verbalment com funciona.
- No tenir por de barrejar i combinar maquinari i programari. Per exemple, si la mida del dispositiu és important, pot ser bona idea fer una fotografia del dispositiu que s'utilitzarà i manipular el prototip en aquest context.

Aquest mètode es complementa molt bé amb l'avaluació heurística que veurem a la secció 2.5.

Veiem a la Figura 3 i Figura 4 dos exemples del prototip de paper que vam fer per a aquest projecte. A més, s'adjunta tot el material a l'Annex 9.2.

2.5. Avaluació heurística

Hi ha moltes tècniques per a avaluar el software. La més habitual és l'empírica on, de diferents maneres segons el mètode, hi ha persones provant l'aplicació. També hi ha mètodes més tècnics amb models més formals on s'extreuen les dades de simulacions.

L'avaluació heurística, que és el mètode que s'ha emprat en aquest projecte, és un mètode empíric basat en les opinions de la gent seguint un procés pautat.

Aquest procés es farà seguint les 10 heurístiques de Jacob Nielsen [4].



Figura 3: Prototip de paper 1

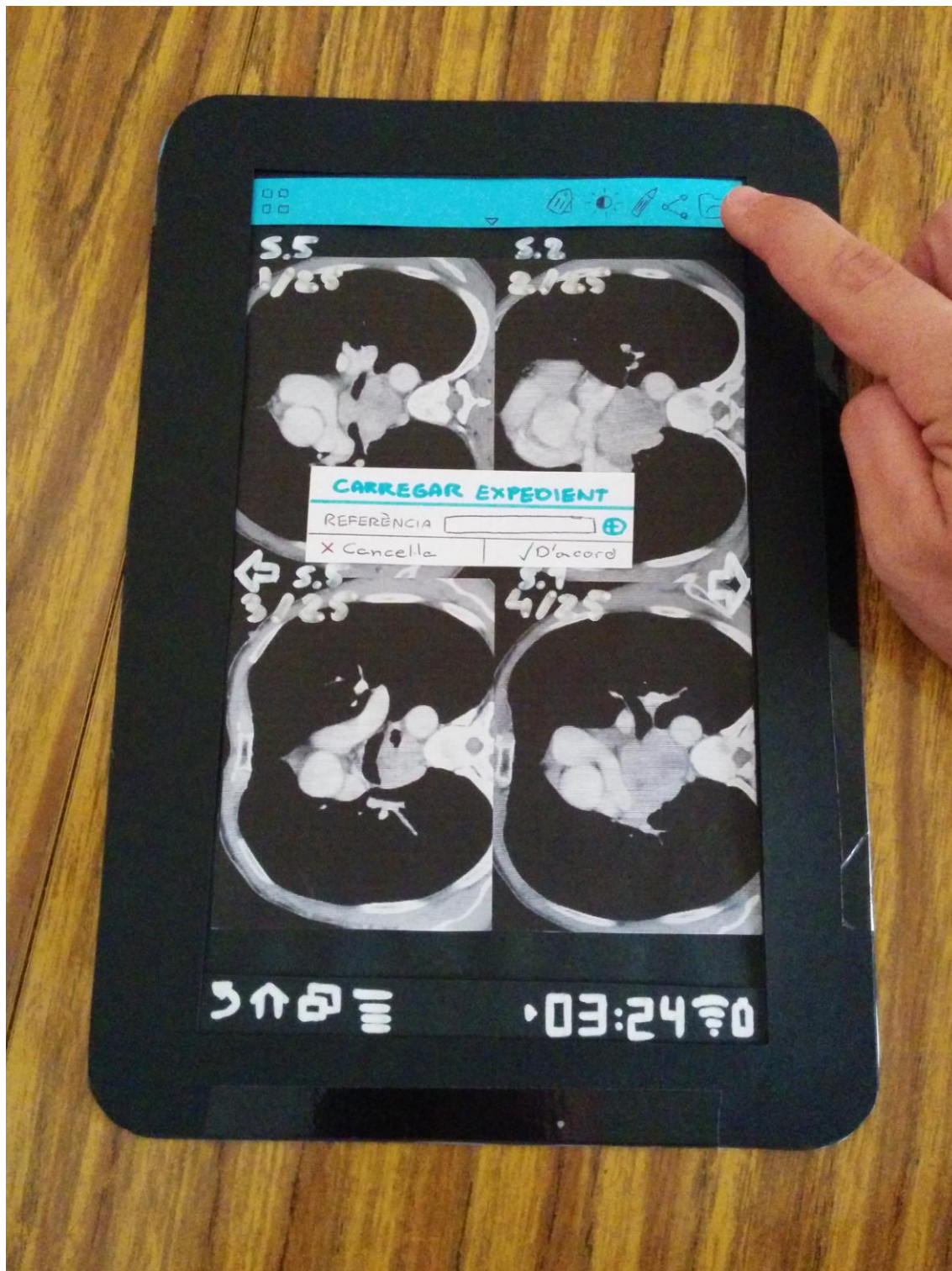


Figura 4: Prototip de paper 2

Per al projecte s'ha preparat un full que és el que haurien de respondre els participants al test de l'aplicació (o dels prototips de paper) per a obtenir un *feedback* que s'adjunta a l'annex 9.6, conjuntament amb una explicació del funcionament del procés, a l'annex 9.5, seguint el procés que s'explica a [5].

Seria important aconseguir que molta gent participi en el procés, ja que d'aquesta manera els resultats seran més significatius i ajustats a la realitat. Com es mostra a la Figura 5, gràfic extret de [5], hi haurà moltes persones que trobaran alguns errors i poques que en trobaran molts. Dificilment un sol usuari els trobarà tots, d'aquí ve la importància de que molta gent participi en el test.

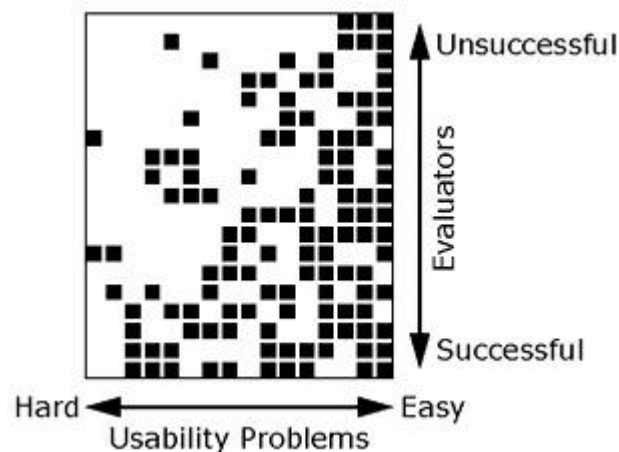


Figura 5: Avaluació heurística

La idea és fer un primer prototip de paper, i presentar-lo als participants en el procés amb el full d'explicació d'avaluació heurística. Una vegada han obtingut els resultats, els hem de solucionar, sempre que el cost sigui menor que no solucionar-los i tornar a repetir el procés fins que el prototip hagi arribat a un punt de refinament satisfactori. Una vegada assolit aquest punt, podem passar a programar l'aplicació real.

Com s'ha pogut veure, l'avaluació heurística és un procés iteratiu que segueix el model de la Figura 6.

Durant aquesta fase cal identificar els problemes menors abans de que els usuaris facin proves, d'aquesta manera no se'ls fa perdre temps i es poden centrar en problemes més grans. Una vegada obtinguts els resultats, s'ha d'avaluar els resultats i decidir què és convenient millorar i què és el que ja està funcionant. Una vegada polits els errors que s'han trobat obtenim una nova versió.

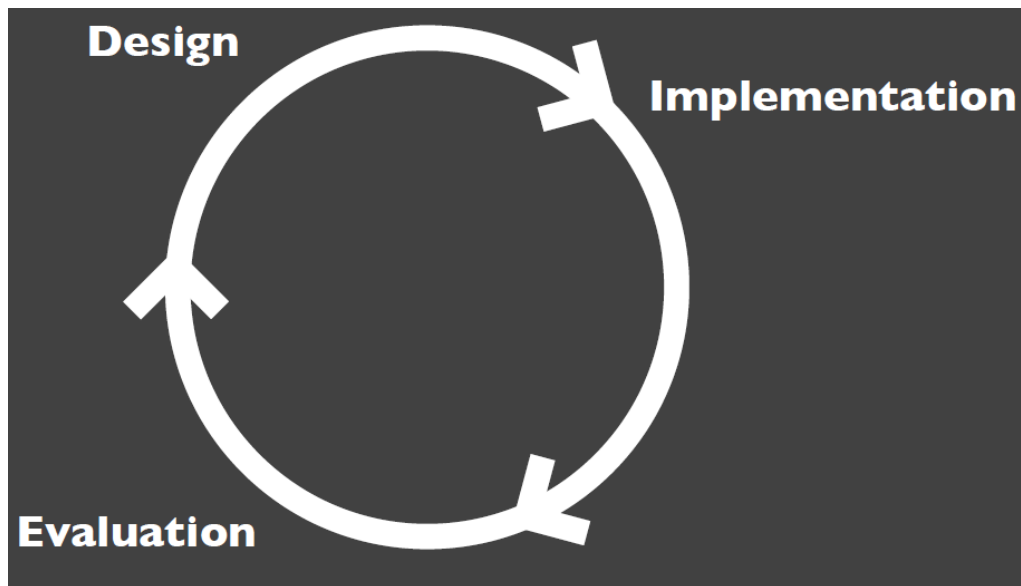


Figura 6: Procés iteratiu

D'acord amb les 10 heurístiques de Jakob Nielsen, l'aplicació que volem dissenyar ha de complir els següents punts:

- 1) **Estat del sistema:** L'usuari ha de saber què està passant i tenir informació del sistema:
 - Mostrar el temps: Per exemple, mostrar una barra de progrés quan es descarreguen dades que triguen més de 2 segons a mostrar-se.
 - Mostrar l'espai: En cas d'haver d'emmagatzemar dades, és important informar a l'usuari de l'espai disponible al seu dispositiu.
 - Mostrar els canvis i les accions: Si hi ha algun canvi, com la descàrrega d'un fitxer, si s'ha d'esborrar dades, si es desloguejarà a l'usuari... S'ha d'informar a l'usuari.
 - Mostrar els pròxims passos: Per exemple, si s'ha enviat un correu electrònic després d'un registre per a que es verifiqui l'usuari, és necessari mostrar la informació del que s'ha de fer.
 - Mostrar que s'ha completat una acció: Quan ha finalitzat la descàrrega d'un fitxer, quan l'usuari s'ha loguejat correctament, o quan s'acaba una acció que requereix un feed-back en general, mostrar un missatge que indica que ha finalitzat satisfactòriament ajuda a l'usuari a tenir la sensació de fluïdesa.

- 2) **Familiaritat amb el llenguatge de l'aplicació:** Hi ha d'haver una relació entre el llenguatge que utilitza l'aplicació i el llenguatge natural. L'usuari ha d'entendre els missatges del sistema, per tant aquest ha d'utilitzar un llenguatge conegut i familiar.
 - Llenguatge i metàfores familiars: Si la icona per fer una fotografia representa una càmera de fotos, l'usuari endevinarà ràpidament la funcionalitat d'aquell botó. O per exemple, tot i que ara ha caigut en desús, la icona del disquet és reconeguda i usada àmpliament per a desar.

- 3) **Llibertat i control per part de l'usuari:** L'usuari ha de poder desfer errors fàcilment i tenir el control sobre el que vol fer amb l'aplicació, no a l'inrevés. En un formulari amb varies pàgines, per exemple, s'ha de donar la opció de poder tornar a la pàgina anterior per a canviar algunes dades si l'usuari s'ha equivocat i ho recorda més tard.
- 4) **Estàndards i consistència:** No hi pot haver ambigüitats i l'aplicació ha de fer servir els estàndards. És important mantenir el mateix criteri de disseny per a tota l'aplicació, si es fa servir un botó a la dreta per a cancel·lar una acció i un a l'esquerra per confirmar, s'ha de mantenir l'ordre durant tota l'aplicació. Si a més és comú aquest ordre en moltes aplicacions és convenient mantenir-lo.
- 5) **Prevenició d'errors:** L'aplicació ha d'afavorir que no es produeixin errors. De vegades és bo mostrar missatges de confirmació per evitar errors greus.
 - Prevenir pèrdua de dades: Si una acció ha de produir canvis a les dades del dispositiu, ha de quedar molt clar què pot fer l'usuari i quines conseqüències tindrà.
 - Prevenir el desordre: Els elements que apareixen a prop per pantalla, han de tenir accions o conseqüències en comú, per no despistar a l'usuari o que passi molta estona buscant com pot fer el que vol fer.
 - Prevenir la mala interpretació: Per exemple, un pop-up amb el missatge "Té dades sense guardar. Premi D'ACORD per tornar i guardar o premi CANCEL·LAR per continuar" indueix a l'error.
 - Prevenir la introducció de dades errònies: Per exemple, als formularis mostrar el tipus de dades que són obligatoris i en quin format cal posar-lo. Si s'ha d'introduir una data, conduir a l'usuari a introduir-ho en el format que ens interessa evitarà errors. A més, si s'equivoca s'ha d'indicar quin error ha comès, en quin camp l'ha comès i què s'espera que hi escrigui en comptes de dir simplement "Has escrit les dades malament".
 - Prevenir restriccions innecessàries: No prohibir realitzar accions a l'usuari sense sentit en el context. Si s'ha de realitzar una cerca, és contraproductiu obligar a fer-la sobre un grup determinat, ja que és probable que l'usuari no sàpiga a quin grup ha de buscar, potser vol fer una cerca general.
- 6) **Evitar que l'usuari hagi de memoritzar o recordar:** Procurar que a la mateixa pantalla aparegui tota la informació necessària i que l'usuari no hagi de recordar el que posava a una pantalla anterior.
 - Evitar codis: Fer servir símbols representatius o altres elements més fàcils de recordar que números o lletres.
 - Mostrar previsualitzacions: En una llista ordenada alfabèticament és d'ajuda mostrar una previsualització de l'element abans de que s'hagi de triar.
- 7) **Flexibilitat i eficiència:** Permetre que els usuaris experts puguin avançar més ràpidament gràcies a que el sistema té mecanismes més complexos que requereixen experiència, però alhora permetre que els usuaris principiants no

tinguin problemes per fer servir l'aplicació amb fluïdesa.

- Dreceres: Els usuaris més habituats a Windows fan servir Ctrl+C per copiar o Ctrl+V per enganxar.
- Oferir opcions per defecte.
- Mostrar informació ambiental, per exemple, en un calendari a la setmana entrant.
- Flexibilitat en les activitats: Com poder donar-se de baixa d'un servei de propaganda per e-mail.
- Recomanacions: Si es té informació de l'usuari (potser l'ha donat a l registrar-se) es pot oferir recomanacions en una tria posterior.
- Destacar les opcions més habituals.

- 8) **Estètica i minimalisme:** No hi ha d'haver informació irrellevant a la pantalla i la informació ha de ser clara i concisa, sense elements superflus.
- 9) **Reconèixer, diagnosticar i recuperar errors:** Els missatges d'error han de ser expressats en un llenguatge amè indicant amb precisió el problema i suggerint una solució.
Per exemple, si a l'entrar a Internet només es mostra que s'ha produït l'error 404, la gran majoria d'usuaris no sabran quin problema hi ha, en canvi si es diu que no s'ha pogut carregar la pàgina del navegador, s'està proporcionant una informació en un llenguatge natural que pot entendre un públic molt més ampli. A més, és convenient mostrar una possible solució, com per exemple, que comprovi que està activat el wifi.
- 10) **Ajuda i documentació:** Tot i que l'aplicació ha de ser intuïtiva per a l'usuari, s'ha de proporcionar ajuda en el cas que l'usuari la requereixi. Ha de ser concisa, ràpida de trobar i entenedora.

CAPÍTOL 3

PROGRAMACIÓ ANDROID

Estem vivint una època on pràcticament el 100% dels habitants dels països més desenvolupats disposen d'un dels anomenats telèfons intel·ligents o *smartphones*. En aquest marc, les dues empreses que han dut a terme la revolució són Apple amb els iPhones amb el sistema operatiu iOS i Google amb Android arribant a una gamma molt àmplia de dispositius. És cert que als inicis BlackBerry va irrompre amb força oferint noves experiències als usuaris que disposaven de mòbils de la seva marca amb connexió de dades a Internet amb els que es podria considerar els primers telèfons intel·ligents comercialitzats a gran escala. Però amb el llançament del primer iPhone i de la irrupció d'Android al mercat, BlackBerry va perdre el seu lideratge. Actualment Microsoft amb Windows Phone és la tercera companyia amb més èxit però a molta distància de les dues primeres.

Moltes informacions afirmen que de tots els *smartphones* actius al món al 2013, el 80% duïen Android, amb una tendència a l'alça.

A aquest capítol explicarem els passos que hem seguit per a aprendre a fer una aplicació per a un dispositiu amb Android i els punts més destacats que hem seguit per a fer la que es proposa al projecte.

Per implementar-la hem après a programar de forma autodidacta seguit les instruccions que s'expliquen a la pàgina web per a desenvolupadors d'Android [6].

3.1. Inici

Per a començar a programar per a Android la manera més còmoda actualment és amb l'Eclipse. A [7] Android ens proporciona totes les eines que necessitem per començar a programar. Si ja tenim instal·lat l'Eclipse només cal descarregar l'SDK d'Android per afegir-li. Sinó, també podem descarregar la versió ADT Bundle que no és més que un Eclipse especial amb les funcions necessàries i l'SDK pre-instal·lat.

Abans de començar amb el codi, hem de tenir en compte que estem programant sobre un dispositiu mòbil, que pot disposar de connexió de dades, això suposa que s'ha de tenir cura a l'hora de llançar connexions a Internet. També tenen una bateria limitada, normalment poca memòria, cosa que restringeix l'ús de recursos perquè en general estan lluny de ser infinits.

El llenguatge de programació que es fa servir és el Java, per tant és important tenir nocions de dit llenguatge i de la programació orientada a objecte.

3.2. AndroidManifest.xml

Hi ha un arxiu xml anomenat `AndroidManifest.xml` on s'ha de declarar els permisos que té l'aplicació, per exemple, si es permet accedir a Internet o vibració entre molts altres. També és on s'especifica la versió mínima permesa per l'aplicació i en la que s'està compilant. Per exemple, si es fixa la versió mínima a la API 10, no serà possible instal·lar l'aplicació en dispositius que tinguin una API inferior a aquesta. Aquesta restricció s'explica per la contínua evolució i la incorporació de noves funcions que no és possible fer servir en antigues versions (tot i que es pot fer servir la llibreria de compatibilitat com veurem més endavant).

Aquest fitxer també és el lloc on s'ha de declarar totes les *activities*, qualsevol que no hagi estat declarada i es faci servir en el codi causarà una tancada forçada de l'aplicació.

Per exemple, si es vol que la versió mínima sigui la API 10, es compila en API 19 i es requereixen permís per accedir a Internet, ja que es vol descarregar imatges des d'un enllaç s'hauria d'introduir el següent codi:

```
<uses-sdk
    android:minSdkVersion="10"
    android:targetSdkVersion="19" />

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

    ...
```

3.3. Activities

Android està instanciat sobre Linux i disposa d'un fil principal des d'on en general es llencen totes les aplicacions.

Les aplicacions estan compostades per *activities*. Cada *activity* representa, a grans trets, una pantalla.

Per cridar a una *activity* primer s'ha de fer servir una classe específica d'Android anomenada `Intent`, a la que se li especifica el context on estem i el nom de l'*activity* a la que volem anar. A més se li poden passar paràmetres que volem que tingui l'*activity* a la que es crida fent servir el mètode `putExtra`. Una vegada configurat l'`Intent` per començar l'*activity* s'ha de fer servir la funció `startActivity(intent)`.

Els "paràmetres extra" que se li passen a l'`Intent` s'han de recollir a l'*activity* que els rep amb la funció `getIntent().getExtras()`.

Un exemple per passar a una *activity* de nom `Activity2` és el següent codi:

```
Intent intent = new Intent(getApplicationContext(), Activity2.class);  
Intent.putExtra("ExtraString", "HelloWorld!");  
startActivity(intent);
```

A l'Activity2 es recull l'extra de la següent manera:

```
String extraString;  
extraString = getIntent().getExtras().getString("ExtraString");
```

Cada *activity* ha d'heredar de la classe `Activity` d'Android per poder sobre escriure els seus mètodes i controlar les accions que es poden fer durant el seu cicle de vida.

El cicle de vida d'una *activity* es pot veure a la Figura 7, extreta de [6]:

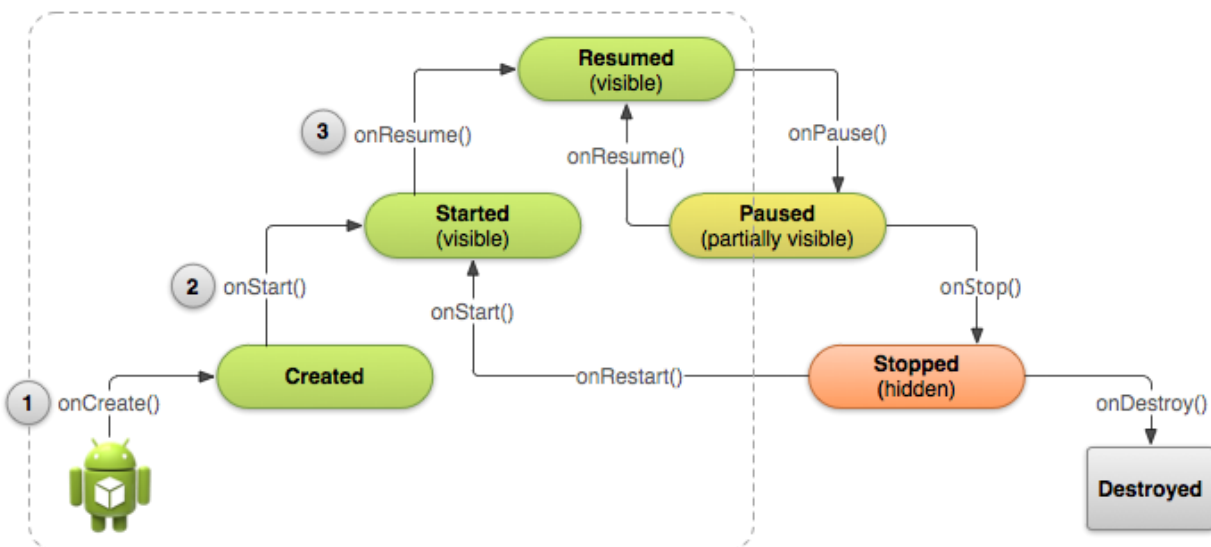


Figura 7: Cicle de vida d'una activity

Cal recordar que s'ha de declarar totes les *activities* a l'arxiu `AndroidManifest.xml`.

3.4. Fragments

Un *fragment* es pot considerar com una *activity* dins d'una altra *activity*, amb el seu propi cicle de vida.

Per fer servir un *fragment* s'ha de declarar un espai on carregar-lo dins de l'*activity* que faci de "contenedor" i d'aquesta manera es poden carregar i intercanviar diferents *fragments*.

3.5. Layouts

Els *layouts* són, a grans trets, la vista de cada *activity*. Són arxius xml on es declaren els elements que hi haurà a la pantalla. Una vegada programat l'arxiu, s'associa generalment a una *activity* o a un *fragment*.

Android proporciona un *plug-in* per a l'eclipse que permet visualitzar i modificar gràficament els *layouts*, però no ofereix un bon rendiment i és preferible escriure directament en codi xml.

3.6. Action Bar

La majoria d'aplicacions disposen d'una barra superior on, en general, hi apareix un títol, la icona de l'aplicació i normalment altres icones amb diferents funcions. Aquesta barra s'anomena *Action Bar* i va ser introduïda amb la API 11. A partir d'aquest moment el seu ús és molt habitual en les aplicacions.

Si marquem que la API mínima suportada és 11 o superior, automàticament l'aplicació hi afegirà una *Action Bar*. Es pot controlar el contingut d'aquesta amb l'arxiu que hi ha a la subcarpeta `res/menu` i carregant-lo al mètode de l'*activity* `onOptionsItemSelected`. També es pot modificar el comportament dels botons amb els mètodes de la classe `ActionBar`. Des d'una *activity* es pot obtenir la seva *Action Bar* amb el mètode `getActionBar()`.

Altrament, si volem que la API mínima sigui inferior o igual a 10, no apareix l'*Action Bar* automàticament, però es pot carregar la llibreria de suport `v7 appcompat` i fer servir la classe `ActionBarActivity` en comptes de `Activity` i la funció `getSupportActionBar()` en comptes de `getActionBar()`. A més al fitxer de la subcarpeta `res/menu` hi hem d'afegir la línia `xmlns:android="http://schemas.android.com/apk/res-auto"` i cridar el paràmetre de les icones (de l'objecte `item`) `showAsAction` de la següent manera: `support:showAsAction="always"` en comptes de `android:showAsAction="always"`.

3.7. Pop-ups

És important, sobretot en el context que ens trobem per la naturalitat del projecte, mantenir informat a l'usuari per a que sàpiga què fer en tot moment i que les accions li semblin intuïtives. En cas que l'usuari realitzi una acció i no tingui conseqüència (per exemple, quan es vol carregar una sèrie i no hi ha carregat un estudi) s'ha de mostrar informació per pantalla.

3.7.1. Toasts

Una manera senzilla que proporciona Android per mostrar missatges per pantalla són els anomenats *Toasts*, que no és més que text sobreposat a un fons fosc que apareix a la pantalla durant uns segons, com es pot veure a la Figura 8 extreta de [6].



Figura 8: Toasts

Per mostrar un *Toast* s'ha de fer servir la classe `Toast` que proporciona Android. Per exemple, per fer sortir per pantalla el missatge "Try again" s'ha d'escriure `Toast.makeText(context, "Try again", Toast.LENGTH_SHORT).show();` on `context` és de classe `Context` i indica a l'entorn on es mostrarà el missatge, per exemple si el codi és part d'una `Activity` només cal posar-hi `this`. `Toast.LENGTH_SHORT` és una constant de la classe `Toast` que indica el temps que s'estarà el missatge a pantalla, també hi ha, entre altres, `Toast.LENGTH_LONG` que trigarà més segons a desaparèixer de la pantalla.

3.7.2. Dialogs

Una altra manera de mostrar missatges a l'usuari és mitjançant una finestra emergent o *pop-up* fent servir la classe `Dialog`.

És important evitar fer servir la classe `Dialog` directament, millor utilitzar una de les seves classes derivades com, per exemple, `AlertDialog`, que és la més convenient per a l'ús que li volem donar a la aplicació.

`AlertDialog` permet mostrar una finestra emergent amb un títol, fins a tres botons i una llista d'elements o un *layout* personalitzat.

Per a controlar el funcionament dels botons i els elements que puguin aparèixer al *pop-up* s'ha de fer servir el `DialogFragment`. Com hom es pot imaginar, és una

classe que hereta de la classe `DialogFragment`, per tant es pot reutilitzar i cridar tantes vegades com es vulgui al codi, igual que es faria amb un `Fragment` usual.

A l'apartat corresponent a les finestres emergents de [6] hi trobem el següent exemple de com instanciar i cridar un element d'aquest tipus:

```
public class FireMissilesDialogFragment extends DialogFragment {
    @Override
    public Dialog onCreateDialog(Bundle savedInstanceState) {
        // Utilitzar la classe Builder per la construcció del dialog
        AlertDialog.Builder builder =
            new AlertDialog.Builder(getActivity());
        builder.setMessage(R.string.dialog_fire_missiles)
            .setPositiveButton(R.string.fire,
                new DialogInterface.OnClickListener() {
                    public void onClick(DialogInterface dialog,
                        int id) {
                        // LLANÇAR MÍSSILS!
                    }
                })
            .setNegativeButton(R.string.cancel,
                new DialogInterface.OnClickListener() {
                    public void onClick(DialogInterface dialog,
                        int id) {
                        // L'usuari cancel·la el dialog
                    }
                });
        // Crear l'objecte AlertDialog i retornar-lo
        return builder.create();
    }
}
```

Una vegada implementada la classe i creada una instància seva, per mostrar la finestra emergent només cal cridar a `show()` des de l'objecte. En aquest exemple es mostraria un missatge com el de la Figura 9, extreta de [6]:



Figura 9: AlertDialog

3.8. Adapters

Per a mostrar una llista d'elements personalitzats s'ha de crear un element anomenat `ListView` al *layout* on es mostrarà. Com el seu nom indica, el `ListView` és una llista i per omplir-la d'un nombre indeterminat d'elements personalitzats

s'ha de fer servir un adaptador (classe `Adapter`). En aquest cas, farem servir `ArrayAdapter<T>` on `T` és un patró, d'aquesta manera podem crear una classe objecte nova que ens interessi i omplir la llista amb aquest objecte. Per exemple, si volem una llista que cada element tingui el nom d'un medicament, el temps que se l'ha de prendre el pacient, la freqüència i la dosi primer hem de tenir un fitxer layout `medicine_list_item.xml` que contingui l'estructura de cada element de la llista.

També s'ha de crear una classe semblant a la següent:

```
public class Medicine {  
  
    String name;  
  
    String duration;  
  
    String frequency;  
  
    String dose;  
  
    ...  
  
    public Medicine(String name, String duration, String frequency,  
        String dose) { // Constructor  
  
        this.name = name;  
        this.duration = duration;  
        this.frequency = frequency;  
        this.dose = dose;  
  
        ...  
  
    }  
}
```

Després s'ha de crear una llista d'objectes `Medicine`, és a dir, una `ArrayList<Medicine>` `medicine` i omplir-la dels elements amb els que volem omplir la llista.

A part, s'ha de crear una classe *adapter* com la següent tal i com hem comentat a l'inici de la secció:

```
private class Medicine extends ArrayAdapter<Medicine> {  
  
    private int name;  
  
    private ArrayList<Medicine> medicineList;  
  
    ...  
  
    public MedicineAdapter(Context context, int layout,  
        ArrayList<Medicine> medicineList) { // Constructor
```

```

    super(context, layout, medicineList);
    this.layout = layout;
    this.medicineList = medicineList;

    ...
}

@Override
public View getView (int position, View convertView,
    ViewGroup parent) {

    LayoutInflater inflater = getActivity().getLayoutInflater();
    convertView = inflater.inflate(layout, parent, false);

    ...

    nameTV.setText(medicineList.get(position).name);
    durationTV.setText(medicineList.get(position).duration);
    frequencyTV.setText(medicineList.get(position).frequency);
    doseTV.setText(medicineList.get(position).dose);

    ...

    return convertView;
}
}

```

Finalment, només cal crear una instància d'aquest *adapter* i assignar-li a la llista:

```

MedicineAdapter mAdapter = new MedicineAdapter(parent,
    R.layout.medicine_list_item, medicineList);

listView.setAdapter(mAdapter);

```

3.9. AsyncTask

És molt important per a gestionar la descàrrega d'imatges fer servir l'anomenada `AsyncTask`. Per defecte totes les accions del codi es realitzen sobre el fil d'execució principal d'Android. El problema apareix quan una tasca és més lenta i manté bloquejada la pantalla durant més de tres segons, ja que passat aquest temps, si ens trobem en un dispositiu amb una API inferior a la 10, es produeix una aturada forçada de l'aplicació (ANR, és a dir *Android Not Responding*), o bé, si l'API és superior, la tasca acaba sense donar cap resultat.

Per solucionar aquest problema és convenient fer servir una classe `AsyncTask` on realitzi l'acció que s'ha de tractar a part dins del mètode `doInBackground` i les tasques que ha de fer al finalitzar-la a `onPostExecute`.

La classe té com a paràmetres el tipus d'entrada, el tipus de sortida i el tipus de dades que es retorna a `doInBackground`.

Un exemple d'`AsyncTask` que descarrega una imatge és la següent:

```
private class DownloadImageTask extends AsyncTask<String, Void,
    Bitmap> {

    ImageView bmImage;

    public DownloadImageTask(ImageView bmImage) {
        this.bmImage = bmImage;
    }

    @Override
    protected Bitmap doInBackground(String... urls) {
        String urldisplay = urls[0];
        Bitmap bitmap = null;
        try {
            InputStream in = new java.net.URL(urldisplay).openStream();
            bitmap = BitmapFactory.decodeStream(in);
        } catch (Exception e) {
            Log.e("Error", e.getMessage());
            e.printStackTrace();
        }
        return bitmap;
    }

    @Override
    protected void onPostExecute(Bitmap result) {
        bmImage.setImageBitmap(result);
    }
}
```

Una vegada construïda la classe `DownloadImageTask` per descarregar una imatge s'ha de fer la següent crida:

```
new DownloadImageTask(imageView).execute(url);
```

on `imageView` és l'`ImageView` on es vol carregar la imatge i `url` és un `String` que conté la URL on és la imatge que es vol descarregar.

3.10. ViewPager

Una eina important per al projecte és el `ViewPager`, que permet passar “pàgines” al lliscar el dit per la pantalla. Per poder aconseguir aquesta funcionalitat cal un objecte de la classe `ViewPager` i un adaptador de la classe `PagerAdapter`, que s'ha

d'instanciar amb una classe que hereti de `FragmentStatePagerAdapter` amb la que es gestionarà el funcionament de cada “pàgina” del `ViewPager`, que són `Fragments`.

Per exemple, per instanciar un `ViewPager` com el que necessitem al projecte s'hauria de fer com segueix:

```
String[] urImages;      // On es guardarà els enllaços corresponents
                       // a cada imatge del ViewPager
                       ...

ViewPager viewPager = (ViewPager) findViewById(pagerId);
ViewPagerAdapter viewPagerAdapter = new
    ScreenSlidePagerAdapter(getSupportFragmentManager(), urImages,
        urImages.length);
viewPager.setAdapter(viewPagerAdapter);
```

Per altra banda, `ScreenSlidePagerAdapter` ha de ser qui gestiona els `Fragments` que corresponen a les pàgines del `ViewPager`. El que gestiona el del codi anterior podria ser:

```
protected class ScreenSlidePagerAdapter extends
    FragmentStatePagerAdapter {

    private String[] url_images;
    private int n_images;

    public ScreenSlidePagerAdapter(FragmentManager fm,
        String[] url_images, int n_images) {
        super(fm);
        this.url_images = url_images;
        this.n_images = n_images;
    }

    @Override
    public Fragment getItem(int position) {
        return Fragment1Image.create(position, R.id.iv_one,
            url_images);
    } // Fragment1Image és el Fragment que gestiona el
        // comportament de cada pàgina
        // R.id.iv_one és el contingut de la pàgina

    @Override
    public int getCount() {
        return n_images;
    }

}
```

Finalment, es pot accedir i operar sobre les pàgines del `ViewPager` amb els seus mètodes.

Per accedir a una pàgina concreta cal fer servir `setCurrentItem`:

```
viewPager.setCurrentItem(n_image);  
  
// n_image és el número de la pàgina del ViewPager que es  
// visualitzarà després d'aquesta crida
```

Un altre mètode important pel control del paginat és `setOnPageChangeListener` al que si li passem un objecte de la classe `SimpleOnPageChangeListener` amb el mètode `onPageSelected` sobreescrit es podrà controlar el comportament de cada pàgina concreta. Per exemple:

```
viewPager.setOnPageChangeListener(  
    new ViewPager.SimpleOnPageChangeListener(  
  
        @Override  
        public void onPageSelected(int position) {  
            // realitzar accions sobre la pàgina número position  
        }  
  
    )  
);
```

3.11. Traduccions

La manera natural d'afegir traduccions a Android és amb el fitxer `strings.xml` que es troba a la carpeta `res` del projecte. A aquest arxiu s'hi afegeixen els textos que es volen mostrar seguint el següent format:

```
<resources>  
  
    ...  
  
    <string name="hello_world">Hello World!</string>  
  
    ...  
  
</resources>
```

La carpeta `res/values` conté el fitxer en l'idioma que volem que sigui per defecte. Per afegir una traducció cal copiar el mateix fitxer a la carpeta `res/values-idioma` on idioma correspon al codi de la llengua. Per exemple, si volem traduir el fitxer anterior al català, el copiarem a la carpeta `res/values-ca` i el traduirem:

```
<resources>
    ...
    <string name="hello_world">Hola Món!</string>
    ...
</resources>
```

Si volem accedir des del codi al text, des d'una `Activity` podem fer una crida a la funció `getString()`. Així, si volem mostrar per pantalla *Hello World!*, escriurem `getString(R.string.hello_world)`. Si tenim el dispositiu en català apareixerà *Hola Món!*, i si no hem afegit cap més traducció sortirà en anglès ja que la carpeta `res/values` és la carpeta per defecte.

CAPÍTOL 4

APLICACIÓ FINAL

A aquest capítol s'explica el funcionament de l'aplicació que ha resultat després de seguir l'aprenentatge que s'ha explicat als capítols anteriors.

La informació més valuosa que hem pogut extreure ha estat a l'entrevista, ja que degut a les limitacions de recursos i de disponibilitat dels metges com a potencials usuaris finals moltes tècniques no les hem pogut explotar tant com voldríem. Per tant, l'aplicació final ha sorgit essencialment de la teoria apresada a les classes del professor Scott Klemmer [1], de l'entrevista, d'on s'han extret les necessitats bàsiques, i s'ha anat perfilant gràcies al prototip de paper que ha fet més fàcil el procés iteratiu de l'avaluació heurística.

Quan es va començar a escriure el codi disposàvem d'una *tauleta* ASUS Transformer Prime amb Android Jelly Bean (API 16) i es va enfocar l'aplicació a dispositius amb una API superior a la 10, per tant no calia fer servir les llibreries de compatibilitat per a utilitzar la *Action Bar* ni altres eines que van ser introduïdes en versions posteriors a la 10. Però a mig procés ja no vam tenir accés aquesta *tauleta* i vam haver de fer servir una Samsung Galaxy Tab amb Android Gingerbread (API 10), i per tant, haver de refer parts del codi per afegir-hi les llibreries de compatibilitat que permeten aprofitar les funcions més noves.

Les captures de pantalla que es mostren durant el capítol són totes de l'aplicació executant-se a la segona *tauleta*.

4.1. Action Bar i navegabilitat

A l'*Action Bar* hi ha 7 botons en total: a l'esquerra aprofitem el botó *home* per fer canviar el tipus de pantalla.

Per defecte, es pot carregar una imatge a pantalla completa, però com a requeriments també teníem la opció de poder carregar-ne quatre alhora. Amb el botó *home* es pot canviar aquesta modalitat, al pulsar-lo es va a la pantalla partida on es poden carregar 4 imatges alhora. Al tornar-lo a pulsar tornem a la pantalla anterior i així successivament.

Aquest procés s'ha implementat fent servir dues *Activities*, una per a la pantalla de una imatge, i una altra per a la pantalla de quatre imatges alhora. S'ha desactivat l'efecte visual al canviar d'*Activity* amb la línia de codi següent que afecta a l'*Intent*:

```
intent.addFlags(Intent.FLAG_ACTIVITY_NO_ANIMATION);  
...  
startActivity(intent);  
finish();
```

Com es pot veure, es finalitza l'Activity des de la que es ve perquè d'aquesta manera al prémer el botó *back* del dispositiu no anirà passant per totes les tasques que hem anat passant. En canvi el comportament del botó *back* és el de sortir de l'aplicació (excepte a la secció d'informació del pacient, com veurem més endavant). Es mostra el missatge que veiem a la Figura 10.

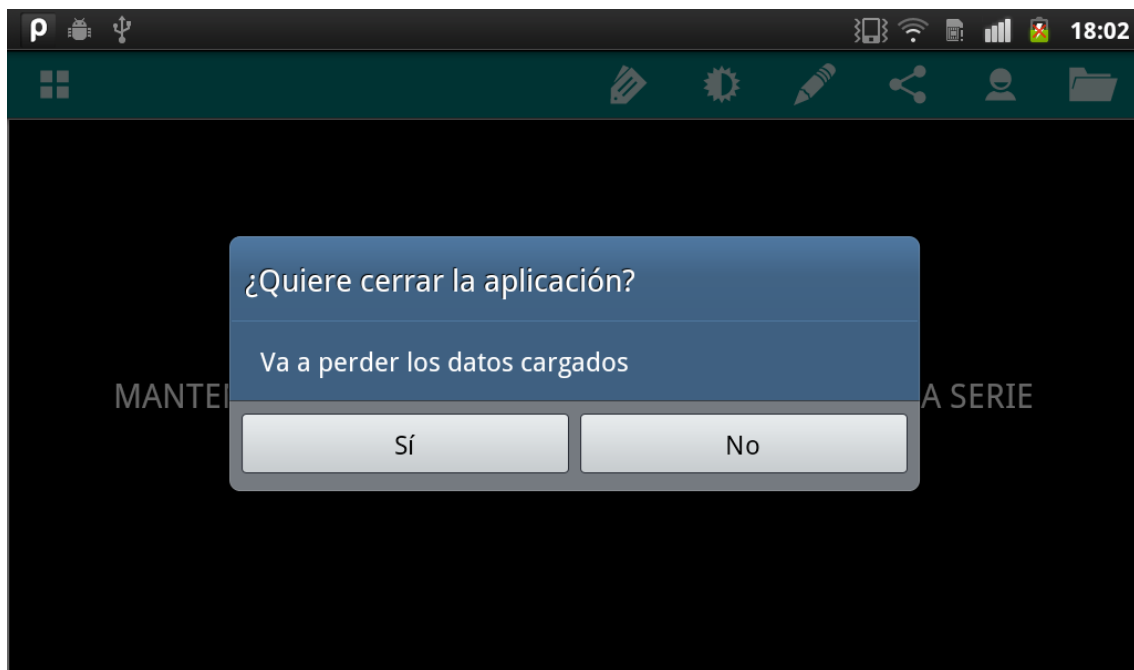


Figura 10: Tancar aplicació

Com es tracta de transicions entre *Activities* passem i rebem les dades de les imatges que hi ha carregades, el color de la imatge i altres als *Intents* i als mètodes *onRestoreInstanceState()* i *onSaveInstanceState()*, que és on es recupera i guarda la informació a l'iniciar-se. Això també permet mantenir l'estat al rotar la pantalla.

Pel que fa a les següents icones de l'Action Bar, les podem veure a la Figura 11.



Figura 11: Action Bar

La icona de més a l'esquerra permet canviar la visualització d'una imatge o quatre
alhora, com podem veure a la Figura 12.



Figura 12: Visualitzar una o quatre imatges

La icona de la de la carpeta ens permet validar-nos i carregar un estudi.

La icona de la seva l'esquerra, la de la persona, ens porta a la secció d'informació del pacient.

La que segueix permet compartir informació.

La icona del llapis permet afegir comentaris a la imatge que hi ha carregada.

La icona de l'esquerra, la del Sol, permet canviar el color de les imatges.

La última icona de les etiquetes ens permet veure la informació sobre les imatges per pantalla i ocultar-la.

Si intentem accedir a alguna de les opcions (excepte la de la carpeta i la del botó *home*) sense haver carregat un estudi es mostrarà un `Toast` amb el missatge de la Figura 13.

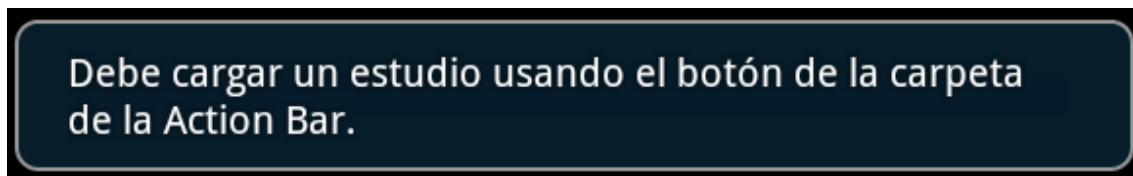


Figura 13:Toast carregar estudi

4.2. Carregar estudi

Amb el botó de la carpeta de l'*Action Bar* podem carregar un estudi de fotos.

Un estudi de fotos, tal com vam extreure de l'entrevista amb els representants de l'UDIAT del Parc Taulí de Sabadell, conté diverses sèries, i cada sèrie és un conjunt d'imatges. El procés de visualització de les imatges de cada sèrie l'explicarem a la següent secció.

Al pulsar el botó ens demanarà validar-nos per poder accedir a l'estudi. Cada vegada que realitzem aquesta acció ens demanarà validar-nos, això és important degut a la confidencialitat de les dades, per tant no es guarda la sessió.

Com veiem a la Figura 14, hem creat un `DialogAlert` amb un *layout* personalitzat que conté dos `EditText` on al segon s'ha especificat que és una contrasenya, i per això es veu el text ocult. Els botons d'acceptar (OK) i el de cancel·lar apareixen en aquest ordre per decisió interna del sistema. En versions més modernes d'Android, han pres la decisió de mostrar-los en ordre invers.

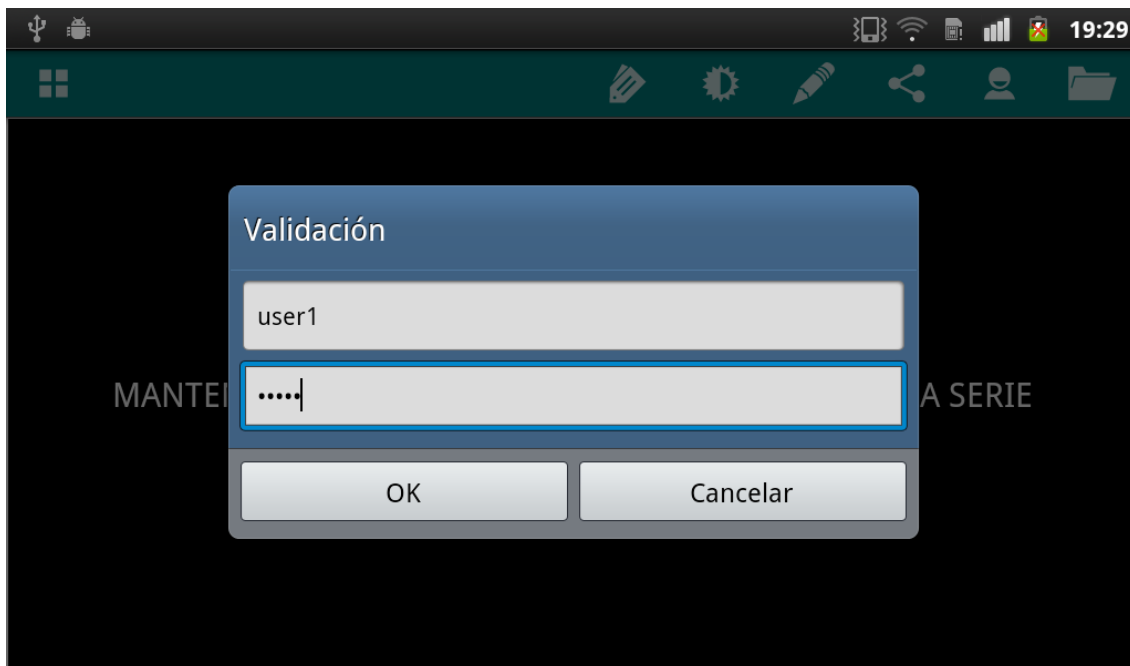


Figura 14:Validació

Si l'usuari o contrasenya no són correctes o s'ha deixat en blanc algun camp s'informa a l'usuari per amb un `Toast` amb un text adient, com veiem a les imatges de la Figura 15.

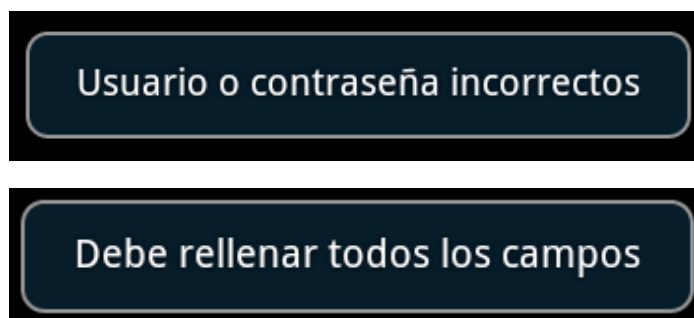


Figura 15:Validació errònia

Si la validació ha estat correcta també es mostra un `Toast` informant, com veiem a la Figura 16.



Figura 16:Validació satisfactòria

Les parelles d'usuaris i contrasenyes vàlides per a accedir són "user1" amb contrasenya "pass1", "user2" amb "pass2" i "user3" amb "pass3".

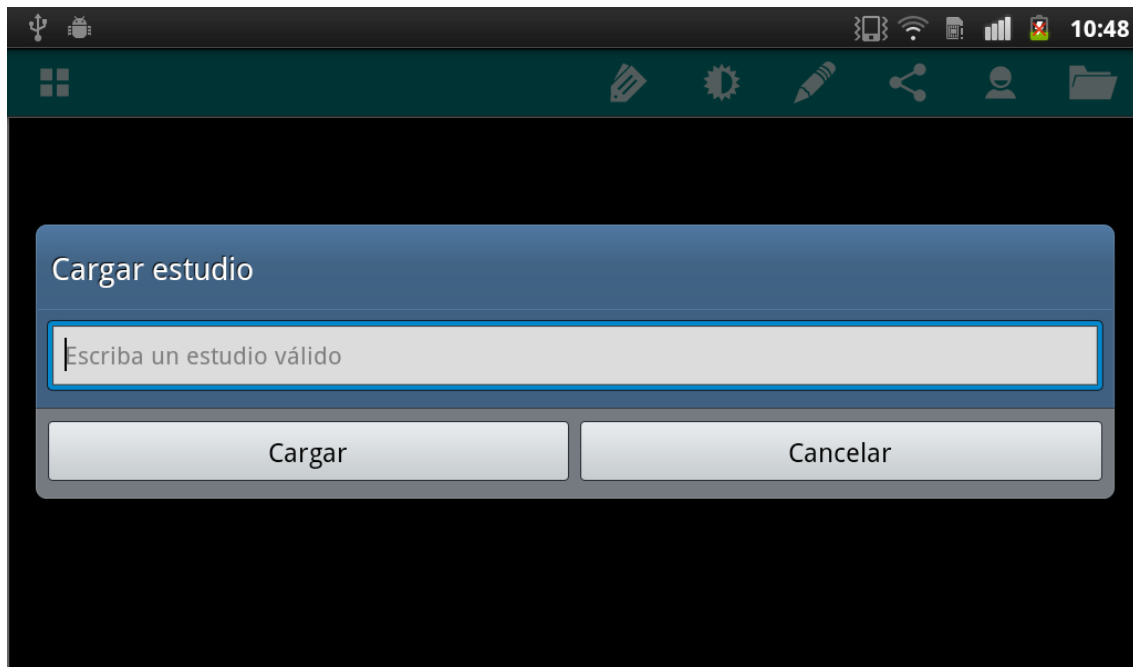


Figura 17:Carregar estudi

Una vegada superada la validació, es mostra una altra finestra emergent per introduir el nom d'un estudi (Figura 17).

Si l'estudi introduït és correcte es mostra un Toast informant (Figura 18).



Figura 18:Estudi no trobat

Si s'ha introduït correctament també es mostra un Toast amb informació de com carregar les sèries d'aquest estudi (Figura 19).

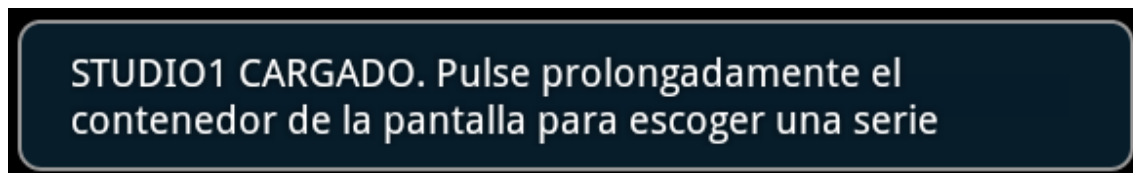


Figura 19:Estudi carregat

L'estudi al que es pot accedir que hi ha carregat per a realitzar proves és "STUDIO1".

4.3. Carregar sèrie

Una vegada carregat un estudi, al mantenir la pantalla polsada tal com s'indica amb el text que hi apareix un *pop-up* amb una llista de les sèries que conté l'estudi i que es podran carregar, com veiem a la Figura 20

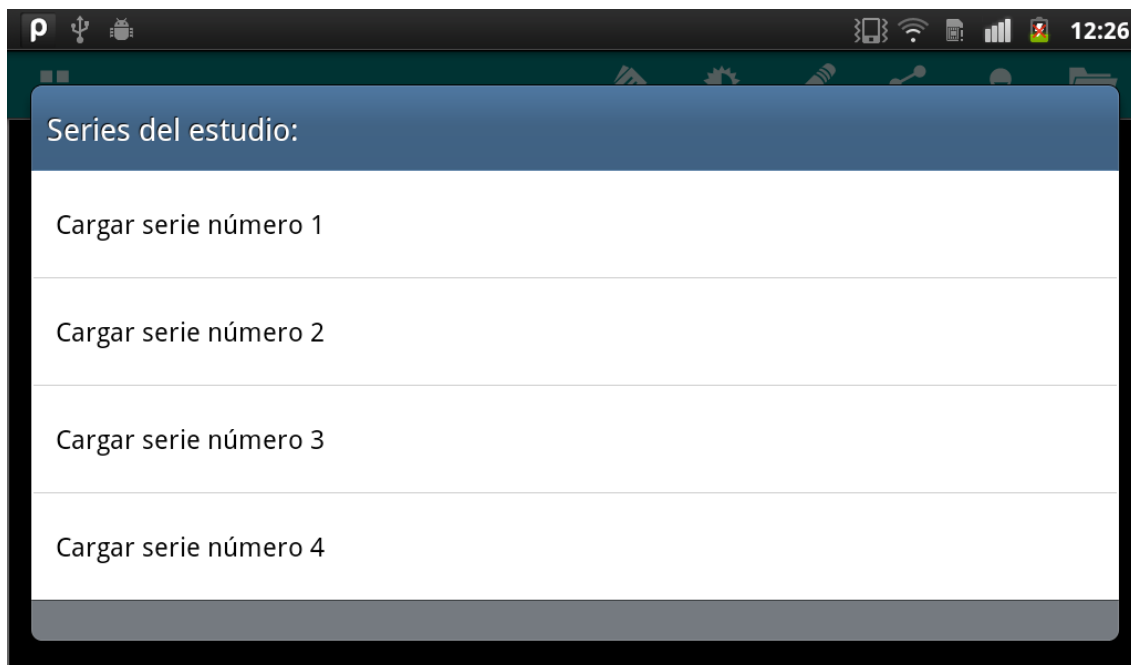


Figura 20:Sèries

La idea principal era realitzar la càrrega de les imatges mitjançant el protocol JPIP per acostar-nos al màxim a les necessitats mèdiques. Però degut a la poca disponibilitat de recursos finalment la càrrega de les imatges es fa des d'unes URLs que contenen unes imatges de prova. En aquest cas tenim disponibles únicament una sèrie i un estudi però a la llista de sèries hi apareixen 4 sèries perquè en realitat són la mateixa, ja que la finalitat és que es vegi com queda la interfície d'usuari que és l'objectiu principal del projecte.

Una vegada carregat la sèrie, automàticament es mostrarà per pantalla la seva primera imatge. Mentre s'està carregant es veu per pantalla un cercle en moviment anomenat *ProgressBar*, com podem veure a la Figura 21. Quan la imatge ja està carregada es veu com a la Figura 22.

Recordem que per poder realitzar aquesta acció és necessari indicar al fitxer *AndroidManifest.xml* que es requereixen permisos d'Internet perquè es realitza una descàrrega directa d'una imatge que està a la xarxa a partir d'un enllaç. Aquesta és la línia de codi que permet accedir a Internet des de l'aplicació:

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

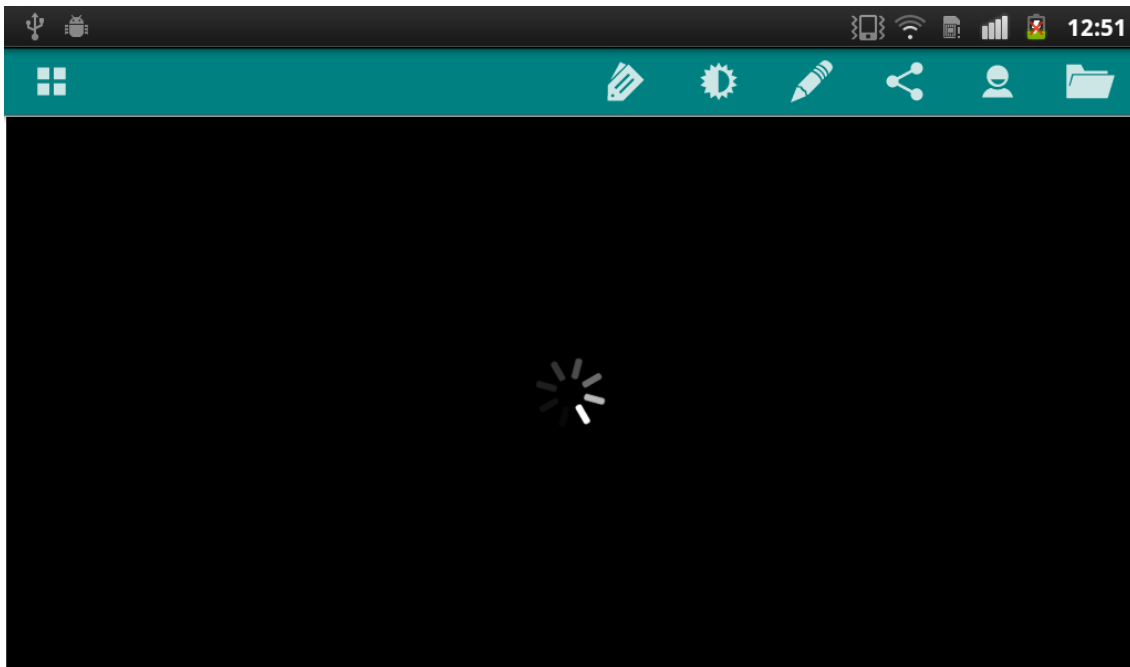


Figura 21:Progress Bar

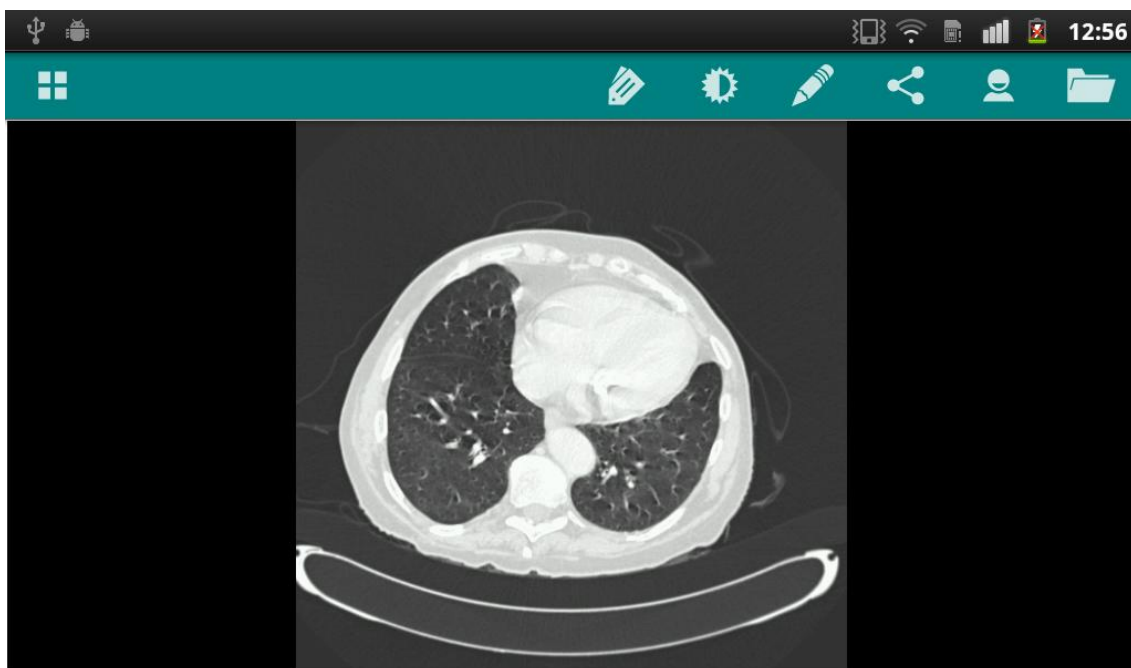


Figura 22:Imatge carregada

A la pantalla de 4 imatges alhora es veu com a la Figura 23. Com podem veure a la imatge, podem seleccionar-ne una de les quatre amb un clic simple i aquesta es ressaltarà canviant la seva vora a color groc. Les accions que es vulguin realitzar, com per exemple canviar el color, sempre es faran sobre la imatge seleccionada.

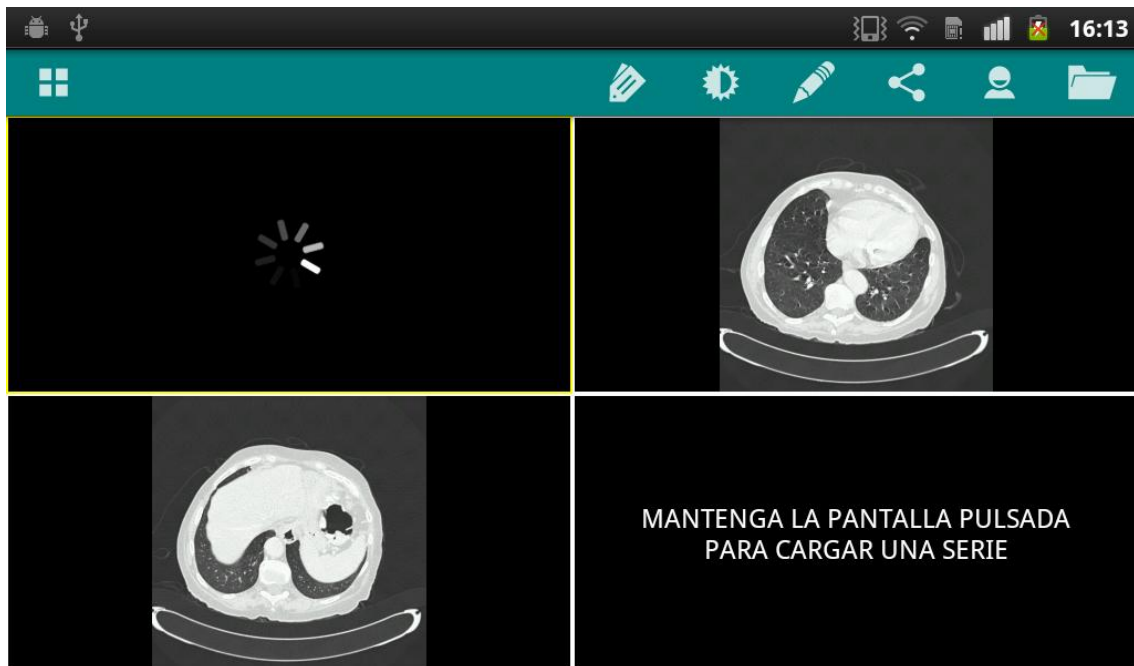


Figura 23: Opció quatre imatges

Per passar a la següent imatge d'una sèrie només cal fer lliscar el dit per la pantalla de dreta a esquerra. I per passar a l'anterior s'ha de fer a la inversa, com es veu a la Figura 24.

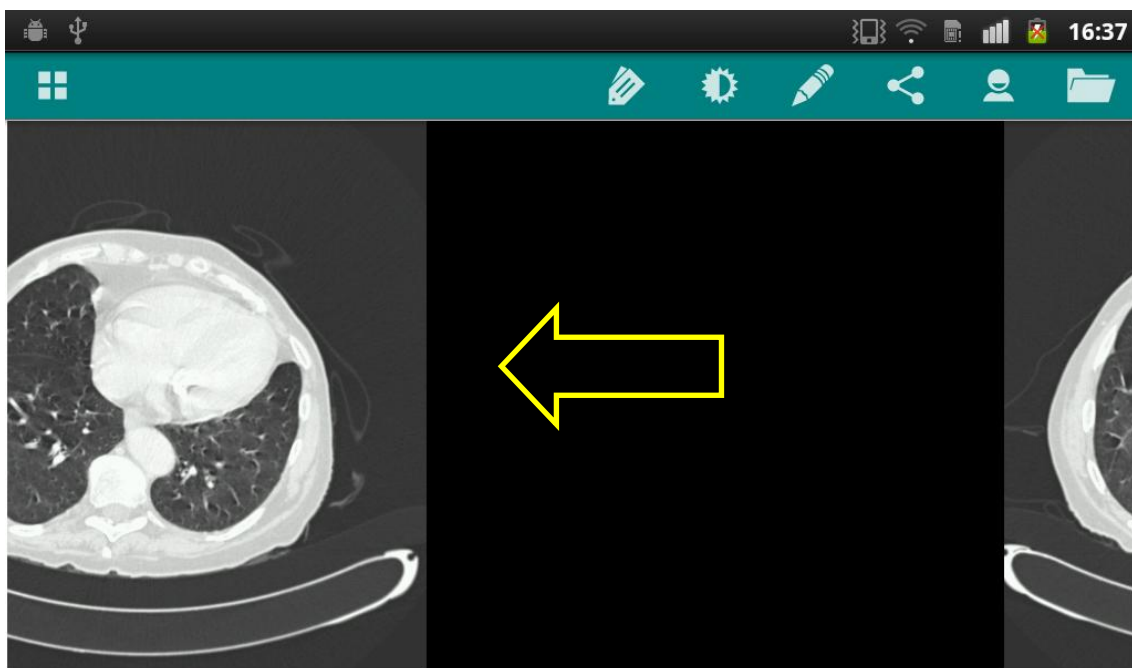


Figura 24: Passar imatges

Per implementar aquest efecte de paginat s'ha fet servir un `ViewPager` en la modalitat d'una sola imatge i quatre `ViewPager` en la modalitat de quatre imatges simultànies.

Mitjançant una llibreria de tercers [8], s'ha afegit un efecte de zoom. Si es fa un doble toc a una imatge, fa efecte de zoom al màxim sobre el punt on s'ha efectuat l'acció. Si posem dos dits sobre la pantalla i els apropem entre si la imatge es redueix i si els allunyem s'amplia, com veiem a la Figura 25.

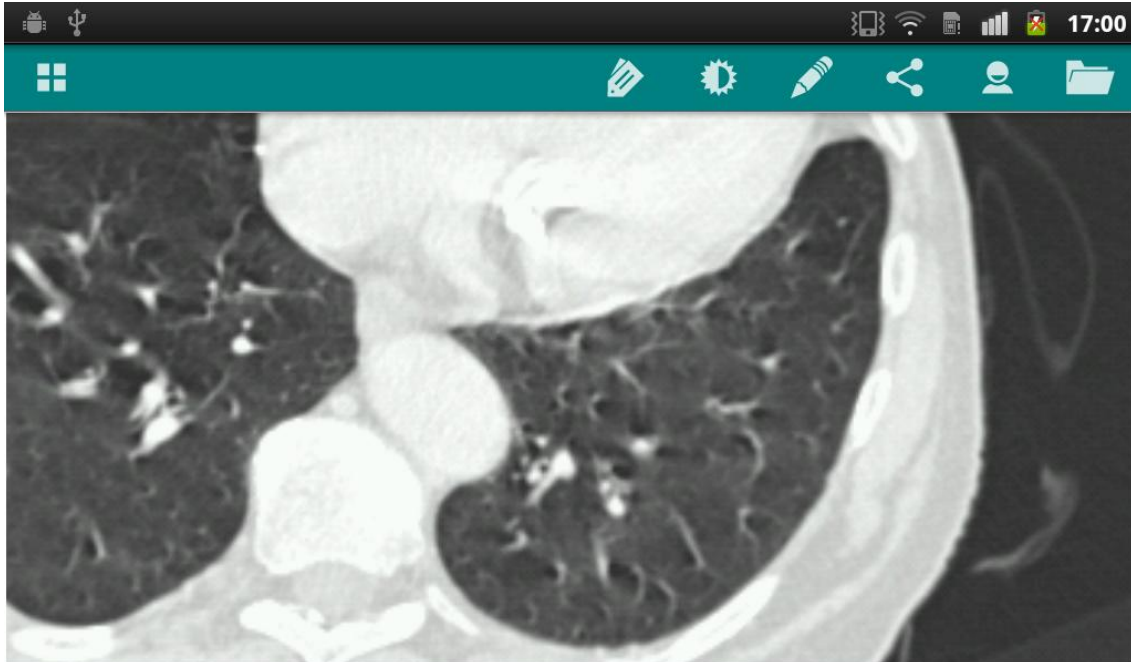
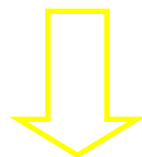


Figura 25: Zoom

A més, una vegada efectuat l'efecte de zoom si desplaçem el dit per la pantalla ens mourem per la imatge.

Si estem en la visualització de quatre imatges simultànies i fem doble clic en una d'elles es mostrarà la imatge a pantalla completa (Figura 26).



Doble toc

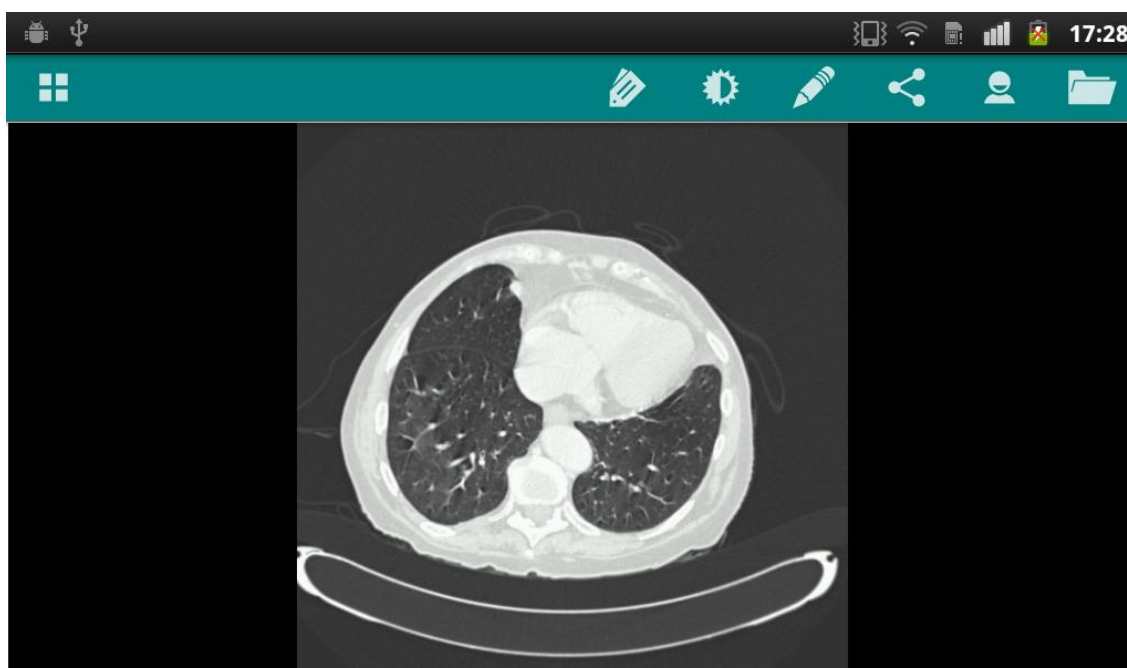


Figura 26: Zoom doble toc

4.4. Mostrar informació de les imatges

Una vegada carregada una imatge es pot mostrar o ocultar informació polsant el botó de les etiquetes.

Seguint les indicacions que vam extreure a l'entrevista sobre la ubicació de la informació de les imatges a la pantalla, hem afegit el número de imatge, la sèrie, el nom del pacient, l'hora i data de la obtenció de la imatge mèdica, els kV i mA en que es va prendre i la part del cos a la que correspon. La ubicació de la informació es pot veure com queda a la següent imatge:

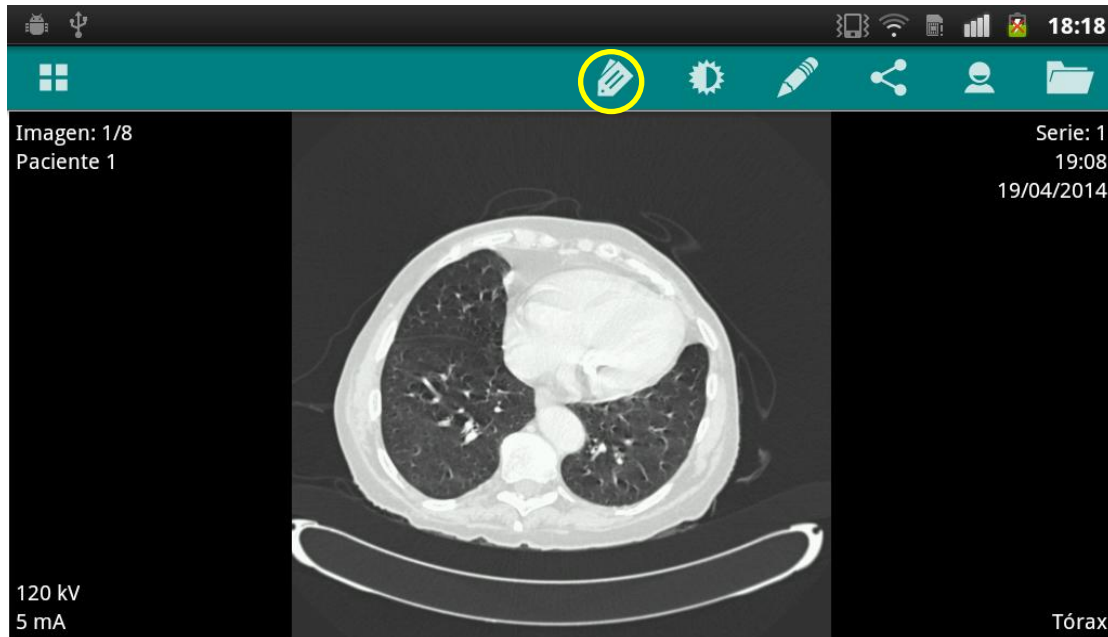


Figura 27: Informació

A la pantalla partida, el funcionament és equivalent, però, com és lògic, es mostrarà la informació de les imatges que estiguin carregades i de cap més (Figura 28).

4.5. Canviar color

Amb la icona del sol tenim una sèrie de colors disponibles predeterminats (usuals en aquest context) com veiem a la Figura 29.

Si triem, per exemple *Green temperature*, a la pantalla partida obtenim la imatge seleccionada en aquest color (Figura 30).



Figura 28: Informació pantalla partida

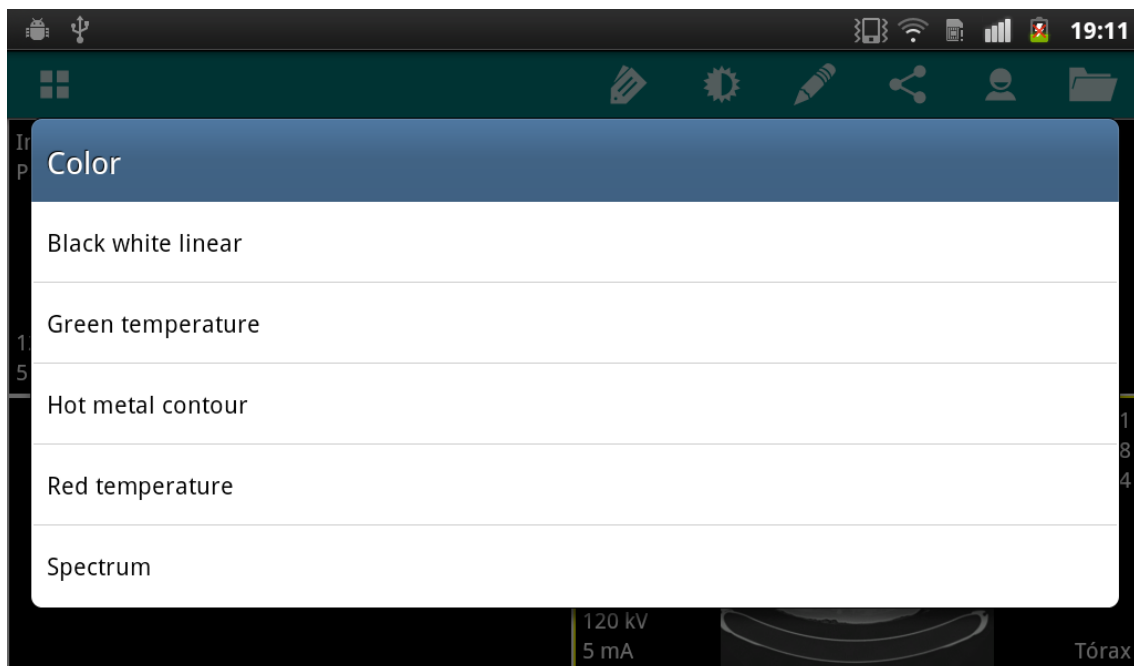


Figura 29: Tria de colors



Figura 30: Green temperature

La resta de colors disponibles són els següents, vistos en pantalla completa:

Black white linear: blanc i negre que veiem a totes les captures de pantalla fins ara.

Hot metal contour (Figura 31).

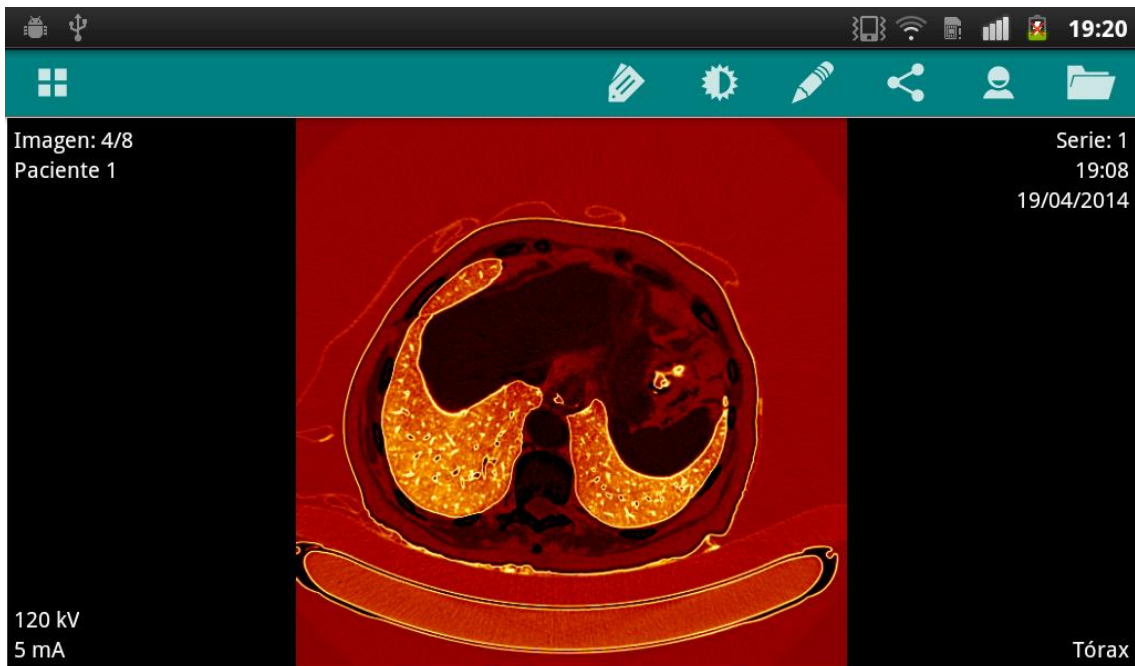


Figura 31: Hot metal contour

Red temperature (Figura 32).

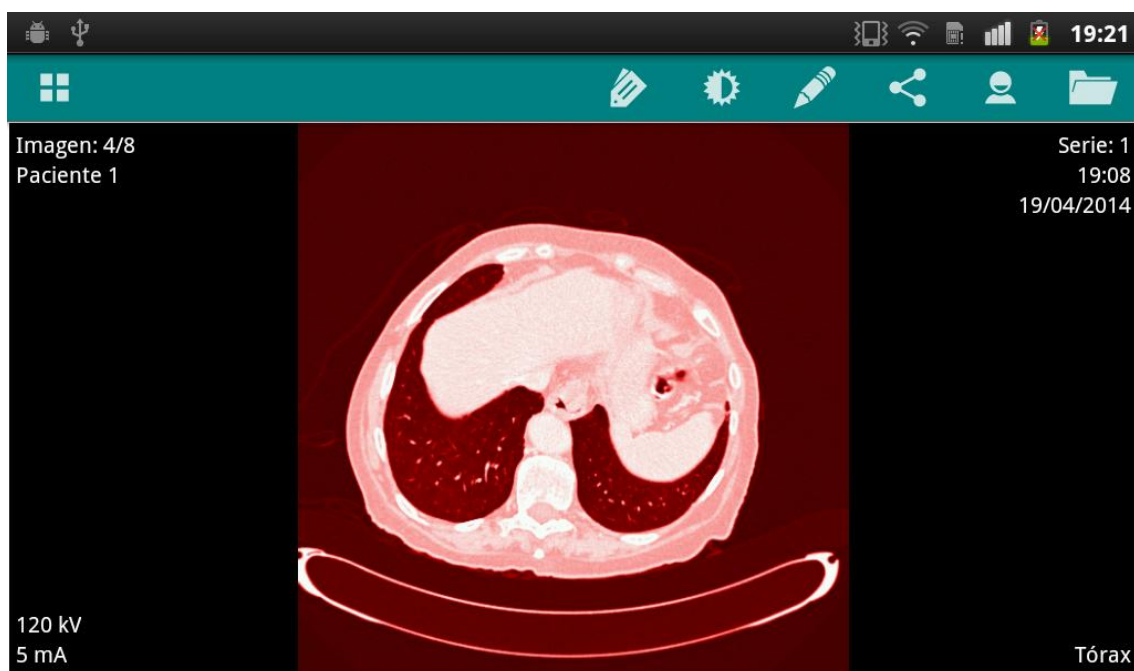


Figura 32: Red temperature

Spectrum (Figura 33).

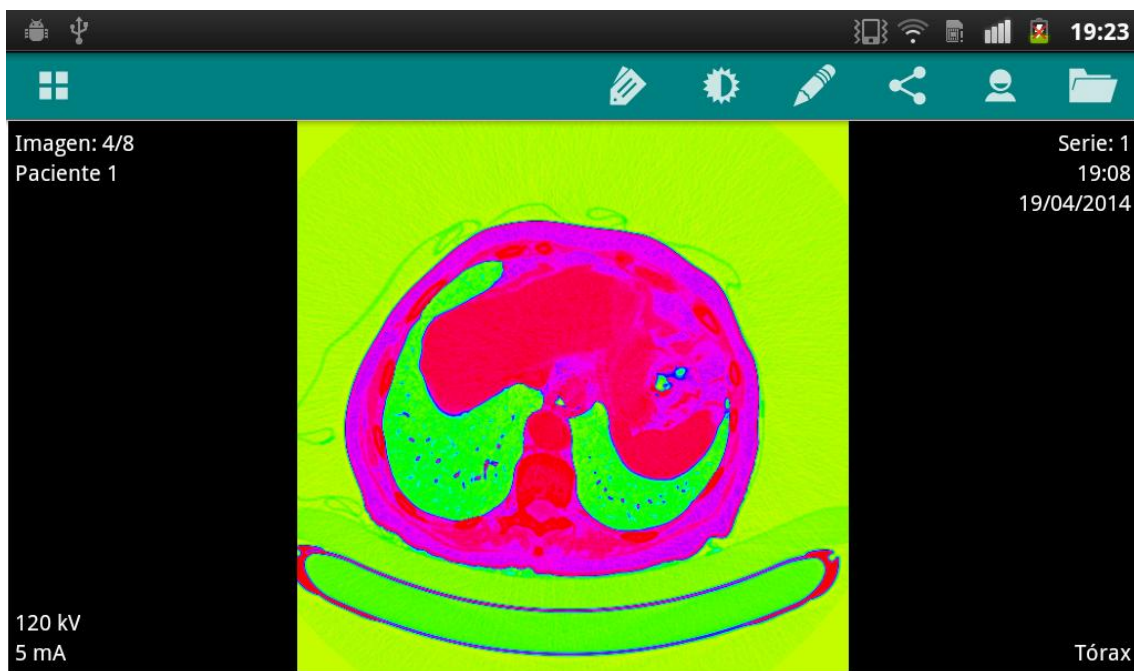


Figura 33: Spectrum

4.6. Comentaris

Una opció per cobrir amb les necessitats per marcar una imatge amb un comentari amb els recursos que tenim és afegir una opció a l'Action Bar (en aquest cas, la del llapis) que permet afegir comentaris a les imatges.

Quan es prem la icona del llapis i hi ha alguna imatge carregada, apareix una finestra emergent com la de la Figura 34.

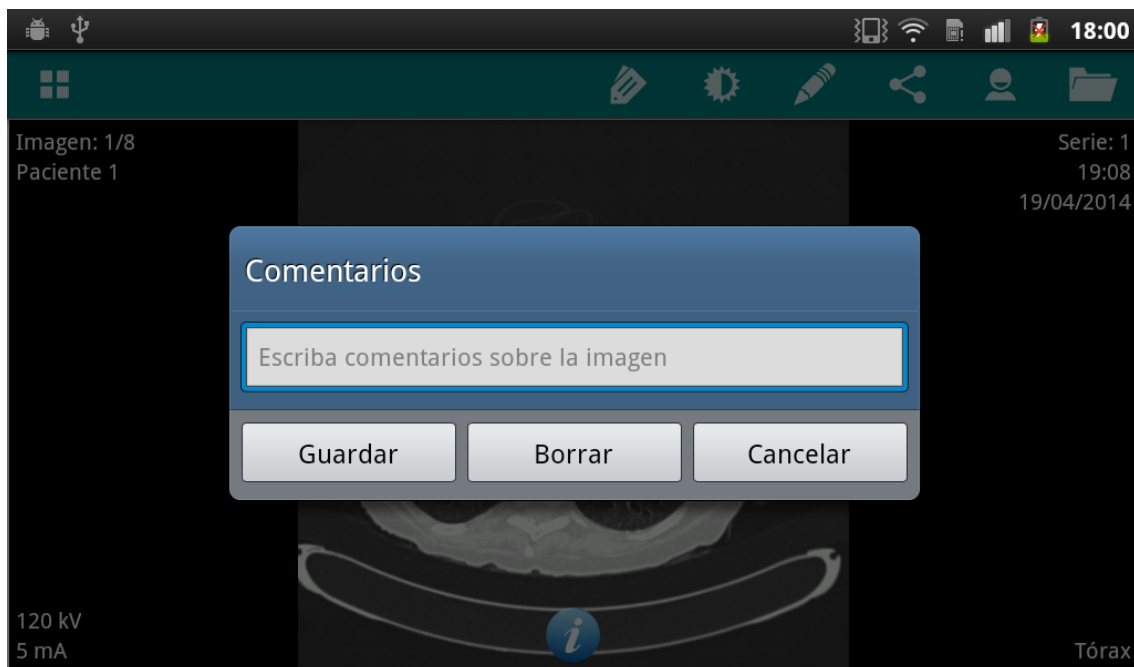


Figura 34: Comentarios

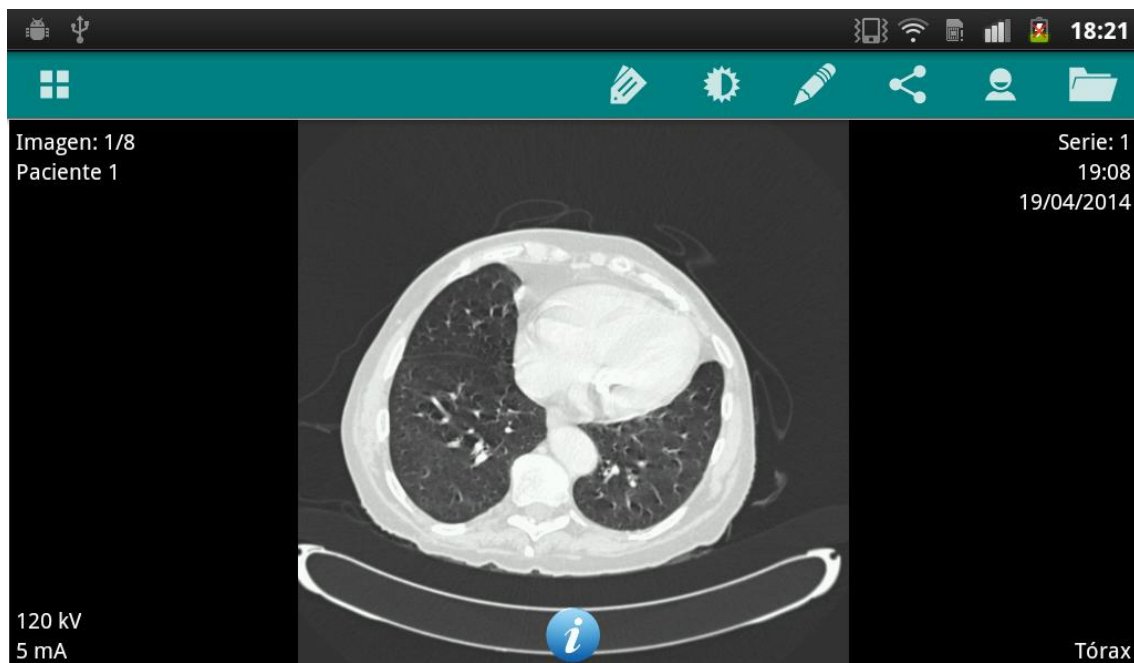


Figura 35: Icona d'informació

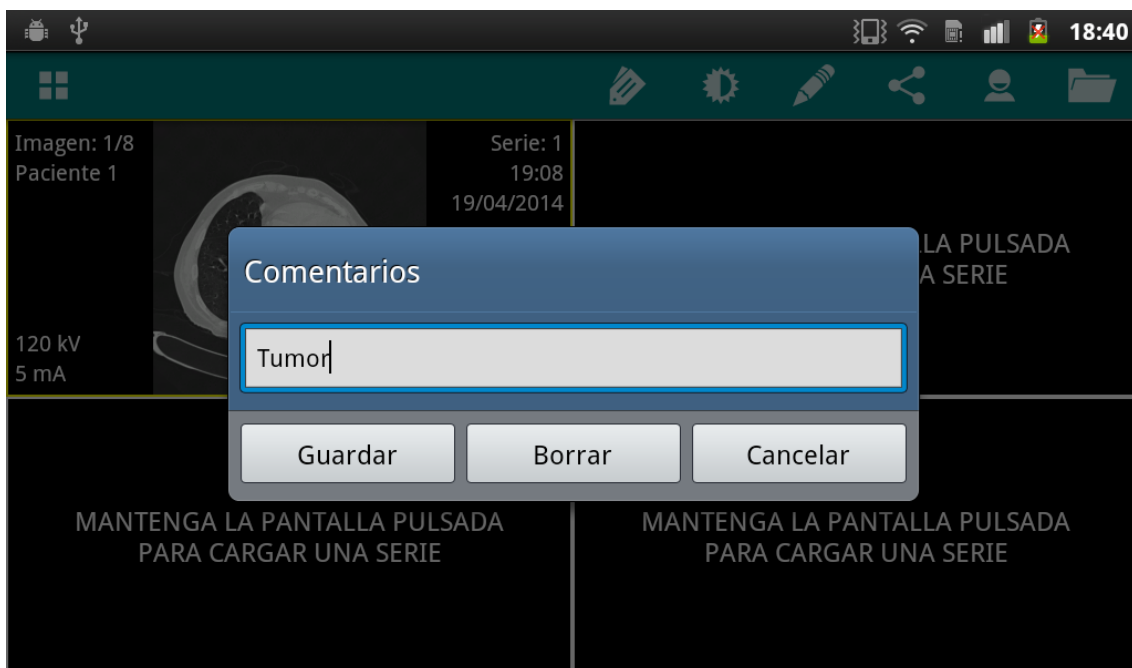
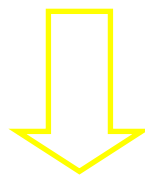
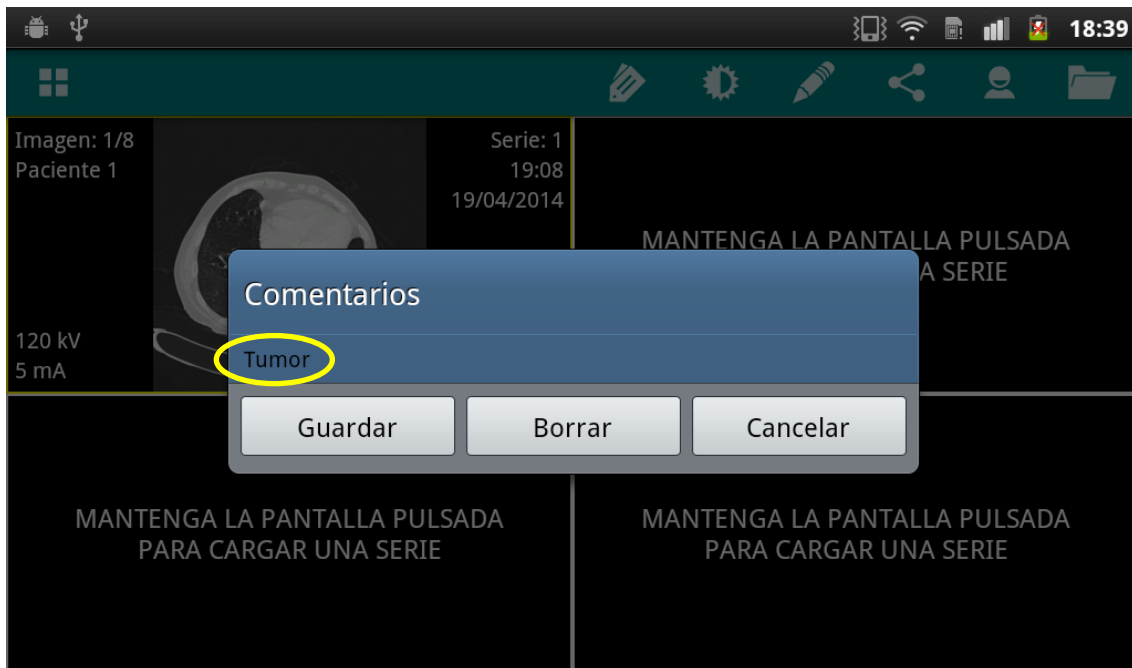


Figura 36: Edició de comentaris

Dins del requadre de text es pot afegir un comentari que es guardarà per a tots els usuaris que accedeixin a l'aplicació.

Per fer-ho, hem creat un fitxer "públic" amb una taula *hash* on hi anem escrivint els comentaris. La clau de cada comentari per a aquesta taula és la concatenació del nom d'estudi, la sèrie i el número d'imatge.

Si escrivim un comentari o algú altre l'escriu, a la part central inferior de la imatge hi sortirà una icona d'informació (Figura 35), que es podrà tocar i accedir a la pantalla que hem vist anteriorment.

Cal afegir que aquesta icona només serà visible si s'ha activat la informació per a les imatges, és a dir, si ha fet *clic* al botó de les etiquetes.

Per acabar, si es vol modificar o esborrar el comentari de la imatge, l'acció està permesa. En el primer cas, només cal prémer el text i automàticament es convertirà en una caixa d'edició amb el comentari anterior per defecte (Figura 36), d'aquesta manera és possible modificar-lo i guardar-lo. En canvi, si el que es vol és esborrar un comentari, només cal tocar el botó d'esborrar que apareix a la finestra emergent quan es prem la icona d'informació. Al fer això apareix un nou *pop-up* preguntant si realment volem eliminar el comentari definitivament amb dues opcions, esborrar, o cancel·lar (Figura 37).

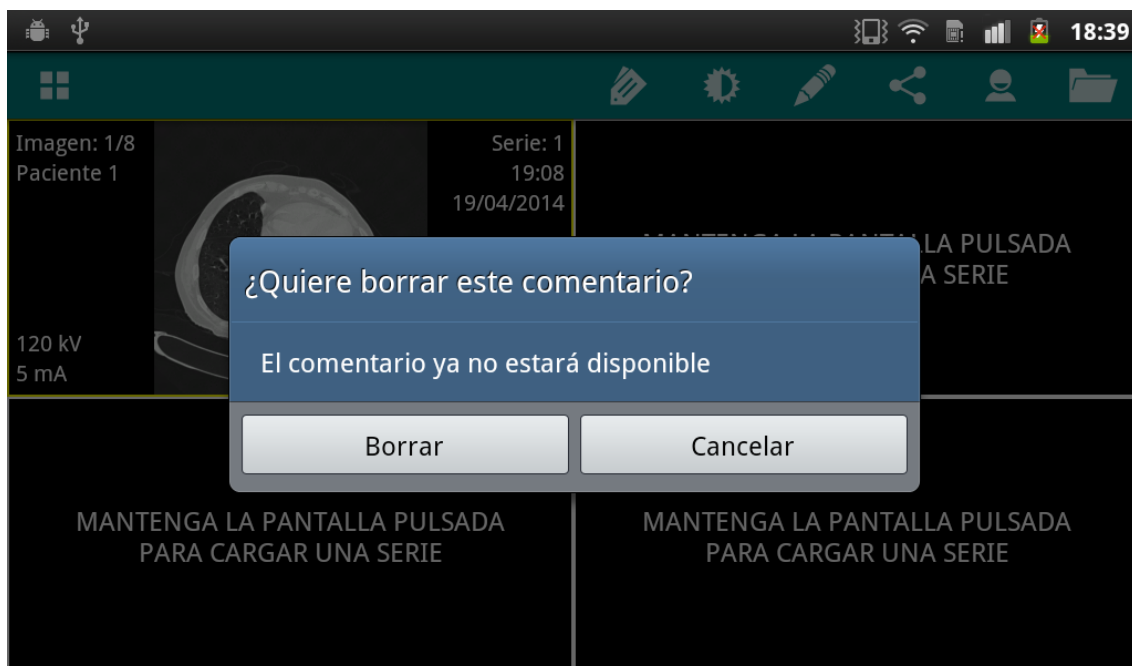


Figura 37: Esborrar comentari

4.7. Compartir

Per poder compartir una imatge amb altres metges farem servir el botó de la icona usual a Android de compartir, que és la que està situada entre la del llapis i la de la persona.

El codi per poder compartir informació fent servir aplicacions que fan servir correu electrònic és el següent:

```
Intent emailIntent = new Intent(Intent.ACTION_SEND);
emailIntent.setType("message/rfc822");
...
startActivity(Intent.createChooser(emailIntent,
    getString(R.string.share)));
```

S'ha de declarar un `Intent`, després especificar el tipus concret "message/rfc822" per a que només busqui aplicacions que estiguin relacionades amb missatgeria electrònica i finalment invocar l'`Activity` amb `startActivity()`.

Amb aquest codi, l'aplicació busca dins del dispositiu quines altres aplicacions hi ha instal·lades que compleixin els requisits i els mostrarà en una llista com la que veurem a la imatge que hi ha a continuació, al fer *clic* a un element d'aquesta llista ens obrirà l'aplicació corresponent (Figura 38).

4.8. Informació del pacient

Per a ampliar l'aplicació i fer-la més complerta, s'ha afegit una secció per a poder accedir a informació sobre el pacient que pot ajudar a donar un diagnòstic.

Arribats a aquest punt, cal aclarir la navegabilitat de l'aplicació. Mentre estàvem a les pantalles on es carregaven les imatges el botó *back* del dispositiu ens feia aparèixer un missatge d'avís per tancar l'aplicació. En canvi, en aquesta secció al prémer el botó tornarem a la pantalla principal, la de la càrrega d'imatges.

Al triar la opció del pacient, es llança una nova `Activity` que té per títol el nom de l'estudi carregat.

Cada estudi s'entén que correspon a una únic pacient, per tant és lògic mostrar informació personal. Tanmateix, una persona pot tenir més d'un estudi a la "xarxa", cosa que no fa perdre la coherència.

En aquesta `Activity` s'ha forçat a mantenir les pantalles en forma vertical per a una millor lectura del contingut i per mantenir una visió agradable (Figura 39).

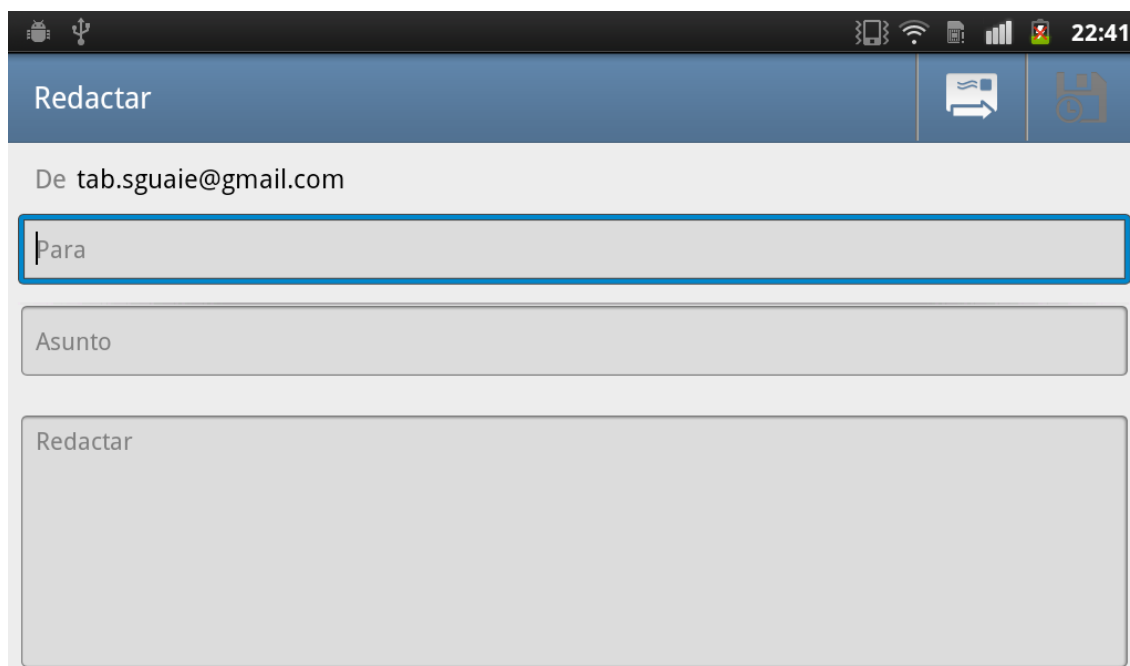
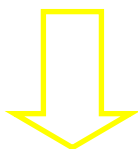
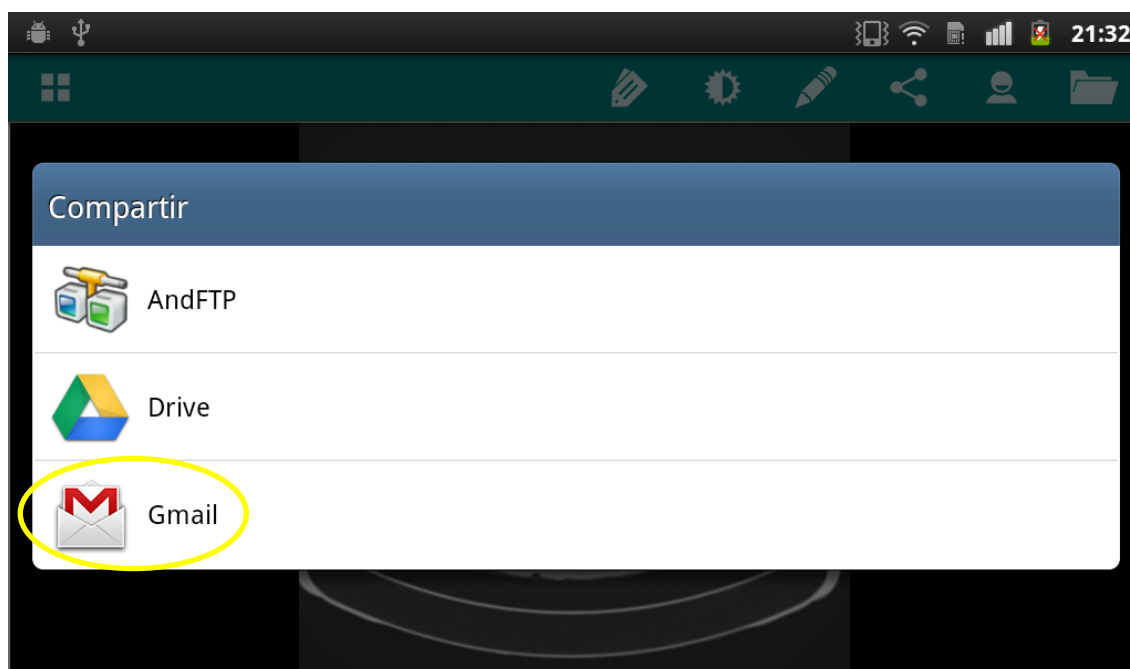


Figura 38: Compartir

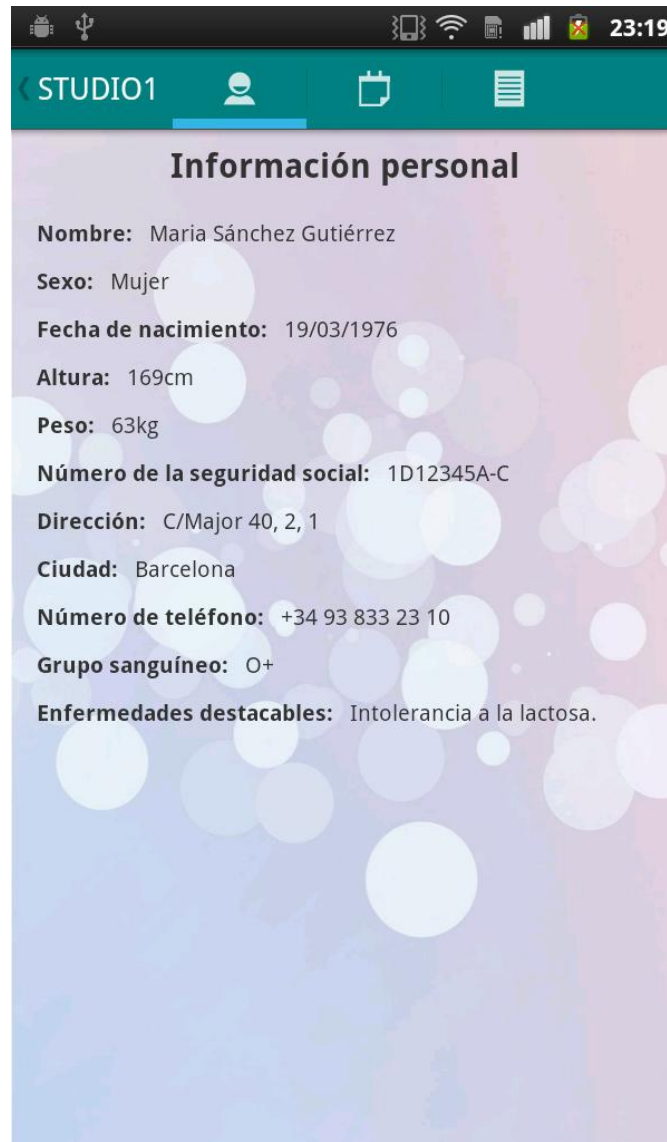


Figura 39: Informació personal

Veiem que hi ha tres icones a l'Action Bar però si ens hi fixem no són exactament iguals. En aquest cas tenen un selector que indica a quina de les seccions estem. Això és perquè no són icones creades a partir d'un fitxer de menú, sinó que són tabuladors (o Tabs). A cada Tab se li assigna un Fragment que es carregarà a la pantalla. El codi que permet fer aquesta acció és el següent:

```
...  
  
Tab tab = actionBar.newTab()  
    .setIcon(R.drawable.ic_action_person)  
    .setTabListener(new TabListener<PersonalInfoFragment>(  
        this, studioName, PersonalInfoFragment.class));  
  
actionBar.addTab(tab);
```

```
tab = actionBar.newTab()

        .setIcon(R.drawable.ic_action_event)
        .setTabListener(new TabListener<MedicalHistoryFragment>(
            this, studioName, MedicalHistoryFragment.class));

actionBar.addTab(tab);

tab = actionBar.newTab()
        .setIcon(R.drawable.ic_action_view_as_list)
        .setTabListener(new TabListener<MedicineListFragment>(
            this, studioName, MedicineListFragment.class));

actionBar.addTab(tab);
```

El primer tabulador, com hem vist anteriorment, ens permet consultar la informació personal del pacient.

El segon, ens porta a la llista de fites mèdiques del pacient, on els elements marcats en color denoten les fites que tenen a veure amb l'estudi que està carregat. Cadascun d'ells pot correspondre a una operació, una visita o qualsevol esdeveniment que es pugui enregistrar com a historial mèdic.

Si seleccionem un element de la llista, per exemple el primer, ens durà a una altra pantalla amb més informació (Figura 40). Aquest procés s'ha realitzat amb una `ListView` a la que se li ha assignat un `ArrayAdapter<MedicalHistory>` on `MedicalHistory` és una classe que hem creat per afegir-li la informació que volem passar a cada element de la llista i a més la informació que volem passar a l'`Activity` que conté la informació més detallada.

Al prémer el botó *home* de l'`ActionBar` a la pantalla dels detalls tornem a la pantalla anterior. Això segueix el patró usual de la navegabilitat que proposa Android per a les aplicacions.

L'últim dels tres Tabs mostra l'historial de medicaments que s'ha pres o es pren el pacient. L'estructura és molt semblant a la de l'historial mèdic, en aquest cas veiem que la informació de cada element de la llista és diferent.

Al fer *clic* a algun element de la llista també mostra informació més detallada. En aquest cas, a més es mostra una imatge corresponent al medicament seleccionat centrada horitzontalment a la part superior de la pantalla (Figura 41).

El procediment per a programar aquest apartat és molt semblant a la programació de l'anterior. En aquest cas s'ha omplert un `ListView` amb un `ArrayAdapter<Medicine>` on `Medicine` té com a particularitat un camp amb una URL on hi ha la imatge del medicament.

La navegabilitat és com la del segon Tab, al donar des de la pantalla dels detalls a la "fletxa" del botó *home* de l'`Action Bar` tornem al menú principal d'informació del pacient.

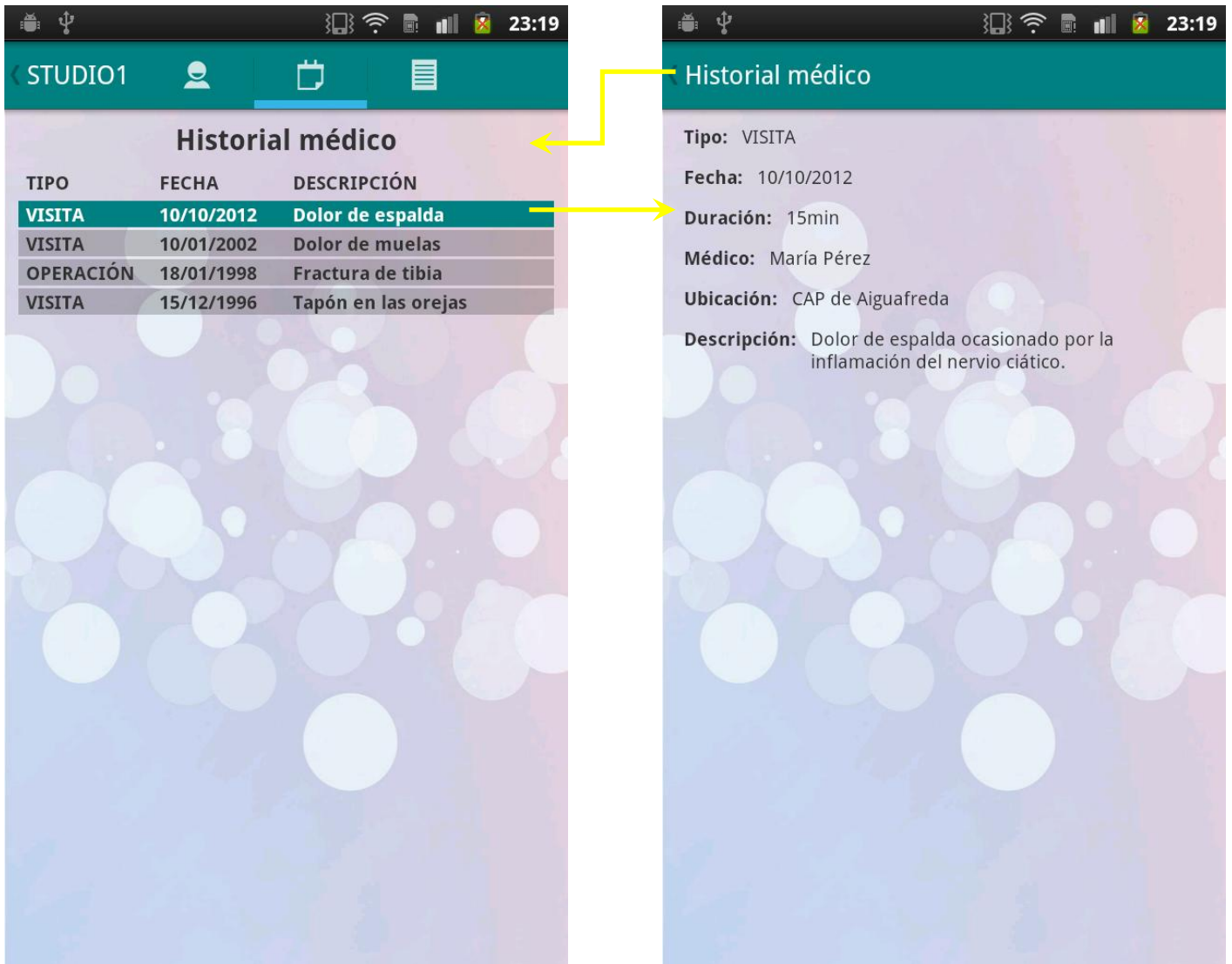


Figura 40: Historial mèdic

4.9. Títol de l'aplicació

Per finalitzar el capítol només cal comentar quin és el títol final que se li ha donat a l'aplicació.

Degut a la funcionalitat que té, hem cregut convenient anomenar-la MIVApp, MIV són les inicials de *Medical Image Viewer*, que en català vol dir Visor d'Imatges Mèdiques. App és el sufix que se li posa a moltes aplicacions per destacar que són productes pensats per a dispositius mòbils, ja que App ve de la paraula Application, que en català significa aplicació.

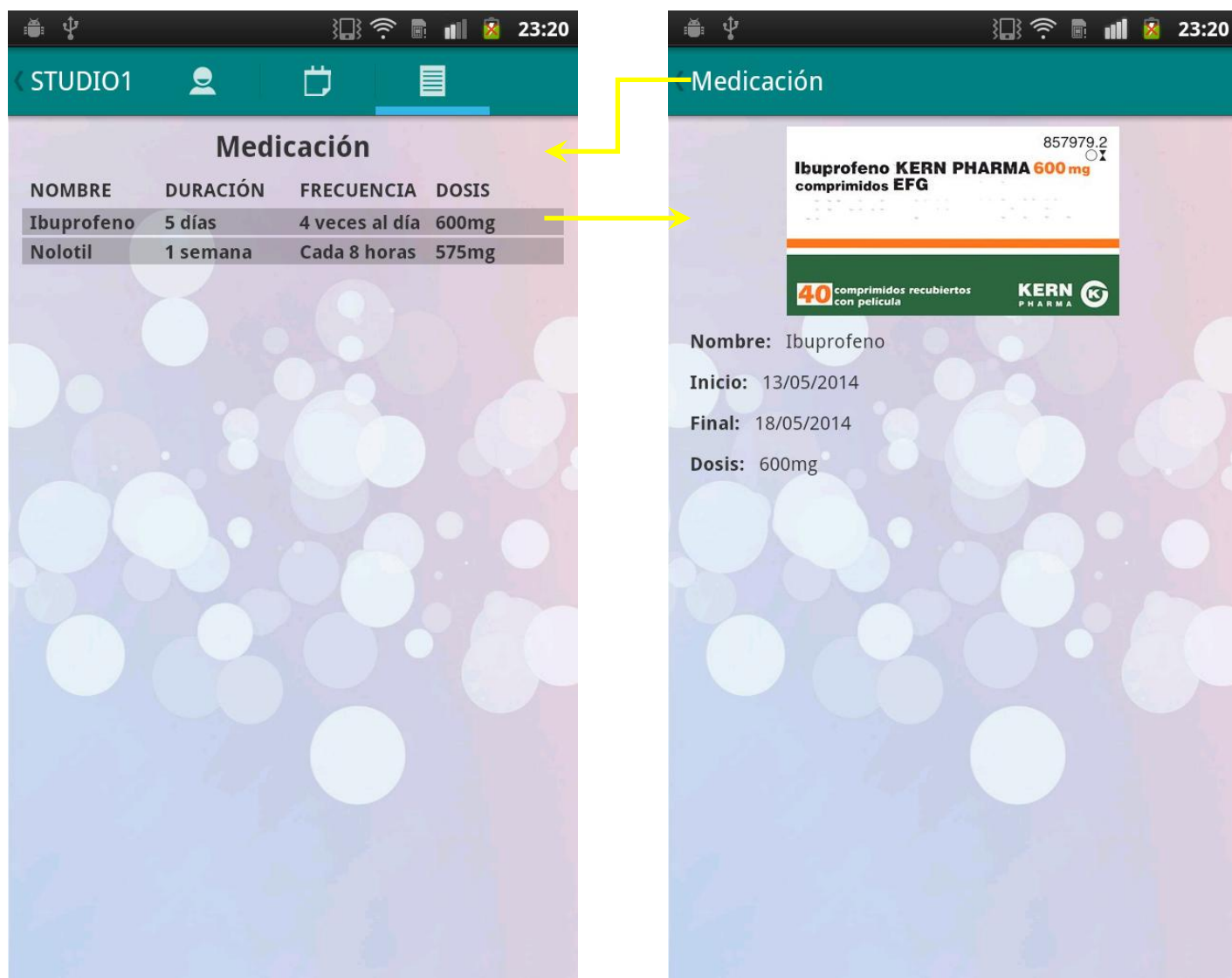


Figura 41: Medicació

CAPÍTOL 5

PLANIFICACIÓ

La planificació inicial del projecte va ser la següent:

Setmana del 17/12/2012	<ol style="list-style-type: none">1) Crear 5 <i>personas</i>.2) Mapeig de processos.3) Entrevista.4) Storyboarding: Vinyetes explicatives del projecte.
Setmana del 18/02/2013	<ol style="list-style-type: none">1) Prototip de paper: Crear la maqueta de la <i>tauleta</i> amb diferents pantalles (ha de contenir varis dissenys diferents, per després poder comparar-los entre ells).2) Avaluació heurística: Realitzar una pauta per saber si s'estan seguint les 10 heurístiques de Jakob Nielsen.
Setmana del 25/02/2013	<ol style="list-style-type: none">1) Maqueta digital. Realitzar una recreació fidel de l'aplicació amb el programa Balsamiq Mockup.2) Avaluació heurística (aplicar-la).
Setmana del 11/03/2013	<ol style="list-style-type: none">1) Versió en Android. Primeres pantalles funcionant.2) Avaluació heurística (aplicar-la).
Setmana del 29/04/2013	<ol style="list-style-type: none">1) Versió final.
Setmana del 27/05/2013	<ol style="list-style-type: none">1) Memòria escrita.

Degut a la càrrega de feina que comportava el projecte, ens vam adonar que seria convenient separar-lo en dos parts diferenciades que s'han dut a terme en dos cursos diferents.

Curs 12/13 (cerca de requeriments i interacció persona-ordinador):

Setmana del 17/12/2012	<ol style="list-style-type: none">1) Primer contacte amb el centre mèdic UDIAT del parc Taulí de Sabadell per demanar col·laboració en el projecte i explicar en què consistirà.2) Analitzar la trobada i realitzar el curs on-line de Human-Computer Interaction [1].
Setmana del 18/02/2013	<ol style="list-style-type: none">1) Curs de Human-Computer Interaction finalitzat.2) Crear 5 <i>personas</i>.3) Storyboarding: Vinyetes explicatives del projecte.4) Mapeig de processos.
Setmana del 25/02/2013	<ol style="list-style-type: none">1) Preparació de l'entrevista.2) Disseny de l'avaluació heurística.
Setmana del 11/03/2013	<ol style="list-style-type: none">1) Realització de l'entrevista.2) Recollida de resultats i planificació de l'estructura i contingut de l'aplicació.3) Avaluació heurística: Realitzar una pauta per saber si s'estan seguint les 10 heurístiques de Jakob Nielsen.
Setmana del 1/04/2013	<ol style="list-style-type: none">1) Primera versió del prototip de paper.2) Avaluació heurística (aplicar-la).3) Programar quedades periòdiques de seguiment per avaluar el prototip fins que arribem a una versió que compleixi els requeriments.
Setmana del 5/05/2013	<ol style="list-style-type: none">1) Versió definitiva del prototip de paper.2) Inici de la recerca per aprendre a programar en Android i realització d'exercicis fins al curs següent.

Observem que en aquest cas hem descartat la realització d'una maqueta digital, ja que el programa era de pagament i hem considerat més important dedicar el temps a aprofundir en altres tasques.

Curs 13/14 (implementació de l'aplicació i memòria):

Setmana del 7/10/2013	<ol style="list-style-type: none">1) Aplicació funcionant amb l'estructura de les icones de l'Action Bar i el canvi visualització d'una a quatre imatges implementat. (Sense descàrrega d'imatges).2) Funcionalitat del botó per validar-se i carregar un estudi implementada.
Setmana del 9/12/2013	<ol style="list-style-type: none">1) Càrrega d'una imatge des d'una URL.2) Canviar de color les imatges.
Setmana del 3/02/2014	<ol style="list-style-type: none">1) Poder seleccionar una sèrie dins d'un estudi i poder paginar-les mantenint totes les funcionalitats i guardant l'estat (com per exemple el color).
Setmana del 3/03/2014	<ol style="list-style-type: none">1) Mostrar i ocultar informació a les cantonades.2) Zoom a les imatges.
Setmana del 31/03/2014	<ol style="list-style-type: none">1) Afegir comentaris, editar-los o esborrar-los.2) Funcionalitat de compartir.
* Setmana del 31/03/2014	<ol style="list-style-type: none">1) Afegir secció per accedir al a informació del pacient, com l'historial mèdic o els medicaments que pren.
Real: 5/05/2014	<p>* aquesta funcionalitat no estava prevista als plans inicials .</p>
* Setmana del 12/05/2013	<ol style="list-style-type: none">1) Versió definitiva de la memòria. <p>* degut a la inclusió d'una funcionalitat nova, la presentació es posposa fins a la convocatòria de setembre per poder realitzar la memòria.</p>

CAPÍTOL 6

CONCLUSIONS I FUTUR DEL PROJECTE

6.1. Conclusions

Els objectius d'aquest projecte es poden resumir en dos grans blocs:

1. Fer un estudi exhaustiu i profund de com dur a terme un anàlisi de requeriments i cerca de necessitats per realitzar un bon producte que ofereixi solucions a problemes reals d'uns usuaris concrets, en aquest cas dels metges de l'hospital Parc Taulí de Sabadell.
2. Aprendre a programar per a dispositius mòbils amb el sistema operatiu Android i plasmar els resultats anteriors en una aplicació real.

El primer objectiu, la part de interacció persona-ordinador, s'ha dut a terme tal com estava planejat, tenint en compte les limitacions que hi ha per a accedir al sector mèdic, perquè no és possible que puguin gastar molt de temps en un projecte com el que ens ocupa.

S'ha aconseguit en gran part el segon objectiu, ja que s'ha programat una aplicació amb la majoria de funcionalitats que es pretenia. Tot i això, per qüestions de temps i dedicació, no s'ha pogut implementar l'accés a imatges mèdiques reals amb el protocol JPIP i s'ha substituït per la descàrrega d'una imatge png emmagatzemada a una URL.

L'error més gran que hem comès és pensar que es podria aconseguir realitzar tota la feina en un sol curs i obtenir resultats satisfactoris i tangibles, però no ha pogut ser així ja que tant el primer com el segon punt portaven molta més càrrega de feina del que es va estimar en primera instància i aleshores s'ha hagut de dividir el projecte en dos cursos, a grans trets, un per a cada objectiu.

El projecte pot ser de molta utilitat per a una utilització real, ja que així ha estat plantejat, tot i que amb el temps que hi hem treballat podríem plantejar un millor enfocament. Encara que hi ha aplicacions per a iOS que cobreixen molt bé les funcions que cobreix aquest projecte, per a dispositius Android no està tan explotat. Per a poder ser una aplicació útil per a un metge hauria d'incorporar la visualització d'imatges amb el protocol JPIP, realitzar algunes iteracions més en el procés d'avaluació heurística i poder tenir més contacte amb usuaris finals reals. Poder continuar el projecte per a adaptar-lo amb més detall a les necessitats reals seria interessant, oferiria més opcions en el mercat a part de l'iPad.

Durant el transcurs del projecte hem après la importància d'un bon plantejament inicial i una bona anàlisi de requeriments a l'hora de realitzar una aplicació, i en el fons de qualsevol projecte. Alguns aspectes en aquest procés poden semblar

ximpleries però són realment importants ja que estalvien molta feina i recursos. No és una tasca menyspreable i requereix més temps, esforç i dedicació de la que en un principi es pot estimar, però es veu recompensat en la qualitat, l'experiència d'usuari i l'assoliment d'objectius. Aquest fet ens ha aportat una experiència i un punt de vista molt més crític en els primers passos de la realització d'un projecte.

Fer aquest projecte també ens ha aportat com a valor afegit aprendre a programar per a Android, fet que té molt valor en els temps actuals on els *smartphones*, en especial els que porten Android, estan arribant a les butxaques de la gran majoria de les persones amb cada vegada més rellevància en les nostres vides.

6.2. Futur del projecte

Per acabar deixem una llista d'alguns punts on es podria millorar o ampliar l'aplicació degut sobretot al que acabem de comentar:

- L'objectiu inicial era realitzar la càrrega d'imatges amb el protocol JPIP, ja que és el mètode que utilitzen al parc Taulí de Sabadell.
- A la opció de compartir, un cop assolit el punt anterior, es podria afegir un enllaç que portés a una imatge o estudi mitjançant JPIP. A més, s'hi podria afegir la informació dels comentaris que s'hi han fet.
- Seria interessant poder tocar un punt de la imatge i que quedi marcat, a més també era un requeriment poder obtenir una distància o una àrea d'una imatge.
- Implementar alguns efectes visuals més elaborats.

CAPÍTOL 7

ÍNDEX D'IMATGES

Figura 1: Memex, càmera.....	10
Figura 2: ENIAC.....	11
Figura 3: Prototip de paper 1.....	18
Figura 4: Prototip de paper 2.....	19
Figura 5: Avaluació heurística.....	20
Figura 6: Procés iteratiu.....	21
Figura 7: Cicle de vida d'una <i>activity</i>	28
Figura 8: Toast.....	30
Figura 9: AlertDialog.....	30
Figura 10: Tancar aplicació.....	38
Figura 11: Action Bar.....	38
Figura 12: Visualitzar una o quatre imatges.....	39
Figura 13: Toast carregar estudi.....	40
Figura 14: Validació.....	41
Figura 15: Validació errònia.....	41
Figura 16: Validació satisfactòria.....	41
Figura 17: Carregar estudi.....	42
Figura 18: Estudi no trobat.....	42
Figura 19: Estudi carregat.....	42
Figura 20: Sèries.....	43
Figura 21: Progress Bar.....	44
Figura 22: Imatge carregada.....	44
Figura 23: Opció quatre imatges.....	45

Figura 24: Passar imatge.....	45
Figura 25: Zoom.....	46
Figura 26: Zoom doble toc.....	47
Figura 27: Informació.....	48
Figura 28: Informació pantalla partida.....	49
Figura 29: Tria de colors.....	49
Figura 30: Green temperature.....	50
Figura 31: Hot metal contour.....	50
Figura 32: Red temperature.....	51
Figura 33: Spectrum.....	51
Figura 34: Comentaris.....	52
Figura 35: Icona d'informació.....	52
Figura 36: Edició de comentaris.....	53
Figura 37: Esborrar comentari.....	54
Figura 38: Compartir.....	56
Figura 39: Informació personal.....	57
Figura 40: Historial mèdic.....	59
Figura 41: Medicació.....	60

CAPÍTOL 8

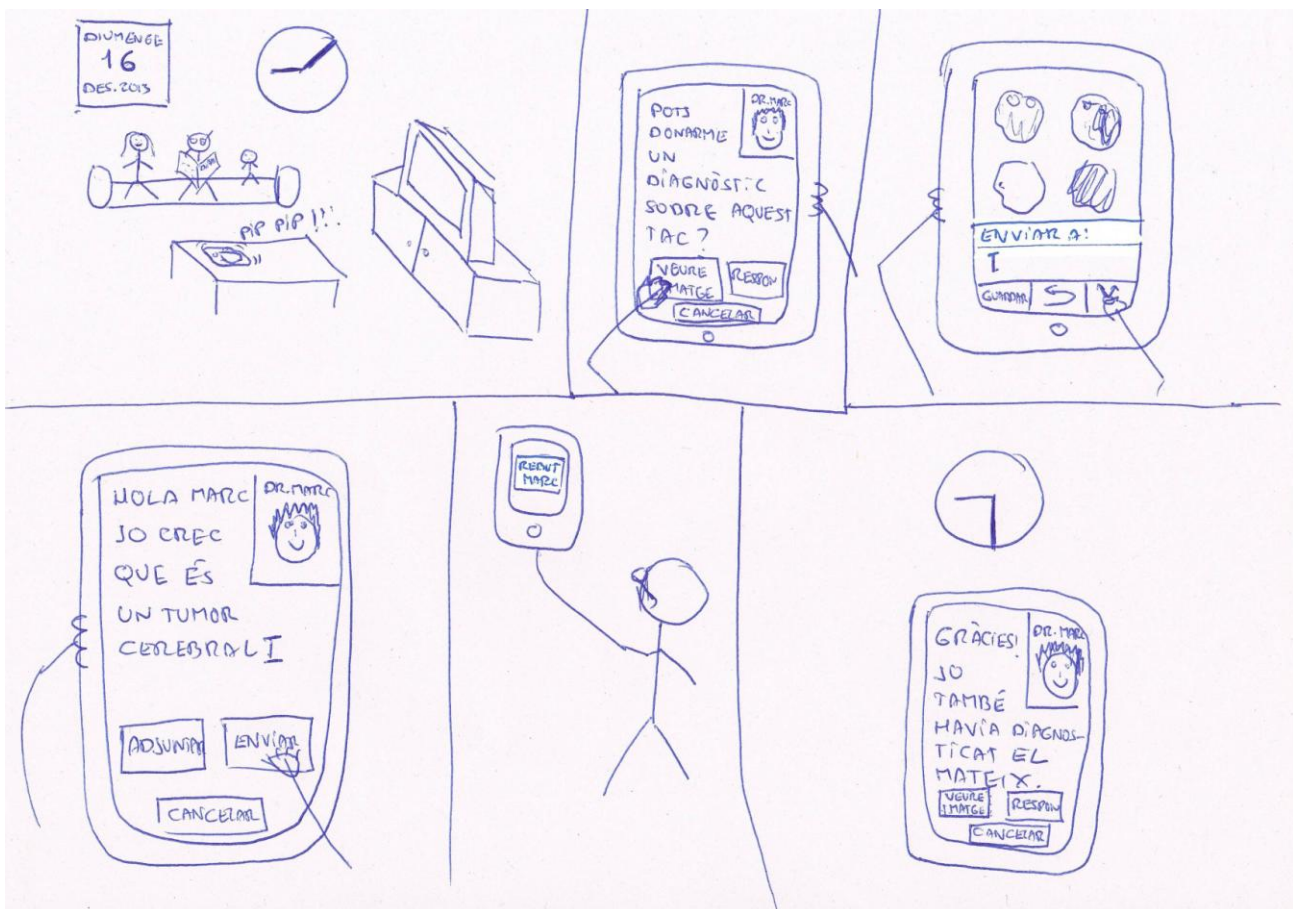
BIBLIOGRAFIA

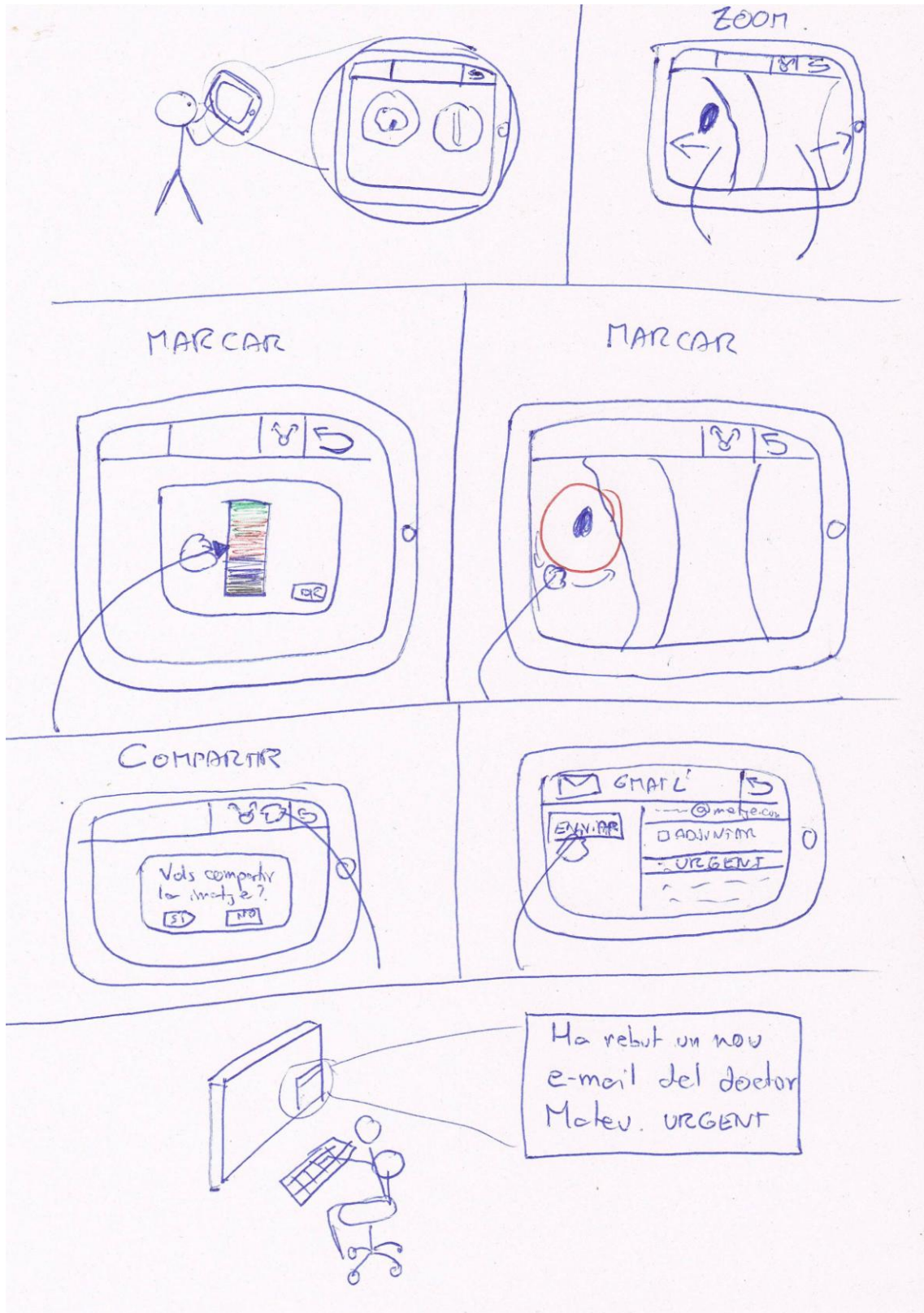
- [1] Klemmer, S. (2012). “Human-Computer Interaction”. Disponible a:
<<https://class.coursera.org/hci-2012-002/lecture/index>> [Consulta: setembre de 2013]
- [2] Kuniavsky, Mike. *Observing The User Experience. A practitioner's guide to user research*, San Francisco:USA, 2003
- [3] Bush, V. (1945). “As We May Think”. Disponible a:
<<http://www.theatlantic.com/magazine/archive/1945/07/as-we-may-think/303881/>>
[Consulta: agost de 2013]
- [4] Nielsen, J. (1995). “10 Usability Heuristics for User Interface Design”. Disponible a:
<<http://www.nngroup.com/articles/ten-usability-heuristics/>> [Consulta: desembre de 2013]
- [5] Nielsen, J. (1995). “How to Conduct an Heuristic Evaluation”. Disponible a:
<<http://www.nngroup.com/articles/how-to-conduct-a-heuristic-evaluation/>> [Consulta: desembre de 2013]
- [6] Google Inc. (2014). “Android Developers”. Disponible a: <<https://developer.android.com/>>
[Consulta: juliol de 2014]
- [7] Google Inc. (2014). “Get the Android SDK”. Disponible a:
<<https://developer.android.com/sdk/index.html#download>> [Consulta: juliol de 2014]
- [8] GitHub (2013). “Mike Ortiz/TouchImageView”. Disponible a:
<<https://github.com/MikeOrtiz/TouchImageView>> [Consulta: gener de 2013]

CAPÍTOL 9

ANNEXOS

9.1 Storyboards





9.2 Prototip de paper

9.3 Entrevista

Introducció

El projecte que ens proposem dur a terme consisteix en realitzar una aplicació enfocada a la visualització d'imatges mèdiques per a tauletes amb sistema operatiu Android.

Donats els recursos que tenim a disposició, no pretenem fer una aplicació que substitueixi l'aplicació que fan servir actualment, però sí proposar-ne una de nova que pugui aportar alguna millora i algun punt de vista diferent que resulti interessant. L'entrevista està enfocada, doncs, a poder traçar un camí a seguir per aconseguir aquests objectius.

Durant la primera trobada vam fer un primer anàlisi de requeriments, però ara pretenem aprofundir molt més amb una entrevista més formal. És per això que és possible que torni a aparèixer algun concepte del que ja es va parlar, perquè l'objectiu és ampliar les respostes i obtenir més detalls.

Escalfament

Abans de començar amb la part del disseny, introduïrem quin és l'objectiu del projecte i el que es vol aportar.

Entrevista

S'ha de tenir en compte que l'entrevistat no està sotmès a cap examen, no s'espera una resposta concreta i cap resposta és incorrecta.

Qüestions Generals:

1. Quin procés segueix un metge des de que decideix realitzar una imatge mèdica fins que dona un diagnòstic?
2. Quin tipus de dispositius electrònics fan servir els metges habitualment?
3. De qui és propietat el dispositiu?
4. A quins dispositius es visualitzen aquestes imatges?
5. Què ha de ser possible fer amb l'aplicació?
6. En quines situacions es fa servir l'aplicació?

7. On es fa servir l'aplicació?
8. Com ho fan els metges per demanar una segona opinió per a un diagnòstic?
9. Es comuniquen amb metges estrangers?

Enfocament Profund:

10. Quins especialistes realitzen diagnòstics o consulten imatges mèdiques?
11. Amb quina finalitat consulta imatges mèdiques cada especialista? (Per exemple, una infermera pot consultar una imatge mèdica però no amb la mateixa finalitat que un cardiòleg).
12. Quines d'aquestes persones utilitzen l'actual aplicació i amb quina finalitat?
13. Quan es dona per definitiu un diagnòstic?
14. Com s'han de mostrar les imatges?
15. Quines operacions es poden realitzar sobre les imatges? (Zoom, contrast, color, marcar...)
16. Per a què es canvia el color d'una imatge?
17. Com funciona la barra d'eines del canvi de colors d'una imatge?
18. Quin tipus de marques es fan a les imatges?
19. Quines conseqüències té marcar una imatge? (Es pot treure la marca? Es queda guardada al guardar la imatge?..)
20. De quina manera es vol fer la comunicació entre dos usuaris de l'aplicació per enviar-se una imatge?
21. On s'han d'emmagatzemar els diagnòstics d'un pacient? (Historial o expedient)
22. Com es mostra la informació del pacient?
23. Com es preserva la privacitat de les dades dels pacients? (Dades encriptades? Permisos?)
24. Quin protocol es segueix per desar-los? (Es fa servir alguna mena de formulari?)

25. Quin tipus d'imatges mèdiques s'espera visualitzar?
26. Quines són les mesures de les imatges que es visualitzaran a l'aplicació?
27. Quina profunditat han de tenir les imatges?
28. On estan emmagatzemades les imatges?
29. Quant ocupen les imatges que s'han de visualitzar a la *tauleta*?
30. Quin protocol s'ha de seguir per accedir-hi? (Hi ha control d'accés? Contrasenyas?)
31. En quins idiomes es comuniquen els metges?
32. Quins elements identificatius han d'aparèixer a la pantalla d'inici? (Logos)

Cloenda

Per acabar, agraïm l'aportació i el temps que ens han dedicat, que és molt valuós. Si hi ha alguna aportació més que es vulgui fer serà molt ben rebuda.

Resum esquemàtic dels resultats de l'entrevista

- Diferents metges intervenen en el procés de realitzar una imatge mèdica: el metge que atén al pacient decideix fer una prova, envia una sol·licitud electrònica explicant els motius, els símptomes, i finalment el radiòleg o el metge corresponent realitza la radiografia o la imatge mèdica corresponent.

- Historial clínic:

Resultats -> laboratori, patologia.

Informe/Imatge -> accés per a obrir imatge (opcional), un enllaç obrirà un

Visor.

- L'aplicació ha d'estar enfocada al metge que realitza informes des de casa (UDIAT Parc Taulí). Segona opinió.

- Tipus d'imatges:

Mamografia / Tòrax -> resolució alta.

Tags / resonància -> resolució mitjana.

- Exemple de cas d'ús:

El tècnic és qui captura la imatge, revisa que s'hagi realitzat correctament.

El radiòleg -> guàrdies/teleclínic fent informes per gent d'Austràlia.

Metge de planta -> passa visita (mateix perfil que metge de referència però passant a planta)

- L'aplicació ha de ser un client, ha de buscar coses, està sotmesa a una altra.
- Associar aplicació amb protocol JPIP.
- Els metges habitualment disposen d'un ordinador compartit, com és un servei públic són antics, es renoven cada 3 o 4 anys.
- Les imatges no es poden guardar, s'ha d'eliminar totes les facilitats per a que una imatge sigui emmagatzemada al dispositiu.
- Aplicació sota demanda, només visualitzar, no descarregar.
- S'ha de poder mesurar en escala (mil·límetres)
- S'ha de poder fer zoom.
- Canvi de color.
- Window Level.
- Les connexions normals a les que tindran accés seran wifi, 3G habitualment.
- Per demanar una segona opinió es passen un número d'historial de referència, intercanvien un enllaç. El protocol per compartir l'enllaç és el DICOM.
- El visor ha de mostrar de manera contínua una sèrie.
- No hi ha comunicació internacional però és interessant afegir traducció.
- Quan un diagnòstic es signa és quan es dona per definitiu.
- Des de la tauleta no es signa.
- La informació de la imatge que s'està mostrant es veu a les cantonades de la pantalla:

Dalt-esquerra: informació pacient, número d'imatge.

Dalt-dreta: informació sèrie, contrast, temps en que va ser feta, configuració de la màquina...

Baix-esquerra: configuració hardware kV.

Baix-dreta: part anatòmica que s'està veient.

- Un estudi està compost per sèries, cada sèrie conté diferents objectes que estan compostats per imatges amb la seva informació.
- S'ha d'implementar una autenticació que estigui per sobre, encriptar url.
- Time-out.

9.4. Personas

Joan Martínez

Cap de radiologia a la UDIAT del parc Taulí de Sabadell.

Descripció Personal

És un home de 54 anys, seriós, educat i elegant. És tímid, però aconsegueix dissimular-ho sempre bastant bé.

Va acabar la carrera de medicina a la Universitat Autònoma de Barcelona als 26 i es va especialitzar en radiologia durant els tres anys següents.

Va estar en pràctiques al centre mèdic UDIAT, i va continuar d'assistent de radiologia 4 anys més. Després va ocupar el càrrec de radiòleg.

Finalment a l'edat de 45 anys va ascendir a cap de radiologia, després de diversos diagnòstics de mèrit, fonamentals per a la recuperació dels pacients.

En Joan porta casat des dels 30 anys amb la Carme Pujadas, una dona de 50.

Actualment resideixen a Sabadell, a una casa gran, amb jardí, amb la seva única filla de 18 anys.

Els dies que no treballa, a en Joan li agrada marxar a córrer per la muntanya escoltant música amb els auriculars.

La família és aficionada a esquiar, és per això que disposen d'una segona residència a la Vall de Boí on hi passen els caps de setmana i les vacances d'hivern.

Coneixements Tecnològics

No ha estat mai amant de les tecnologies, però disposa d'un ordinador personal que fa servir quan té feina acumulada i per parlar amb amics.

No està inscrit a cap xarxa social perquè no li dóna confiança fer públiques les seves dades per Internet.

Preocupacions

La seva major preocupació és la seva filla, ja que està en una època rebel i sempre s'estan discutint perquè no vol continuar estudiant després de l'últim curs de batxillerat que està cursant enguany.

Objectius

La seva feina és coordinar tots els radiòlegs i enviar les imatges mèdiques a l'especialista corresponent, com per exemple, els tacs cerebrals al neuròleg.



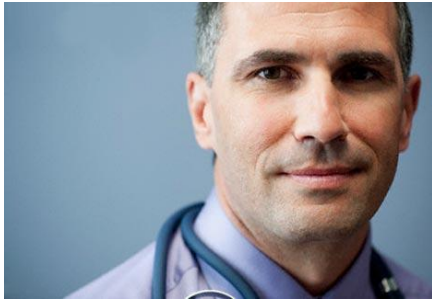
Eventualment fa els diagnòstics sobre les ressonàncies més dubtoses, però mai tracta amb pacients directament.

Sempre es proposa no tenir gens de feina acumulada pel divendres, perquè és el dia que té més ganes de descansar i estar amb la família.

Es sent amb la responsabilitat de no donar mai cap diagnòstic erroni i es sent satisfet cada vegada que contribueix a millorar la qualitat de vida d'un pacient.

Vol ascendir a cap de planta perquè té en ment canviar-se el cotxe per un 4x4 i li agradaria tenir més diners.

Pedro Jiménez



Radiòleg en plantilla a la UDIAT del parc Taulí de Sabadell.

Descripció Personal

És un home de 40 anys, amb caràcter molt obert, simpàtic, rialler i atractiu. El seu atractiu físic i la seva confiança en ell mateix l'han conduït a tenir moltes relacions amoroses amb noies durant tota la vida. Tot i això mai ha aconseguit tenir una relació llarga amb cap d'elles, ja que la seva prepotència i supèrbia acaba per decebre-les.

Actualment és solter i viu de lloguer a un pis a Cerdanyola del Vallès.

Va néixer a Granada, ciutat on es va treure la carrera de medicina. Va estar treballant de radiòleg a l'hospital de Granada fins els 37 anys, fins que degut a la crisi econòmica hi va haver una reestructuració de plantilla i va haver de marxar a Catalunya, on va entrar a formar part de l'equip de radiologia de la UDIAT del parc Taulí de Sabadell.

Els seus veïns són molt sorollosos, per això no li agrada passar-se gaire estona a casa, és per això que es passa les hores lliures a la biblioteca llegint.

Coneixements Tecnològics

Una de les seves aficions és jugar a les consoles portàtils.

Li agraden molt els smartphones i es descarrega moltes aplicacions pel seu dispositiu mòbil amb sistema operatiu Android.

Fa servir la seva *tauleta* per comunicar-se amb la seva família a Granada. A part de per a les seves funcions de metge quan està de guàrdia.

Preocupacions

Cada vegada està més preocupat per la seva vida amorosa, ja que a la seva edat encara no ha aconseguit mantenir una relació seriosa amb cap dona i cada vegada ho veu més complicat.

Com està cansat de la seva situació al seu pis, està cada vegada més preocupat pel sou que cobra, que creu insuficient.

Objectius

La seva feina a l'hospital consisteix en realitzar diagnòstics a partir de les imatges mèdiques que li proporcionen. És molt observador i això el fa ser molt bo en la seva feina.

Té tracte directe amb cada pacient el cas del qual estudia, té molta cura amb les persones que tracta i sap donar els seus diagnòstics de la manera més suau possible, en cas que siguin negatius, i esperançadora en cas de que siguin positius.

Un cop fet el diagnòstic fa un informe detallat, l'adjunta a l'historial del pacient i finalment li envia una còpia al cap de radiologia.

Té en ment tornar a Granada, quan tingui una oportunitat laboral que l'ompli. La seva il·lusió és ser director general d'hospital. Però no es limita només a buscar feina de metge.

Pere Juancomartí

Estudiant de medicina en pràctiques encarregat de fer els tacs al centre mèdic UDIAT del parc Taulí de Sabadell.



Descripció Personal

És un noi de 22 anys, humil, discret i molt amable.

És superdotat.

És molt emprenedor i sovint autodidacta.

Ha tret notes de mèrit a la carrera i per això ha obtingut una beca que li ha permès estar al centre mèdic per ampliar la seva experiència.

Viu a casa dels pares, a Barcelona, al barri de Pedralbes. És fill únic.

Manté una relació sentimental amb una noia de Terrassa, de 23 anys, estudiant de matemàtiques a la UAB.

Una de les seves aficions és viatjar i conèixer món. Amb la seva xicota ha viatjat a nou països diferents.

Sap parlar anglès i xinès a un gran nivell, llengües que ha après pel seu compte.

Coneixements Tecnològics

És un apassionat de la tecnologia i el seu ídol és Steve Jobs.

Ha après a programar en el seu temps lliure i ha realitzat unes quantes aplicacions exitoses que són disponibles per descarregar-se per a Android al PlayStore.

Preocupacions

És una persona molt feliç, no li preocupa el seu sou, tampoc acabar els estudis immediatament. Segurament, hi influeix el fet que cobra molt bé per ser estudiant en pràctiques i que els seus resultats acadèmics són extraordinaris.

Objectius

Té pensat demanar matrimoni a la xicota i marxar a viure amb ella després d'acabar la carrera. El pis que tenen pensat llogar és a Sabadell.

La seva feina al parc Taulí consisteix en fer i revisar les imatges mèdiques. Si hi troba alguna anomalia, mana repetir-la, i si l'anomalia persisteix li hi fa una marca a l'anomalia i hi adjunta un informe abans d'enviar-la al metge pertinent.

Quan acabi la carrera, vol marxar a treballar de metge cirurgia als Estats Units.

Rosa Ridao



Neuròloga especialista en Parkinson al parc Taulí de Sabadell.

Descripció Personal

La Rosa Ridao, de 34 anys, es va especialitzar en neurocirurgia a la Universitat de Barcelona. En acabar va ser contractada per la UDIAT del parc Taulí de Sabadell per a tractar amb pacients amb problemes neurològics.

És una noia molt compromesa amb el medi ambient, amb molta empatia.

Sap cinc idiomes: Català, castellà, anglès, alemany i francès.

Després de 5 anys intentant l'adopció d'una nena de l'Àfrica, fa mig any que va anar a Zàmbia a buscar la seva filla Sheila, que ara té un any, i s'ha convertit en mare soltera.

Viu feliç i orgullosa amb la seva filla a una caseta a Bellaterra, al mateix carrer que els seus pares, amb qui té una molt bona relació.

Al seu temps lliure queda amb els amics per dinar, fer un cafè o anar al cinema.

Li agrada molt la natació, és per això que fa dues hores tres dies a la setmana sessions de natació a una piscina coberta. Com creu que és important aprendre a nedar des de nadons, també hi ha apuntat la seva filla petita amb monitors especialitzats.

Coneixements Tecnològics

No és una gran apassionada de la tecnologia, tot i això, disposa d'un iPhone.

També té una càmera fotogràfica Canon EOS 650D que ha adquirit recentment per a poder fer fotos a la seva filla.

A part, de més jove va realitzar cursos d'ofimàtica.

Amb l'avenç de les tecnologies domina perfectament els ordinadors i habitualment xateja, entra a les xarxes socials. També navega per Internet per realitzar compres alguna vegada.

Preocupacions

Actualment la seva màxima preocupació és la seva filla Sheila.

També està neguitosa per la situació econòmica del país i li preocupa que en un futur empitjori i hagi de deixar de treballar amb la conseqüència de la pèrdua d'ingressos.

Objectius

La seva feina consisteix en interpretar els tacs cerebrals que li arriben. És molt bona diagnosticant prematurament malalts de Parkinson i d'aquesta manera se'ls pot facilitar una vida normal.

Es dedica tant a la investigació com a realitzar diagnòstics. Per la part d'investigació treballa a mitja jornada amb un equip de científics aportant la seva opinió a partir de l'experiència. Per la part de diagnòstics, té consultes amb els pacients, si creu que presenta símptomes de Parkinson mana fer-li un tac, si presenta símptomes d'una altra malaltia els deriva a l'especialista oportú.

Degut a la seva recent maternitat, s'està plantejant deixar la part d'investigació per a dedicar-se més a la seva filla.

Maria Planas



Cap d'infermeria a l'hospital de Sabadell.

Descripció Personal

La Maria Planas, de 48 anys, va realitzar la carrera d'infermeria i va estar treballant a l'hospital Sant Joan de Déu de Barcelona d'infermera tot just acabar la carrera. Va esdevenir cap d'infermeres al cap dels anys al mateix centre i va obtenir una molt bona reputació, pel seu tracte humà i simpatia amb els malalts i per la seva eficiència a la feina.

És una dona patidora, senzilla i molt familiar. És molt estalviadora.

Està casada amb en Martí i té un fill de dotze anys, en Miquel i dues bessones de quinze, la Júlia i la Sílvia. Viuen a Sant Quirze del Vallès a un pis gran amb quatre habitacions de lloguer.

Va demanar el trasllat quan al seu home es va quedar a l'atur fa quatre mesos i va trobar feina a Sabadell, aleshores va entrar directament com a cap d'infermeres al Parc Taulí de Sabadell, ja que va substituir a l'anterior que es jubilava.

Els fills van a l'escola Pía de Sabadell i en Martí els porta cada dia quan marxa cap a la feina.

Els agrada molt passejar en família els caps de setmana, fan moltes sortides a museus i turisme per Catalunya.

Coneixements Tecnològics

No és gaire entesa en els dispositius tecnològics de nova generació. A casa disposen d'un ordinador i de consoles portàtils que van servir els fills, però ella no en fa mai ús.

Preocupacions

Li agradaria molt aprendre anglès, ja que considera que és el seu punt feble. Però no disposa de gaire temps lliure per apuntar-se a una acadèmia per aprendre'n.

El seu principal objectiu és mantenir una bona relació amb tot el personal i que els seus pacients agraeixin la seva presència en la mesura del possible, ja que a l'entrar directament com a cap d'infermeres li preocupa no ser ben vista.

Objectius

S'està planejant, juntament amb el seu home, mudar-se a Sabadell, ja que tenen un bon sou i porten estalviant molts anys, d'aquesta manera estarien més a prop de la feina i de l'escola de dels fills.

La seva feina consisteix en donar el vist i plau dels medicaments i les dosis que apliquen les infermeres a les persones malaltes que estan a planta, per tant, ha de tenir nocions bàsiques de totes les malalties possibles. Tot i això, no vol dir que no tingui tracte amb les persones, ja que ella també passa per les habitacions per donar un informe dels malalts cada matí, com una infermera més.

Quan hi ha una malaltia que no coneix, ha de realitzar un informe detallat, amb la descripció dels símptomes, amb la corresponent radiografia, tac, ressonància, si s'escau i lliurar-lo al personal d'investigació.

9.5. Avaluació heurística

Introducció

Les persones que faran servir l'aplicació i avaluaran el seu funcionament les anomenarem avaluadors. Un sol avaluador trobarà menys problemes que si l'aplicació l'avaluen més persones, perquè lògicament com més persones provin l'aplicació més errors trobaran. En general es recomana entre 3 i 5 avaluadors.

La persona que supervisa l'avaluació s'anomena observador. L'observador guia als avaluadors i els hi resol els dubtes que els hi sorgeixin sobre el funcionament de l'aplicació. La presència de l'observador és opcional. Té les avantatges que l'avaluació es durà a terme més ràpid i que reduirà el treball dels avaluadors, ja que en cas de dubte de funcionament de l'aplicació no perdran temps i no caldrà que omplin l'informe ells, doncs s'encarregarà l'observador. En canvi suposa un cost addicional.

Al nostre cas hi haurà un (o dos) avaluador que serà el representant (o representants) de la UDIAT i un observador que serà l'equip de disseny que ens encarregarem de guiar l'avaluació del procés i apuntar els inconvenients a l'annex 7.4 a mesura que es vagin trobant.

L'avaluador haurà de seguir els següents casos d'ús:

1. Consultar una imatge mèdica d'un pacient determinat.
2. Modificar diferents aspectes essencials d'una imatge mèdica, afegir o treure les sèries que es mostren, canviar de zona, etc.
3. Diagnosticar i desar els resultats.

Cada problema que es trobi s'ha d'apuntar per separat (és important considerar els errors individualment) i valorar la seva gravetat en una escala del 0 al 4, on el significat numèric és el següent:

0. No és un problema d'usabilitat.
 1. És un problema estètic.
 2. És un problema menor.
 3. És un problema important.
 4. És un problema molt greu.

La puntuació es pot fer primer individualment i després en grup. Com el nombre d'avaluadors serà molt limitat, aquest aspecte és superflu.

Finalment es revisen els resultats amb l'equip de disseny, els avaluadors i els observadors i es decideix, tenint en compte la gravetat i el cost d'arreglar cada problema, com es redissenyarà l'aplicació.

Procés

- Explicació als avaluadors del funcionament de l'avaluació heurística que s'ha preparat i del prototip de la *tauleta* amb l'aplicació, ja que l'avaluació es durà a terme sobre el prototip.
- Avaluació de l'aplicació.
- Classificació dels errors segons la gravetat
- Revisió amb l'equip de disseny, els avaluadors i els observadors.

10 heurístiques de Jacob Nielsen

- 11) Estat del sistema: Sempre s'informa a l'usuari del que està passant. Per exemple, si s'està carregant una imatge mostrar algun indicador (barra de càrrega, cercle de càrrega...) que mostri que s'està fent una acció i no s'ha quedat el sistema penjat.
- 12) Familiaritat amb el llenguatge de l'aplicació: Hi ha d'haver una relació entre el llenguatge que utilitza l'aplicació i el llenguatge natural. L'usuari ha d'entendre els missatges del sistema, per tant aquest ha d'utilitzar un llenguatge conegut i familiar.
- 13) Llibertat i control per part de l'usuari: L'usuari ha de poder desfer errors fàcilment i tenir el control sobre el que vol fer amb l'aplicació, no a l'inrevés.
- 14) Estàndards i consistència: No hi pot haver ambigüitats i l'aplicació ha de fer servir els estàndards.
- 15) Prevenició d'errors: L'aplicació ha d'afavorir que no es produeixin errors. De vegades és bo mostrar missatges de confirmació per evitar errors greus.
- 16) Evitar que l'usuari hagi de memoritzar o recordar: Procurar que a la mateixa pantalla aparegui tota la informació necessària i que l'usuari no hagi de recordar el que posava a una pantalla anterior.
- 17) Flexibilitat i eficiència: Permetre que els usuaris experts puguin avançar més ràpidament gràcies a que el sistema té mecanismes més complexos que requereixen experiència, però alhora permetre que els usuaris principiants no tinguin problemes per fer servir l'aplicació amb fluïdesa.
- 18) Estètica i minimalisme: No hi ha d'haver informació irrellevant a la pantalla i la informació ha de ser clara i concisa, sense elements superflus.
- 19) Reconèixer, diagnosticar i recuperar errors: Els missatges d'error han de ser expressats en un llenguatge amè indicant amb precisió el problema i suggerint una solució.
- 20) Ajuda i documentació: Tot i que l'aplicació ha de ser intuïtiva per a l'usuari, s'ha de proporcionar ajuda en el cas que l'usuari la requereixi. Ha de ser concisa, ràpida de trobar i entenedora.

9.6. Full d'avaluació heurística

Problema:

Gravetat:

Heurística violada:

Descripció:

CAPÍTOL 10

ABSTRACT

Aquest treball tracta sobre la realització d'una aplicació per a dispositius Android que permeti visualitzar imatges mèdiques.

Amb la col·laboració del parc Taulí de Sabadell s'han realitzat tasques de recerca de requeriments com l'elaboració d'una entrevista o la realització d'una maqueta de paper que s'ha sotmès a una avaluació heurística en un procés iteratiu.

Finalment, s'ha implementat una aplicació on es poden visualitzar imatges mèdiques, fer accions sobre elles com canviar el color o fer zoom i accedir a informació del pacient.

Este trabajo trata sobre la realización de una aplicación para dispositivos Android que permita visualizar imágenes médicas.

Con la colaboración del parc Taulí de Sabadell se han realizado tareas de búsqueda de necesidades como la elaboración de una entrevista o la realización de una maqueta de papel que se ha sometido a una evaluación heurística en un proceso iterativo.

Finalmente, se ha implementado una aplicación donde se pueden visualizar imágenes médicas, hacer acciones sobre ellas como cambiar el color o hacer zoom y acceder a información del paciente.

This project is about making an Android application for mobile devices that allows viewing medical images.

Needfinding tasks were made with the collaboration of parc Taulí de Sabadell, such preparing and doing an interview or making a paper prototype that has been subjected to a heuristic evaluation in an iterative process.

Finally, we have implemented an application where you can view medical images, perform actions on them as changing the color or zoom in and access patient information.