TREBALL FINAL DE MÁSTER

Raül Broto Cervera

Consultor: Ignasi Lorente Puchades
Data d'entrega: 9/1/2015

INDEX

Apartat	Pàgina
1 – Introducció	3
Descripció del Projecte	4
Objectius	5
Calendari del Projecte	6
Documentació del projecte	9
Tecnologia utilitzada	10
Riscos del projecte	11
2 – Disseny	12
Llistat de funcionalitats	13
Usuaris i context d'ús	15
Disseny conceptual	21
Prototipatge	25
Avaluació	32
3 – Disseny Tècnic	35
Base de dades	36
Disseny software aplicació	37
Arquitectura del Sistema	40
4 - Desenvolupament	42
Eines utilitzades	43
Aplicació mobile	45
Aplicació wear	48
5 – Proves	51
Procés de proves	52
Proves unitàries	53
6 - Conclusions	54
Propers passos	57
7 - Annex	58
Bibliografia	58

1) INTRODUCCIÓ

Actualment, estem vivint una evolució trepidant de molts aspectes tecnològics de la nostra vida quotidiana. Però sense dubte, durant els últims anys hi ha un camp que destaca per sobre dels altres, l'anomenat "*Internet of Things"* (IoT). En els propers anys ens espera un gran salt qualitatiu pel que fa a *"Wearables", "iHealth", "SmartHome"*, etc., ja que la indústria tecnològica hi veu un gran mercat, i nosaltres els usuaris, cada cop estem més habituats a tractar amb múltiples dispositius en el nostre dia a dia.

Dins de l'IoT hi ha dos camps que em semblen molt interessants: l'SmartHome per vocació professional (ho comentaré amb més detall al proper apartat) i els wearables per l'espectacularitat del que poden arribar a ser i fer. Durant el passat any 2014 hem viscut el tret de sortida dels wearables per al gran públic, i gran part de la culpa és de Google i els dispositius amb Android Wear. Per aquest motiu, em va semblar una gran oportunitat aprofitar aquest TFM per a aprofundir en aquest ecosistema, ja sigui per la seva utilitat en la meva futura carrera professional o simplement per curiositat. El que està clar, és que els wearables, i més concretament Android Wear, són el futur (un futur no gaire llunyà, m'atreviria a dir).

A més, la realització d'aquest treball de final de Màster també presenta moltes altres oportunitats per a augmentar els meus coneixements i "*expertise*" en altres àrees. D'una banda, el clar enfocament a la usabilitat del disseny, pot millorar la meva habilitat en quant al disseny d'aplicacions, tant pel que fa al propi disseny gràfic com a la implementació amb codi de la interfície d'usuari. També m'ha permès endinsar-me tímidament en una tecnologia que és el futur proper, el reconeixement de veu. I tampoc cal oblidar l'experiència relacionada amb el "*management*" del projecte, ja que pot ser molt important en el meu futur professional.

I finalment, i per a mi el més important, és que la realització d'aquest projecte ha estat un gran repte. El fet d'enfrontar-me en un període de temps tant curt a una tecnologia desconeguda i assolir un resultat exitós des del meu punt de vista, ha estat una grandíssima satisfacció. Crec que aquesta experiència em pot ser molt útil en el futur, per a confiar en les meves possibilitats, habilitats i per a prendre decisions valentes en properes ocasions.

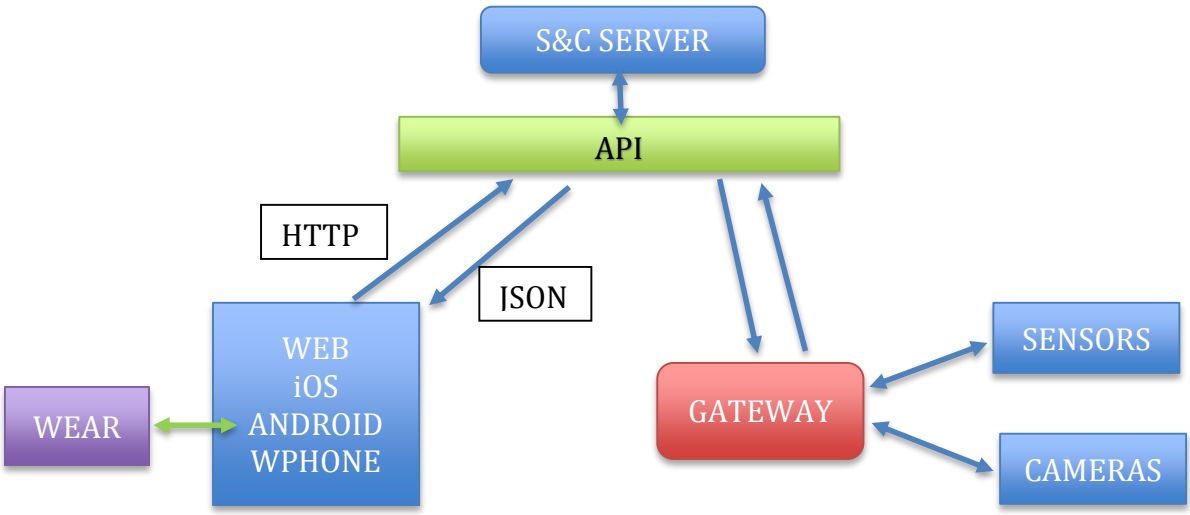
Descripció del Projecte

L'objectiu final del projecte és la creació de l'aplicació "enControl Wear", és a dir, l'extensió de l'aplicació enControl ja existent a la nova plataforma Android Wear. Però primerament, per a entendre correctament tot el que es pretén fer en aquest projecte, són necessàries unes quantes explicacions prèvies.

Actualment treballo com a desenvolupador de software a l'empresa Sensing&Control, start-up amb seu a Barcelona. El producte principal de l'empresa (tot i que encara es troba en la fase final del seu desenvolupament) és la plataforma enControl, que consisteix en una plataforma de SmartHome des de la qual pots controlar/visualitzar diversos actuadors/paràmetres de casa remotament (o d'una instal·lació si pensem en un cas més genèric). En línies molt generals, una explicació de la plataforma es pot trobar en el següent vídeo de Youtube: https://www.youtube.com/watch?v=qSYzRE9t-sw#t=192.

Aquesta plataforma consisteix a grans trets en una sèrie de serveis corrent al servidor de Sensing&Control que anomenem "*backend*", a partir dels quals les plataformes Web o mòbil (Android, iOS i Windows Phone) es comuniquen mitjançant una API i els seus *"endpoints"*, per a obtenir dades o realitzar peticions de canvis d'estat de dispositius. El propi backend es comunica (a petició de l'usuari o en alguns casos periòdicament) amb el Gateway que l'usuari té a casa, el qual fa de pont amb el dispositius (Sensors i Càmeres) que es vulguin accionar.

A continuació podem observar un esquema simplificat de l'arquitectura de la plataforma:



En el meu cas, jo m'encarrego de desenvolupar l'aplicació d'Android (entre d'altres coses), amb la qual pots realitzar gairebé les mateixes accions que a la web. Per exemple, és possible encendre/apagar llums, pujar el termòstat, activar/desactivar l´alarma, planificar accions en un scheduler, rebre notificacions de canvis d´estat, entre moltes d´altres. És per això que se'm va acudir la idea de realitzar una ampliació d´aquestes funcions a un dispositiu Android Wear, és a dir, que puguis encendre/apagar llums, posar l´alarma, rebre notificacions, etc. directament des del rellotge.

Per tant, aquest projecte es centra únicament en realitzar l'aplicació per a Android Wear, utilitzant les eines ja existents de la plataforma enControl (l'API i l'aplicació Android principalment).

Objectius

Els objectius que es pretenien assolir mitjançant la realització d'aquest projecte són:

- Conèixer l'entorn de programació d'Android Wear, i adquirir els coneixements necessaris per a ser capaç de realitzar una aplicació totalment funcional.

- Ser capaç d'enfrontar-me a una tecnologia desconeguda, obtenint un resultat positiu al final del projecte.

- Millorar les meves habilitats referents al disseny d'aplicacions. Aquest punt engloba tant el disseny previ de l'aplicació (estructura, components, *"flow process"*, usabilitat, etc.), com el propi disseny gràfic i amb codi de la interfície d'usuari.

- Conèixer, tot i que sense entrar en profunditat, noves tecnologies punteres (reconeixement de veu, "*geofencing"*, per exemple).

- Augmentar les capacitats de "*management"* d'un projecte, i realitzar correctament cadascuna de les fases que el composen.

Calendari del Projecte

En primer lloc, crec que és necessari presentar la calendarització tal i com va ser realitzada durant el pla de treball, per a poder posteriorment (al final de la secció) valorar el seguiment i compliment d'aquest durant la realització del projecte.

Estimació inicial del Calendari del Projecte (Pla de Treball)

En quant al calendari del projecte, és evident que la seva temporització està directament relacionat amb el corresponent calendari d'entregues (PACs i Memòria Final). A la pàgina 7 es pot observar el diagrama de Gantt amb la temporització completa.

Per tant, la primera tasca (PAC1) i l'última (Memòria Final + Vídeo presentació) estaran condicionades per la realització d'ambdós documents complint el corresponent termini d'entrega (al gràfic els lliurables apareixen com un rombe que significa "*milestone*"). La segona fase, corresponent a la PAC2, s'ha dividit en tres tasques, cadascuna relacionada amb el lliurable que s'haurà d'entregar. És per això que s'ha establert tres activitats diferents, cadascuna amb el seu propi termini.

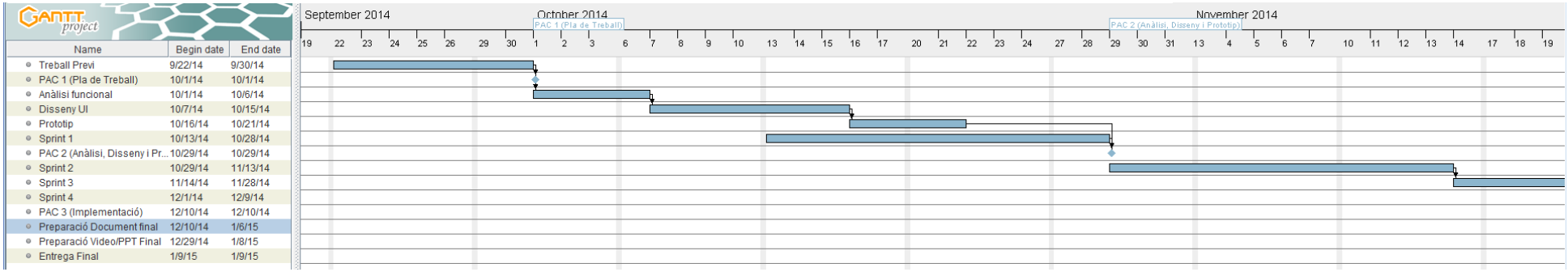
Finalment, la fase d'implementació (PAC3) que és la més llarga i important de tot el projecte estarà basada en un aspecte que em sembla molt interessant de la metodologia SCRUM, el ***desenvolupament incremental***. Aquesta metodologia consisteix en que en comptes de tenir calendaritzat des de l'inici del projecte cadascuna de les funcionalitats a realitzar, aquestes es van planificant en fases més petites anomenades "*sprints*", i al final de cada sprint es fa una valoració del treball realitzat, de manera que per al següent és més fàcil ajustar la relació temps/treball. En aquest projecte, em sembla molt adient el seu ús, ja que a priori no puc assegurar la dificultat i el temps que comportarà cada tasca, i per tant, mitjançant els sprints serà molt més fàcil adaptar els imprevistos que puguin anar sorgint.

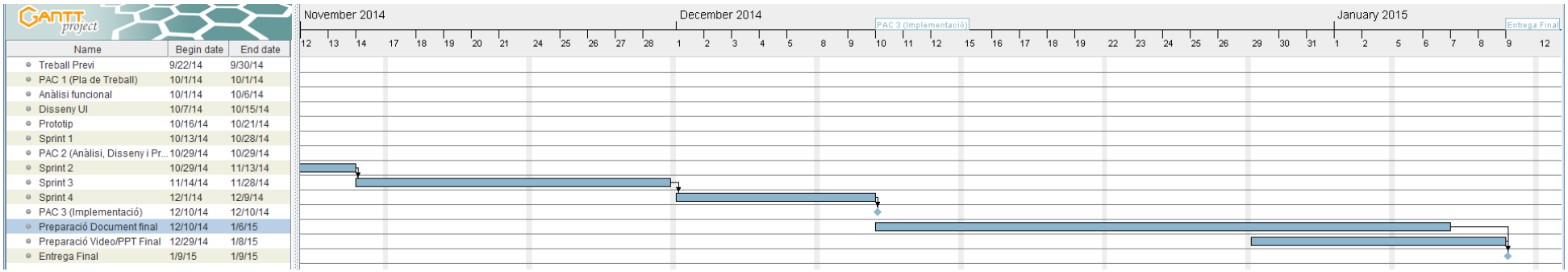
Per a implementar-ho, he separat el temps corresponent a aquesta fase en 4 sprints de 15 dies aproximadament. Al final de cada sprint realitzaré un testeig i valoració del treball realitzat, i això em permetrà ajustar la feina de cara a l'sprint següent. Les funcionalitats previstes per al primer sprint són totes les corresponents al grup de funcionalitats bàsiques:

- Rebre notificacions al rellotge.
- Implementar actuacions/visualitzacions a partir de la seva recepció.
- Descartar (o marcar com a llegida) la notificació.
- Visualització de la temperatura.
- Visualització de l'estat d'alarma.
- Visualització de l'estat dels sensors
- Actuació d'encendre/apagar interruptors.
- Actuació per connectar l'alarma de seguretat.
- Actuació per a controlar la temperatura del termòstat.

El seguiment del sprints es realitzarà mitjançant les Visual Studio Tools (degut a que ja les he utilitzat anteriorment), les quals permeten portar un control sobre les tasques realitzades o pendents per a cada sprint, i que ofereix diversos paràmetres útils per a la documentació, com ara la velocitat (relació entre el temps emprat i la feina realitzada) o el gràfic de "*burndown*" (gràfic resum respecte a la feina realitzada durant l'sprint en comparació amb la velocitat ideal).

A continuació, presento el diagrama de Gantt corresponent al calendari inicial del projecte:





Valoració del Seguiment del Calendari inicial del Projecte

Pel que fa a la valoració del seguiment del calendari inicial del projecte, estic prou satisfet amb la temporització portada a terme, ja que si bé és cert que no ha sobrat temps per diversos motius, he pogut realitzar totes les parts importants i imprescindibles dins dels terminis marcats.

En primer lloc, vull destacar com a factor positiu el fet de tenir els “milestones” (PACs), ja que marcaven punts d’inflexió clars i em motivaven per al compliment de les tasques en el temps previst.

Pel que fa a les PAC1 i PAC2 van ser completades satisfactòriament en el termini previst, i fins i tot en el cas de la PAC2 em van sobrar uns pocs dies que vaig aprofitar per a practicar amb l’entorn d’Android Studio i realitzar algunes aplicacions de prova d’Android Wear. Aquests dies “extres” considero que van ser molt positius, ja que van fer que en el moment que va començar l’sprint 1 ja estigués acostumat a l’entorn, i per tant, em vaig dedicar exclusivament al desenvolupament de l’aplicació.

Pel que fa a la PAC3, cal dir que vaig trobar problemes en algunes parts de la implementació, fet que va fer que el temps s’ajustés més. Per aquest motiu, va ser necessari una reducció del temps dedicat a les proves unitàries, per exemple. Un altre efecte d’aquest ajustament, és que la part de notificacions no va quedar totalment enllestida (sí que era funcional, però no tot lo fluida que desitjava). Per aquest motiu, vaig utilitzar algun dia dels destinats a la redacció de la Memòria Final. Respecte a aquesta PAC, també voldria destacar la idoneïtat d’emprar SCRUM i el seu desenvolupament incremental. Això em va ajudar a no quedar-me encallat quan no em sortia una funcionalitat, i a la vegada, a anar perfeccionant-les a cada sprint.

Finalment, un canvi menor però que no ha afectat al projecte, és que finalment vaig marxar de vacances al Nadal, fet que va provocar una major càrrega de treball abans del dia 19 de desembre i posteriorment a partir del dia 2 de gener, però que van ser assolides sense cap problema.

Documentació del projecte

Durant la realització del projecte, s'han documentat les seves diverses parts en un conjunt de documents, els quals coincideixen amb les diverses entregues del curs (PACs i Memòria final).

En primer lloc, es va entregar un pla de treball corresponent a la PAC1, on s'explica en línies generals una idea general del conjunt del projecte. A continuació, per a la PAC2 es van realitzar diversos entregables enfocats principalment al disseny de l'aplicació, centrant-nos principalment en la usabilitat. Aquesta entrega estava composada per un disseny funcional, un disseny tècnic i un prototip amb la interfície d'usuari que va servir d'orientació per al disseny de l'aplicació. A partir d'aquest punt, va començar la implementació del projecte (PAC3), de la qual es va lliurar el codi i els corresponents instal·lables.

Per acabar, es lliura aquesta Memòria Final que engloba tota la documentació del projecte, fent que les diverses parts abans esmentades formin part d'un conjunt global que documenti tot el projecte. Per tant, a més d'afegir els documents de les PACs (convenientment modificats), s'han afegit elements importants en apartats globals, com ara la introducció o les conclusions.

Finalment, en els propers dies també s'entregarà un vídeo de demostració del funcionament real de l'aplicació, afortunadament en un dispositiu real (Moto 360) adquirit recentment.

Tecnologia utilitzada

A continuació presento un llistat amb la tecnologia que s'ha utilitzat durant el projecte:

- **Android Wear:** Ecosistema on s'executa l'aplicació.

- **Android Studio:** Software editor de codi, molt recomanat per al desenvolupament d'aplicacions per a Android Wear (en comptes del clàssic Eclipse).

- **Llenguatge Java, adaptat per Android:** Llenguatge de programació.

- **Team Foundation Server (TFS):** Repositori de codi de Windows. L'utilitzo diàriament, així que és amb el que em sento més còmode.

- **Visual Studio Tools:** Eina online per a controlar els "*commits"* de codi al TFS, a més dels diversos gràfics per a controlar l'estat dels Sprints de SCRUM.

- **API enControl:** S'ha utilitzat l'API existent de la plataforma enControl, la qual està formada per diversos endpoints. Mitjançant crides HTTP (GET, POST, PUT, DELETE i PATCH), s'obtenen les dades desitjades o es realitzen les peticions d'acció demanades per l'usuari.

- **Hardware Instal·lacions Sensing&Control:** S'ha utilitzat hardware existent a les instal·lacions de Sensing&Control com ara sensors, gateways, càmeres, etc.

Riscos del projecte

Des del meu punt de vista, aquest projecte comptava amb un nivell de risc força elevat, ja que és una tecnologia desconeguda i relativament nova, i per tant, a priori no podia saber el nivell de dificultat podien tenir les tasques, i fins i tot, podia ser que alguna de les funcionalitats desitjades fos impossible d'implementar (tal i com ha passat amb els elements relacionats amb GPS, per exemple).

Directament relacionat amb aquest aspecte, trobo que un altre gran risc del projecte és el temps. Degut a que no podia calcular, ni tan sols aproximadament, el temps que m'ocuparia cada tasca, era possible que finalment no es poguessin realitzar algunes funcionalitats desitjades per manca de temps abans de l'entrega.

Tot i això, voldria remarcar que aquests riscos em van servir a la vegada com a repte i motivació.

A continuació, presento una descripció més esquemàtica dels riscos, per a poder comparar-los més fàcilment amb els requeriments:

Risc	Importància	Probabilitat	Afectació
Manca de temps	Alta	Alta	Greu
Funcionalitat bàsica impossible d'implementar	Alta	Mitja	Greu
Funcionalitat mitja impossible d'implementar	Mitja	Mitja	Lleu
Funcionalitat avançada impossible d'implementar	Baixa	Alta	Lleu

2) DISSENY

En aquest apartat està documentada una de les parts més importants del projecte, el seu disseny. En ocasions, els programadors cometem l'error d'anar massa de pressa i voler començar a escriure codi ràpidament, fet que pot portar a problemes posteriors durant el desenvolupament o quan els usuaris proven l'aplicació. En canvi, un bon disseny que identifiqui el que realment volen/necessiten els usuaris, i que permeti tenir una idea global del que haurem d'implementar, fa que la fase de desenvolupament sigui molt més profitosa.

En aquest cas, el disseny s'ha centrat principalment en la usabilitat. Les aplicacions mòbils (en el meu cas d'aplicació wearable, encara amb més motiu) són un tipus d'aplicacions especials, i és evident que la usabilitat juga un paper clau: si l'aplicació no és usable els usuaris no la faran servir, ja que hi ha moltes alternatives al mercat. És per aquest motiu pel que li dono tanta importància a aquest apartat, perquè l'èxit de l'aplicació en depèn en gran mesura.

Més concretament, en aquest apartat dedicat al disseny hi ha, per exemple, el llistat de funcionalitats que crec l'aplicació hauria de tenir (és l'anàlisi a priori, per a poder analitzar posteriorment que s'ha pogut realitzar i que no). També trobem l'anàlisi dels usuaris i el context d'ús, per a poder determinar quin és el target group i en quines ocasions empraran l'aplicació. A partir d'aquest dos punts, s'ha realitzat també un disseny conceptual, que permet fer-se una idea global de l'aplicació mitjançant els escenaris i els fluxos d'interacció. Finalment, també podem veure el prototipatge inicial sorgit d'aquests anàlisis, així com el qüestionari d'avaluació dels dissenys.

Llistat de funcionalitats

A continuació presento un llistat amb les funcionalitats que es volien incloure inicialment en la realització del projecte. Crec necessari presentar la llista inicial, per a que sigui possible veure quines han estat realitzades i quines no (aquest anàlisi es pot trobar a la secció ***Estat actual del projecte*** de l'apartat ***6)Conclusions***). S'han dividit les possibles funcionalitats en tres grups, tenint en compte bàsicament dos aspectes: importància de la funcionalitat en el global del projecte i dificultat per a realitzar-la.

En aquest primer grup, apareixen funcionalitats que considero que són **bàsiques** per a obtenir el resultat de qualitat del projecte desitjat:

- Notificacions: Des del meu punt de vista, les notificacions són l'element clau d'aquest tipus de wearables. Al no tenir un ús autònom (són una extensió de l'smartphone), crec que la principal utilitat que tenen és facilitar l'arribada de notificacions a l'usuari. Per tant, les funcionalitats esperades respecte a aquest punt són:
    - Rebre notificacions al rellotge.
    - Implementar actuacions/visualitzacions a partir de la seva recepció.
    - Descartar (o marcar com a llegida) la notificació.

- Visualització: Un altre element indispensable és la visualització dels diversos paràmetres de la instal·lació des del rellotge, ja que l'usuari voldrà consultar aquesta informació directament des del rellotge. Per tant, les funcionalitats seran:
    - Visualització de la temperatura.
    - Visualització de l'estat d'alarma.
    - Visualització de l'estat dels sensors

- Actuació: Aquest punt es refereix a la possibilitat de poder realitzar accions als diversos dispositius mitjançant el wearable. Des del meu punt de vista, és un punt molt important de cara a l'usuari, ja que li permetrà realitzar accions immediates. Les funcionalitats respecte a aquest punt seran:
    - Actuació d'encendre/apagar interruptors.
    - Actuació per connectar l'alarma de seguretat.
    - Actuació per a controlar la temperatura del termòstat.

A partir d'aquestes funcionalitats indispensables, també tenim un altre grup de funcionalitats que també suposen molt atractives per a l'aplicació (cal tenir en compte que a priori no puc saber exactament la dificultat que implicarà cada funcionalitat). Per tant, **la seva implementació dependrà del temps disponible i la dificultat que comportin**.

- Visualització de fotografies: Les instal·lacions també poden disposar de càmeres IP, amb les quals es poden realitzar fotografies "on demand" o fotografies automàtiques quan salta l'alarma. Per tant, una funcionalitat interessant seria:
    - Visualitzar les fotografies directament al rellotge.

- "Geofencing": Aquest terme es refereix a accions automàtiques que es realitzen en un àrea geogràfica determinada. Per exemple, una funcionalitat podria ser:
    - Connectar l'alarma quan es surt de casa.
    - Desconnectar l'alarma quan s'arriba a casa.

- Reconeixement de Veu: Aquest punt potser és massa ambiciós, però crec que seria una magnífica funcionalitat poder realitzar accions a partir de la interacció amb el rellotge mitjançant veu. Una funcionalitat que implementi aquest punt podria ser:
    - Descartar (marcar com a llegides) les notificacions.

Finalment, a l'últim grup tenim un llistat de funcionalitats que exceptuant un cas excepcional (sobretot si finalment la dificultat és menor de l'esperada), **no tindran cabuda dins l'abast del projecte,** però que sens dubte poden ser interessants com a futures funcionalitats:

- Visualització de Vídeo: Les càmeres IP comentades anteriorment, també proporcionen vídeo "on demand", per tant, tot i que considero que no és tan prioritari com veure les fotografies, podria ser interessant fer arribar aquest vídeo al rellotge. Per tant:
    - Visualització de vídeo streaming al rellotge.

- Ampliació funcionalitats de Reconeixement de Veu: Consisteix en ampliar la funcionalitat descrita anteriorment amb noves funcionalitats:
    - Actuació mitjançant veu per a d'encendre/apagar interruptors.
    - Actuació mitjançant veu per connectar l'alarma de seguretat.
    - Actuació mitjançant veu per a controlar la temperatura del termòstat.

- Visualització de gràfics amb dades històriques dels sensors: Tot i que aquesta funcionalitat potser no té gaire sentit realitzar-la al wearable (pantalla massa petita), es podria trobar alguna manera d'adaptar el disseny per a poder-ho mostrar correctament. Per tant, una funcionalitat interessant seria:
    - Visualització de l'històric de consum del sensors

- "HomeFencing": Es refereix al mateix que el "Geofencing" però encara més concret, és a dir, a nivell de l'interior de casa. Per exemple, l'entrada a una habitació podria encendre la llum. En aquest cas, ho considero 99% irrealitzable, ja que també s'haurien d'integrar elements com ara iBeacons o NFC. La funcionalitat associada seria:
    - Implementar l'acció d'encendre/apagar la llum quan l'usuari entra/surt d'una habitació.

Usuaris i context d'ús

Abans d'exposar els aspectes relacionats amb els mètodes d'indagació i d'identificar els perfils d'usuari, crec que és necessari avaluar la situació general del meu projecte respecte als usuaris i el context d'ús, ja que es tracta d'un cas molt específic i avançat tecnològicament, i crec que per això mereix un tracte especial.

Des del meu punt de vista, cal separar dues casuístiques molt diferents. D'una banda, tenim la tecnologia emprada per a la realització de l'aplicació, Android Wear. Aquesta tecnologia (els wearables en general m'atreviria a dir) es troba en un estat molt inicial, i per tant, és evident que tot i que intentem arribar al màxim d'usuaris possible proporcionant-los la màxima usabilitat, es produeix un xoc cultural i d'aprenentatge que trigarà anys en adaptar-se.

Google és molt conscient d'aquest tema i per això proporciona eines i guies estrictes que els desenvolupadors han de seguir per a intentar millorar la usabilitat de les aplicacions, i intentar que aquestes arribin al màxim de gent possible. Tot i això, el perfil d'usuaris que utilitzaran l'aplicació és molt clar (encara més clar que en el cas d'aplicacions mòbils per a smartphones, l'ús de les quals ja s'està estenent a usuaris més generalistes). El perfil d'usuari d'Android Wear compleix dos aspectes clars:

- ***Experimentat amb les noves tecnologies***. És clar que al ser un producte tan innovador, el target group seran usuaris experimentats amb les noves tecnologies i oberts a nous productes. Això descarta els grups d'edat avançada o gent jove/adulta reticent als canvis.

- ***Poder adquisitiu alt***. El preu dels propis smartwatch, unit a que necessites un smartphone de gama mitja-alta per a controlar-lo, fa que sigui una gran despesa que moltes persones no poden assumir.

D'altra banda tenim la pròpia plataforma sobre la que funcionarà l'aplicació, la plataforma de smarthome enControl. En aquest cas, la meva empresa ja ha realitzat els estudis adequats, i el resultat és la plataforma existent (tot i que encara es troba en fase alfa, i subjecta a canvis constants). Això implica que en aquest projecte podré modificar alguns petits aspectes per a adaptar-los a la tecnologia wearable, però no podré fer grans canvis, ja que moltes funcionalitats ja venen donades de la plataforma "mare".

Per tant, durant aquesta fase del projecte em centraré en indagar sobre quan els usuaris consideren que podrien fer ús de l'aplicació al seu rellotge i en les funcionalitats que els hi agradaria tenir disponibles. La indagació es realitzarà entre els usuaris participants en les proves pilots, ja que coneixen el funcionament de la plataforma i això farà que puguin centrar-se exclusivament en l'aplicació per a Android Wear. Per aquests motius, queden fora del focus principal de l'estudi: el funcionament de la pròpia aplicació (marcat per les línies de disseny que proporciona Google) i la plataforma enControl (aquest estudi es realitza a un nivell superior, i no forma part d'aquest projecte).

Mètodes d'indagació

Un cop comentades les idees generals per aquesta fase, és el moment de realitzar la primera investigació amb potencials usuaris, per a poder identificar els seus perfils, requeriments i el context d'ús de l'aplicació. De tots els mètodes disponibles, he escollit la realització d'enquestes.

El principal motiu és que tal i com he comentat a la introducció d'aquesta secció, crec que la indagació s'ha de realitzar sobre usuaris de les proves pilot de la plataforma enControl, ja que ells ja coneixen l'entorn, i per tant, em podran ajudar millor a definir el que volen els futurs usuaris de l'aplicació d'Android Wear. Les enquestes permetran obtenir una millor informació de la seva opinió sobre quan podrien utilitzar l'aplicació, i les principals funcions de la plataforma existent que els hi agradaria tenir al smartwatch.

Aquesta enquesta s'ha realitzat sobre 5 usuaris de les proves pilots. He intentat abastar el màxim de perfils possibles entre els 5 usuaris, per a poder tenir una visió més global del que volen. Per això, he escollit 2 usuaris de menys de 20 anys, 2 d'entre 30-40 anys i 1 usuari de 40-50 anys. L'enquesta està pensada principalment per a obtenir un llistat de funcionalitats a implementar a l'aplicació, per això, és curta però amb respostes concretes i directes:

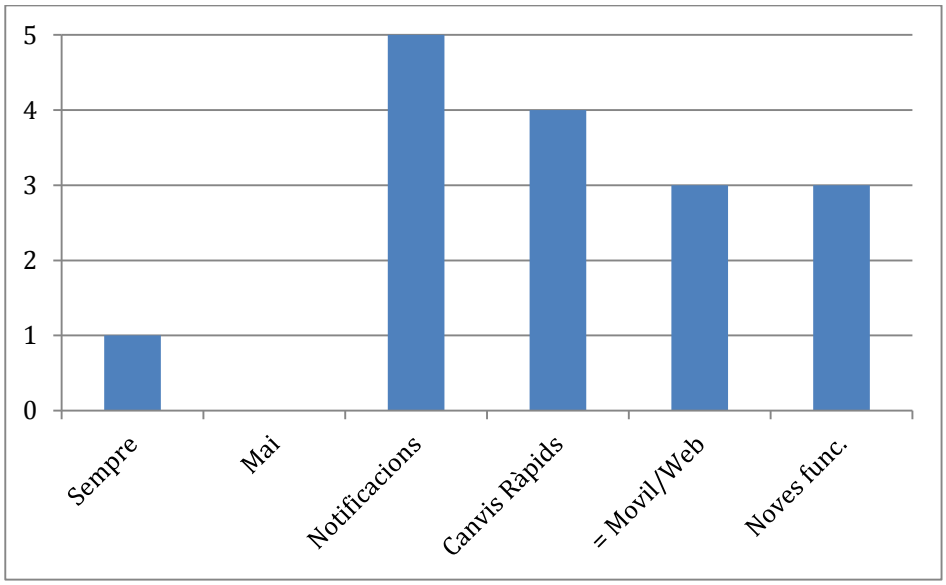
1) Quan creus que pot ser útil l'aplicació del rellotge?

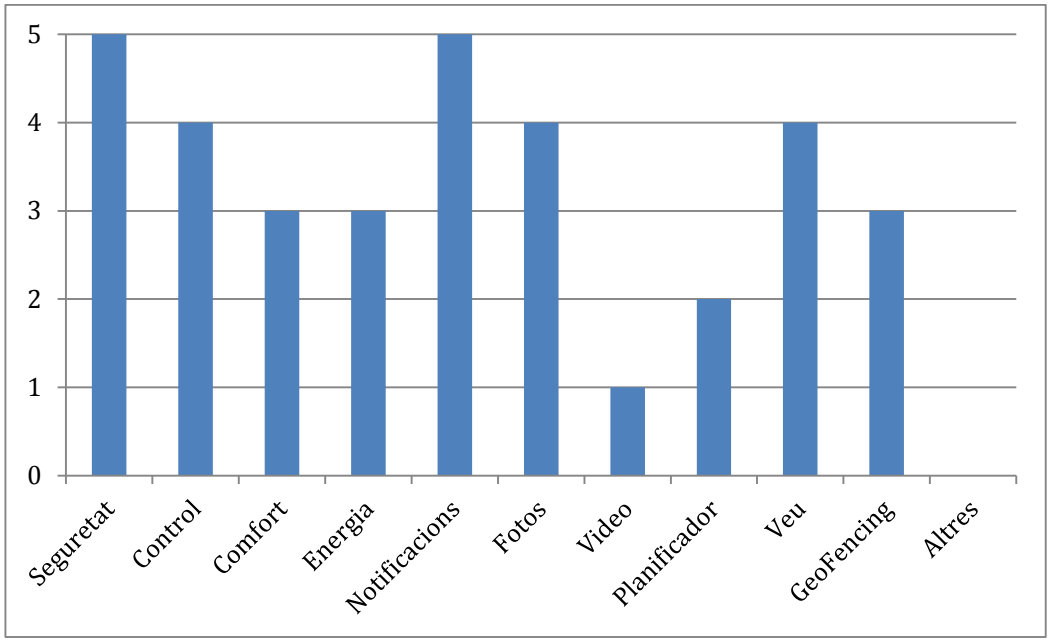
- Sempre
- Mai
- Per a veure notificacions
- Per a fer canvis ràpid
- El mateix que l'smartphone o la web
- Aplicació de noves funcionalitats

2) Quines funcionalitats existents a web i aplicacions mòbils t'agradaria disposar al rellotge?

- Seguretat
- Control (ON/OFF interruptors)
- Confort
- Energia (Visualització del consum)
- Notificacions
- Fotografies
- Vídeo
- Planificador
- Reconeixement de veu
- GeoFencing
- Altres

Els resultats obtinguts amb l'enquesta han estat:





Si analitzem els resultats obtinguts a partir de les dues preguntes, podem veure que aquests són els esperats. De la primera pregunta es pot extreure el possible ús futur que realitzaran els usuaris, ja que les respostes han fugit dels extrems (sempre/mai) i s'han ubicat en accions ràpides com ara veure les notificacions o realitzar canvis ràpids. Aquest fet entra dins de la lògica esperada, ja que si els usuaris pretenen fer alguna acció més complexa sempre tenen la possibilitat d'utilitzar l'smartphone o l'ordinador personal, però el rellotge els hi pot servir per a veure el motiu d'una notificació sense haver de treure el mòbil de la butxaca.

Pel que fa a la segona pregunta, les respostes estan una mica més repartides, tot i que podem veure quatre accions molt valorades pels usuaris i dues que no ho són. D'una banda, veiem que per a la mostra d'usuaris serà important que l'aplicació d'Android Wear implementi les funcionalitats de seguretat, notificacions, control i veu. Penso que el motiu principal és que aquestes accions són les més utilitzades al llarg del dia, i tenir-les més a mà pot ser beneficiós. Destacar l'opció del reconeixement de veu, ja que a l'aplicació mòbil es troba en un estat molt embrionari, però els usuaris han vist convenient incorporar-la al rellotge. L'altra cara de la moneda són les funcionalitats de vídeo i el planificador, els quals no són considerats necessaris pels usuaris. El principal motiu donat és la petita mida de la pantalla per als dos casos.

Per a concloure aquest apartat, cal remarcar els resultats obtinguts d'aquesta indagació, ja que han influït directament en la resta del projecte. Hem vist que els usuaris volen utilitzar l'aplicació per a realitzar tasques ràpides i immediates, i per a rebre les notificacions més a mà. Seguint les línies de disseny marcades per Google i les funcionalitats donades per la plataforma enControl s'han intentat assolir aquests objectius en la mesura del possible durant el disseny de les properes fases del projecte.

Perfils d'usuari

Per a l'anàlisi dels perfils d'usuari he realitzat un plantejament amb 3 perfils de grups d'usuaris concrets. Abans de separar els perfils ha estat necessari realitzar un filtre per a obtenir el conjunt d'usuaris futurs. He seguit bàsicament dues directrius bàsiques (comentades a la introducció de la secció) per als possibles usuaris d'Android Wear:

- *Experimentat amb les noves tecnologies*.
- *Poder adquisitiu alt*.

A partir d'aquí, he considerat bàsicament dos factors: edat i reticència a canvis. D'una banda, és evident que els joves (<30 anys) ja estem molt endinsats en les diverses tecnologies, i això ens dóna un plus en aspectes com la velocitat d'aprenentatge o el no tenir por a provar nous productes i tecnologies. Pel que fa als adults (<50 anys, ja que a partir d'aquí ja poden tenir problemes per a entendre-ho o problemes de visió fins i tot), he aplicat la separació extra de la reticència a canvis. Des del meu punt de vista hi ha certes persones que assoleixen qualsevol millora de la tecnologia o nou gadget ràpidament, ja que no tenen por als canvis i no els hi importa provar noves experiències. En canvi, considero que hi ha un altre perfil que està acostumat a fer servir els aparells actuals, però un canvi li pot arribar a provocar fins i tot por.

Perfil de grup d'usuaris 1: Joves < 30 anys

- *Factors demogràfics:*
  Els joves menors de 18 a 30 anys representen aproximadament entre un 15% i 20% de la població a Espanya.

- *Responsabilitat i tasques laborals:*
  En general es tracta d'estudiants o aturats (l'atur juvenil a Espanya és molt elevat). Els pocs que treballen tenen feines de perfil baix, com ara becaris o treballadors de baixa qualificació. En l'estudi que ens pertoca, ens centrarem en els joves amb poder adquisitiu mitjà-alt, pels motius comentats anteriorment.

- *Freqüència d'ús:*
  Els smartphones, internet i els PCs ja formem part de l'ús quotidià dels joves, utilitzant-los multitud de vegades durant el dia. Respecte a Android Wear específicament, es probable que mai l'hagin utilitzat, ja que la tecnologia porta poc temps al mercat.

- *Hardware:*
  Aquest grup es caracteritza per utilitzar tot tipus de dispositius, ja siguin els ordinadors de sobretaula, portàtils, com els dispositius mòbils.

- *Entorn:*
  L'entorn d'utilització del smartwatch serà el mateix que el d'un smartphone, és a dir, qualsevol lloc amb connexió de dades d'Internet. Tot i això, al controlar coses del propi domicili, és evident que hi haurà un ús més elevat a casa.

- *Experiència informàtica:*
  En línies generals alta. La generació digital ha nascut gairebé amb l'existència de la informàtica, i per tant, hi està acostumada.

- *Experiència amb aplicacions mòbils:*
  En quant a les aplicacions *mòbils*, també posseeixen un coneixement alt, ja que són usuaris habituals.

- *Coneixement de la tasca:*
  En quant al coneixement de la tasca, el valoraria com a mitjà, ja que és possible que sigui de les primeres vegades que utilitzen aquesta tecnologia, però està clar que degut a les similituds amb les aplicacions Android convencionals ho aprendran ràpid.

Perfil de grup d'usuaris 2: Adults entre 30 i 50 anys, NO reticents a canvis

- *Factors demogràfics:*
  Els adults entre 30 i 50 anys representen aproximadament entre un 20% i 25% de la població a Espanya.

- *Responsabilitat i tasques laborals:*

En general es tracta de persones amb ocupació laboral. Poden arribar a assolir llocs de responsabilitat i ben remunerats. En l'estudi que ens pertoca, ens centrarem en els adults amb poder adquisitiu mitjà-alt, pels motius comentats anteriorment.

- *Freqüència d'ús:*
  Els smartphones, internet i els PCs ja formem part de l'ús quotidià dels adults, utilitzant-los multitud de vegades durant el dia, tot i que potser menys freqüentment que els usuaris joves. Respecte a Android Wear específicament, es probable que mai l'hagin utilitzat, ja que la tecnologia porta poc temps al mercat.

- *Hardware:*
  Aquest grup es caracteritza per utilitzar tot tipus de dispositius, ja siguin els ordinadors de sobretaula, portàtils, com els dispositius mòbils.

- *Entorn:*
  L'entorn d'utilització del smartwatch serà el mateix que el d'un smartphone, és a dir, qualsevol lloc amb connexió de dades d'Internet. Tot i això, al controlar coses del propi domicili, és evident que hi haurà un ús més elevat a casa.

- *Experiència informàtica:*
  En línies generals alta o mitja-alta. Tot i rebre la tecnologia sent joves o adults, han sabut (o s'han hagut d'adaptar).

- *Experiència amb aplicacions mòbils:*
  En quant a les aplicacions mòbils, també posseeixen un coneixement alt o mitjà, ja que són usuaris habituals.

- *Coneixement de la tasca:*
  En quant al coneixement de la tasca, el valoraria com a mitjà, ja que és possible que sigui de les primeres vegades que utilitzen aquesta tecnologia, però està clar que degut a les similituds amb les aplicacions Android convencionals ho aprendran ràpid.

Perfil de grup d'usuaris 3: Adults entre 30 i 50 anys, SÍ reticents a canvis

- *Factors demogràfics:*
  Els adults entre 30 i 50 anys representen aproximadament entre un 20% i 25% de la població a Espanya.

- *Responsabilitat i tasques laborals:*
  En general es tracta de persones amb ocupació laboral. Poden arribar a assolir llocs de responsabilitat i ben remunerats. En l'estudi que ens pertoca, ens centrarem en els adults amb poder adquisitiu mitjà-alt, pels motius comentats anteriorment.

- *Freqüència d'ús:*
  Els smartphones, internet i els PCs ja formem part de l'ús quotidià dels adults, utilitzant-los multitud de vegades durant el dia, tot i que potser menys freqüentment que els usuaris joves. Respecte a Android Wear específicament, es probable que mai l'hagin utilitzat, ja que la tecnologia porta poc temps al mercat. Considero que el fet que siguin reticents a canvis, no implica una freqüència d'us menor.

- *Hardware:*
  Aquest grup es caracteritza per utilitzar tot tipus de dispositius, ja siguin els ordinadors de sobretaula, portàtils, com els dispositius mòbils.

- *Entorn:*
  L'entorn d'utilització del smartwatch serà el mateix que el d'un smartphone, és a dir, qualsevol lloc amb connexió de dades d'Internet. Tot i això, al controlar coses del propi domicili, és evident que hi haurà un ús més elevat a casa.

- *Experiència informàtica:*
  En línies generals mitja. Tot i rebre la tecnologia sent joves o adults, han sabut adaptar-se, però sense assolir un nivell d'expertise alt.

- *Experiència amb aplicacions mòbils:*
  En quant a les aplicacions mòbils, també posseeixen un coneixement mitjà, ja que són usuaris habituals.

- *Coneixement de la tasca:*
  En quant al coneixement de la tasca, el valoraria com a mitjà, ja que és possible que sigui de les primeres vegades que utilitzen aquesta tecnologia, però degut a les semblances amb les aplicacions Android podran utilitzar alguns dels models mentals existents per assolir-les.

Disseny conceptual

Pel que fa al disseny conceptual, aquest bàsicament ha consistit en la realització d'escenaris que es poden trobar els futurs usuaris (a partir dels perfils obtinguts a l'apartat anterior) i els corresponents fluxos d'interacció. Aquests dos apartats permeten fer-se una idea general de com serà el *"flow process"* general de l'aplicació, i per tant, considero que és un apartat de vital importància.

Escenaris

S'ha escollit tres escenaris (un per a cada perfil d'usuari), de manera que la dificultat d'aquests estigués relacionada amb el nivell de dificultat que aquests són capaços d'assolir. És a dir, per al tercer perfil (adults reticents a canvis) es plantejarà una tasca de dificultat més reduïda, de manera que ens assegurem que els usuaris són capaços de portar-la a terme. En canvi, el tercer escenari, pensat per als usuaris joves, tindrà un nivell de dificultat elevat en consonància amb el que aquest grup d'usuaris són capaços de fer.

Escenari 1: Josep, Adult 42 anys i reticent a canvis. Tasca: Desactivar l'alarma de casa.

En Josep acaba de comprar un pack d'enControl, una solució d'SmartHome recomanada per un amic, que li ha assegurat que és molt còmode controlar els dispositius de casa des de la web, l'smartphone o l'smartwatch que la seva dona li va regalar els Reis passats. Ell no hi està gaire convençut, ja que tot i que fa servir el mòbil diàriament, veu que els canvis aquestes noves tecnologies li venen una mica grans. Tot i això, decideix donar-li una oportunitat a veure si pot ser útil.

Avui a la feina ha estat provant la web, i s'ha sorprès de la senzillesa de la interfície però que a la vegada li ha permès realitzar les accions que ell volia (connectar l'alarma, ja que no hi havia ningú a casa). Per la tarda, quan arriba a casa vol provar les funcionalitats de l'aplicació que ha descarregat per al rellotge. Per tant, un cop entra per la porta decideix obrir l'aplicació al rellotge.

El primer que veu és la icona del programa, i un missatge que li indica que pot variar entre les opcions possibles fent "swipe" (moviment lateral típic a les pantalles tàctils). Prova el moviment i veu les diverses icones que hi ha presents a la pantalla principal de la web. Després de tocar una mica per a veure totes les opcions disponibles, es centra en la icona de seguretat i la clica per a observar que hi ha a continuació.

Després del clic la pantalla canvia cap a una altra on se li mostra l'estat actual d'alarma connectada ("Arm Away"). Clica sobre la icona per provar a veure si passa alguna cosa, però no passa res. Llavors pensa que potser amb un swipe veurà altres opcions. Ho prova i descobreix que a la següent pantalla apareixen el llistat de possibles estats de l'alarma ("Arm Away", "Arm Stay" i "Disarm"). Pitja sobre "Disarm" esperant que això desactivi l'alarma.

Finalment, després d'un o dos segons d'espera, rep una notificació al propi rellotge que li confirma que l'estat de l'alarma ara és "Disarm" (desactivat). En Josep, molt satisfet, decideix que ha estat una bona idea comprar enControl, i pensa que l'ha de seguir provant durant el futur per a descobrir totes les seves funcionalitats.

Escenari 2: Ana, Adult 38 anys i NO reticent a canvis. Tasca: Rebre notificació i realitzar acció.

L'Ana ja fa uns mesos que va comprar enControl, i està molt satisfeta de poder controlar remotament alguns aspectes de la casa, poder planificar tasques i veure el nivell d'energia que gasten. Els primers dies va tenir alguna dificultat a acostumar-se, però ella no té por d'anar tocant els botons fins a descobrir finalment el propòsit que tenen. Fa una setmana va comprar-se en nou LG G3 i es va sorprendre que pel mateix preu li venia el LG Smartwatch. Ella va

acceptar sense pensar-s'ho, amb moltes ganes de provar l'aplicació d'enControl per a Android Wear.

Després d'uns quants dies de prova per acostumar-se al nou aparell, n'està molt satisfeta. Moltes de les accions que realitzava amb l'smartphone, ara les pot fer directament des del rellotge, més ràpidament i més fàcilment. Tot i això, ella creu que hi ha algunes parts de l'aplicació que no ha acabat de provar del tot. És per això que quan li arriba una notificació d'alarma mentre estava a la feina es decideix a provar un camí nou.

Aquesta notificació d'alarma li indica que tenint l'alarma posada, el sensor de moviment ha notat presencia dins de casa. Una mica espantada, es decideix a buscar més informació sobre aquest esdeveniment. Ella recorda que en altres aplicacions d'Android Wear, quan reps una notificació i realitzes el moviment de swipe pots fer altres accions addicionals. Fa el swipe i veu com apareixen 3 accions addicionals:

- Descartar notificació.
- Canviar l'estat de l'alarma.
- Veure fotografies associades.

Molt contenta amb aquestes opcions, decideix primer veure les fotografies associades, per a saber el motiu que ha fet saltar el sensor. La seva sorpresa és que al rellotge apareix una fotografia (un pèl petita, però suficientment nítida) del seu fill i la cangur, que deuen haver arribat a casa fa poc i no la cangur no ha recordat desactivar l'alarma. Rep la notificació corresponent a que l'estat de l'alarma és "disarm" (desactivat). Més tranquil·la, torna a fer swipe cap a la dreta (moviment per a tornar enrere) i escull l'opció de canviar l'estat d'alarma. La desactiva i segueix treballant tranquil·lament i contenta del gran ús que està fent d'enControl.

Escenari 3: Lluís, Jove 24 anys. Tasca: Comandes de veu per apagar una llum.

En Lluís és un estudiant universitari d'enginyeria informàtica, i pel seu aniversari va rebre com a regal dels seus pares el Moto360, smartwatch que incorpora Android Wear. Ell és una persona que està molt al dia de totes les novetats tecnològiques, i per tant, està encantat amb el regal, ja que a més ha vist que la plataforma enControl que tenen a casa també disposa d'aplicació específica per a aquesta plataforma.

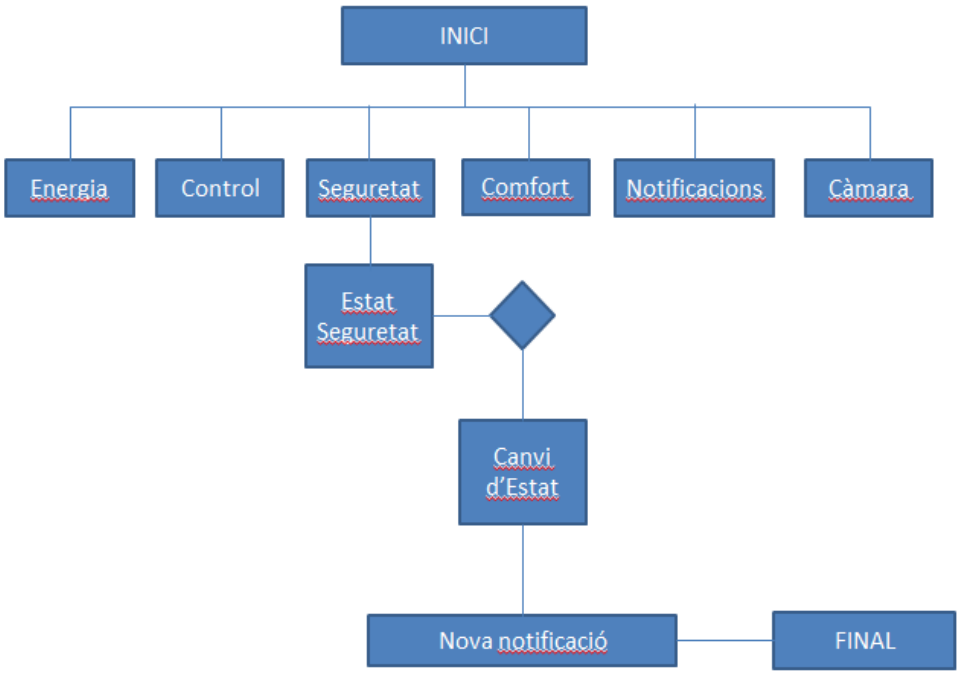
Un dia qualsevol, en Lluis espera a la porta de casa a la seva novia per anar a sopar plegats. Quan arriba, li diu que hi ha llum a la seva habitació, probablement se l'ha oblidat encesa. Per a comprovar-ho ràpidament, en Lluis obre l'aplicació al rellotge, va a la secció de Control i observa com l'interruptor corresponent a la seva llum de taula està encès.

Disposat a fer-se el fatxenda davant de la seva noia, es disposa a apagar la llum, però no pel sistema convencional (fent servir la pantalla del rellotge), sinó que utilitzant les comandes de veu. Per tant, diu la paraula clau que activa l'escolta activa de l'aplicació: primer "Ok, Google" i a continuació "enControl". Tot seguit, caldrà dir a l'aplicació el que ha de fer, i per tant, diu: "Control – Switch 2 – Off". La llum s'apaga en uns segons, i la noia es queda totalment bocabadada. A més, rep una notificació que l'informa del canvi d'estat de l'interruptor. En Lluís està molt satisfet.

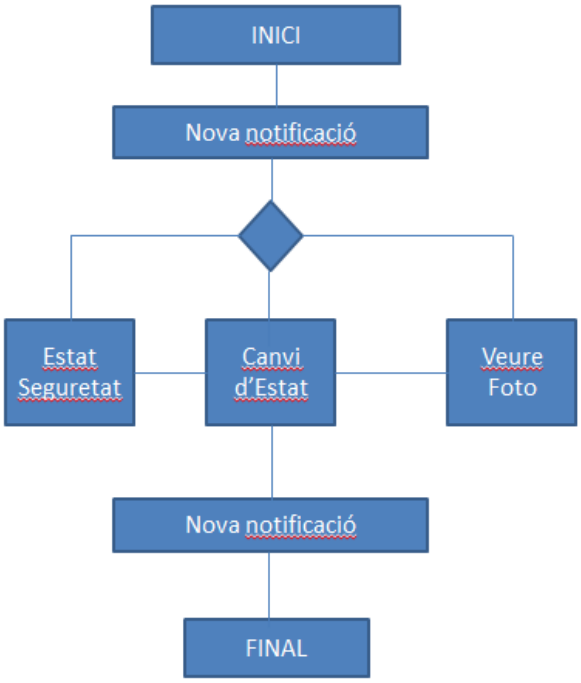
Fluxos d'interacció

L'objectiu d'aquest apartat és mostrar el flux d'interacció general de l'aplicació. Tot i això, crec que pot facilitar la comprensió d'aquest si primer observem cadascun del flux d'interacció dels tres escenaris, ja que al ser les tres tasques corresponent a les 3 principals tipus d'acció que es poden realitzar s'entendrà millor.

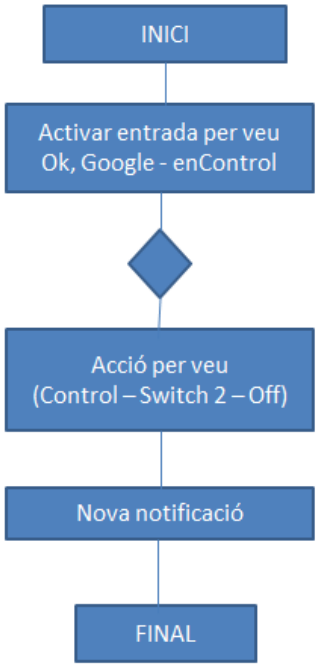
Escenari 1: Josep, Adult 42 anys i reticent a canvis. Tasca: Desactivar l'alarma de casa.



Escenari 2: Ana, Adult 38 anys i NO reticent a canvis. Tasca: Rebre notificació i realitzar acció.



Escenari 3: Lluís, Jove 24 anys. Tasca: Comandes de veu per apagar una llum.



Flux d'interacció general

El flux d'interacció general és simplement la conjunció d'aquests tres fluxos d'interacció que hem vist. L'aplicació tindrà tres grans blocs: la pròpia aplicació del rellotge, les notificacions i l'entrada de veu. El més destacable, és que les tres parts estan totalment connectades, i es pot anar canviant entre elles sense cap problema.

El bloc principal, que és l'aplicació a la que s'accedeix des del rellotge, tindrà en primer lloc una vista on s'escollirà entre cadascuna de les àrees a les que es vol accedir (seguretat, control, confort, energia, notificacions i càmeres). Un cop s'esculli àrea, per a cadascuna d'ella es podrà realitzar les accions oportunes (consulta de l'estat actual, canvi d'estat, etc.).

El segon bloc consisteix en les notificacions. L'usuari pot rebre múltiples notificacions, i quan les rebi, hem vist que voldrà realitzar accions immediates. Per això es disposarà directament d'un sistema "drecera" per a que a partir de la notificació es pugui, per exemple, canviar un estat sense tenir que obrir l'aplicació i recórrer totes les àrees fins a la desitjada.

Finalment, el tercer bloc correspon a les comandes de veu, que és bàsicament el mateix que fan els blocs anteriors però controlat per veu. En aquest cas, el flux d'interacció pot semblar més directe, però simplement es tracta de seguir mitjançant la veu un camí imaginari en els fluxos d'interacció anteriorment descrits.

Prototipatge

Pel que fa al prototipatge, és evident que té una gran importància la informació recollida durant les dues fases anteriors. Tot i això, tal i com he comentat anteriorment, en el cas d'Android Wear, Google ofereix una sèrie de recomanacions a seguir, per a que l'usuari pugui utilitzar els models mentals que ja té tant d'altres aplicacions del wearable, com fins i tot per les similituds que pot tenir amb les aplicacions Android d'smartphones.

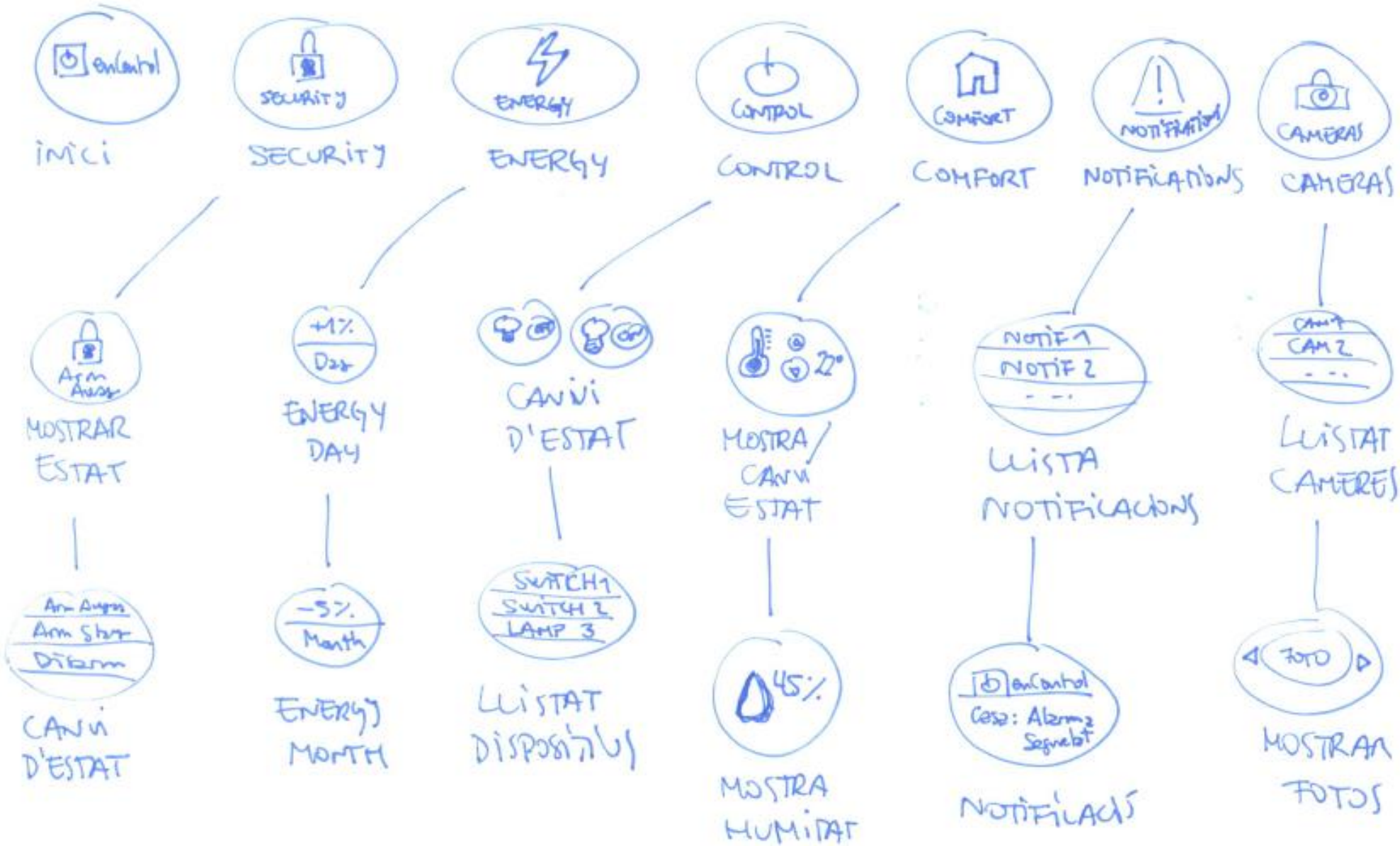
A continuació es poden observar un conjunt de links presents a la documentació oficial, els quals toquen la majoria dels temes referents al disseny gràfic de l'aplicació i a la seva estructura. L'objectiu és seguir els consells al màxim possible durant el disseny d'aquests prototips:

- Design Principles: https://developer.android.com/design/wear/principles.html
- UI Patterns: https://developer.android.com/design/wear/patterns.html
- Style: https://developer.android.com/design/wear/style.html
- Creative Vision: https://developer.android.com/design/wear/creative-vision.html
- App Structure: https://developer.android.com/design/wear/structure.html

Abans d'entrar a valorar els primers dissenys, també cal comentar una particularitat especial d'Android Wear. Des de sempre, estem habituats als rellotges amb l'esfera rodona. Tot i que actualment estan apareixent-ne amb la pantalla rodona, els primers models d'Android Wear la tenien quadrada. En realitat, seguint els consells de Google, no ha de suposar excessiu problema aquest canvi de forma, ja que si la UI està ben dissenyada s'ha d'adaptar a les dues formes. Tot i això, els prototips d'aquesta fase els he realitzat considerant la pantalla rodona per dos motius: el dispositiu per a les proves reals serà un Moto360, i per tant, seran amb un rellotge de pantalla rodona. La segona és una opinió personal, i és que la pantalla rodona em sembla molt més elegant i espectacular.

"Sketches" escanejats

A la següent pàgina es poden observar els primers "sketches" que vaig realitzar a mà alçada per a tenir una idea inicial de com havien de ser les diverses interfícies, i sobretot, per a decidir en primera instància que havia d'incorporar a cadascuna. Seguint els consells de disseny per a Android Wear, he intentat que aquestes interfícies fossin el màxim senzilles dins del possible, i per això, un element a destacar és la seva simplicitat.


enControl
SECURITY
ENERGY
CONTROL
COMFORT
NOTIFICATIONS
CAMERAS
INICI
SECURITY
ENERGY
CONTROL
COMFORT
NOTIFICATIONS
CAMERAS
Arm Away
Arm Stay
Disarm
MOSTRAR ESTAT
+1%
Day
ENERGY DAY
CANVI D'ESTAT
22°
MOSTRA/ CANVI ESTAT
NOTIF 1
NOTIF 2
- - -
LLISTA NOTIFICACIONS
CAM 1
CAM 2
- - -
LLISTAT CAMERES
CANVI D'ESTAT
-5%
Month
ENERGY MONTH
SWITCH 1
SWITCH 2
LAMP 3
LLISTAT DISPOSITIUS
45%
MOSTRA HUMITAT
enControl
Casa: Alarma Segurebat
NOTIFICACIÓ
FOTO
MOSTRAR FOTOS

Prototipus horitzontal d'alta fidelitat

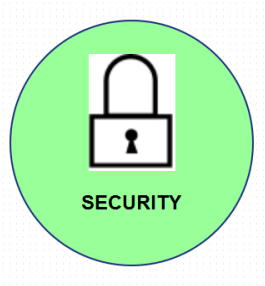
Per a la representació dels prototipus horitzontal d'alta fidelitat, s'ha partit dels sketches mostrats anteriorment, però aquest cop intentant donar una representació el màxim semblant al que acabaran sent les interfícies. Al ser un prototipus horitzontal, s'ha donat prioritat a la representació d'aquestes interfícies enfront de la navegació entre elles (la navegació es pot extreure dels fluxos d'interacció explicats a la fase anterior.

Pantalla 1

La pantalla principal, és la primera que l'usuari veu quan accedeix a l'aplicació. Simplement es tracta del logo de l'aplicació i el nom d'aquesta, per a que l'usuari tingui ben present que ha accedit a l'aplicació.

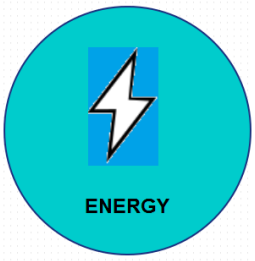
Pantalla 2

Aquesta pantalla correspon a l'àrea de seguretat. S'ha optat per la simplicitat simplement amb la icona del candau i el nom de l'àrea. En aquest cas, (igual que en la resta de pantalles corresponents a les àrees principals), no es mostra l'estat actual, ja que això correspon a quan l'usuari fa clic per accedir a l'àrea.



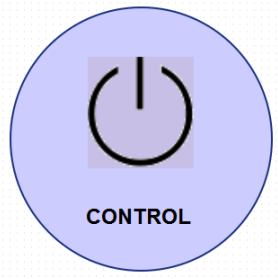
Pantalla 3

Similar a la pantalla 2, aquesta pantalla correspon a l'àrea d'energia. S'ha optat per la simplicitat amb la icona del llamp i el nom de l'àrea.



Pantalla 4

Similar a la pantalla 2, aquesta pantalla correspon a l'àrea de control. S'ha optat per la simplicitat amb la icona típica de "power on/off" i el nom de l'àrea.



Pantalla 5

Similar a la pantalla 2, aquesta pantalla correspon a l'àrea de confort. S'ha optat per la simplicitat amb la icona de la casa i el nom de l'àrea.

Pantalla 6

Similar a la pantalla 2, aquesta pantalla correspon a l'àrea de notificacions. En un primer moment vaig optar per a indicar-la amb una d'exclamació i el nom de l'àrea. Tot i això, i seguint el consell del consultor expert en usabilitat, és cert que el símbol d'exclamació pot donar la sensació a l'usuari que es tracti d'errors o esdeveniments perjudicials, i per tant, vaig decidir canviar-lo per el típic símbol d'informació amb una "i" de color blau.

Disseny inicial (NO implementat):

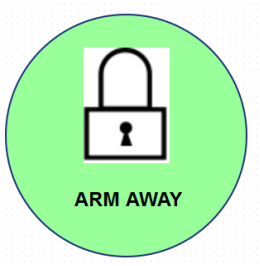
Disseny final:

Pantalla 7

Similar a la pantalla 2, aquesta pantalla correspon a l'àrea de càmeres. S'ha optat per la simplicitat amb la icona de la càmera i el nom de l'àrea.

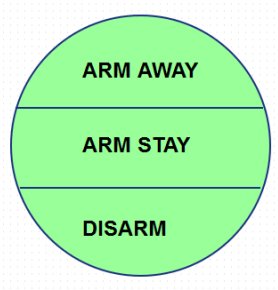
Pantalla 2.1

Aquesta pantalla mostra l'estat actual de la seguretat de la instal·lació. En aquest cas veiem que l'estat és "Arm Away" (alarma connectada) i per això l'he associat amb el candau tancat. En el cas de l'estat "Disarm" (alarma desconnectada) aquest candau apareixerà obert. Crec que aquestes dues imatges poden donar a l'usuari una idea ràpida de l'estat, a part del propi text.



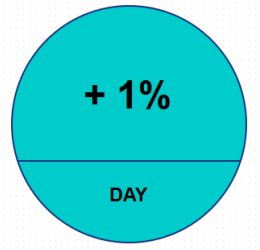
Pantalla 2.2

Aquesta pantalla correspon al canvi d'estats de l'àrea de seguretat. Simplement es tracta d'una llista amb les opcions disponibles. Amb un clic, l'usuari podrà canviar entre ells. Des del meu punt de vista, és la millor representació, ja que l'usuari té a la vista els tres estats possibles (si estiguessin en 3 pantalles diferents accessibles mitjançant swipe, potser l'usuari no veuria que pot canviar entre elles, a més del problema per a tornar a la pantalla anterior 2.1).



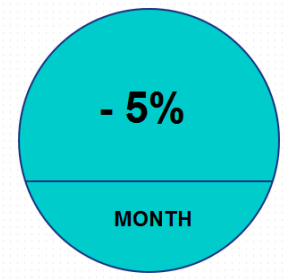
Pantalla 3.1

Aquesta pantalla destaca per la seva simplicitat, però crec que l'usuari simplement necessita veure al wearable el valor total de l'energia consumida durant el dia. Si vol veure el desglossament d'aquesta energia per àrees pot veure-ho al mòbil o a la web, on la pantalla més gran li permetrà observar millor el detall de les dades.



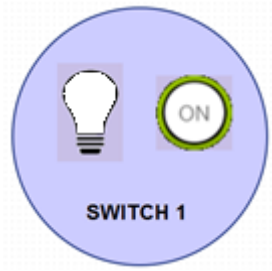
Pantalla 3.2

És exactament el mateix que la pantalla 3.1 però amb el valor i el text corresponent al mes.



Pantalla 4.1

En aquesta pantalla podem veure dos elements: d'una banda una icona d'una bombeta, que quan l'estat és ON apareixerà encesa (com és aquest cas) i quan és OFF apareixerà apagada. Crec que pot ser útil per a que l'usuari tingui la confirmació respecte al botó del costat. Al costat tenim un botó clicable, que ens mostra l'estat actual i al clicar-lo mostrarà el nou estat.

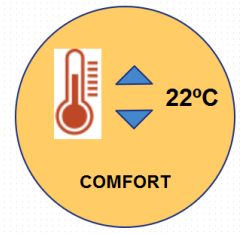


Pantalla 4.2

Igual que a la pantalla 2.2, en aquest cas veurem un llistat amb tots els sensors de control disponibles. Això ens permetrà navegar entre ells per a veure el seu estat o canviar-lo. En aquest cas no adjunto la interfície ja que serà igual que en la 2.2 (ListView) però canviant els noms pels noms dels sensors.

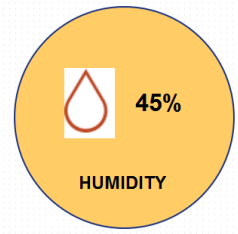
Pantalla 5.1

En aquesta pantalla podem veure que el conjunt del termòmetre, les fletxes i la temperatura formen un grup corresponent al termòstat de la instal·lació. Les fletxes són un bon element per a indicar a l'usuari que pitjant-les pot canviar la temperatura objectiu del termòstat.



Pantalla 5.2

Aquesta pantalla simplement mostra el nivell d'humitat present a la casa. No té cap element d'interacció

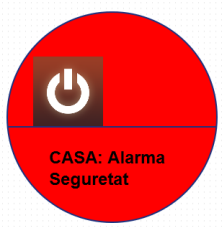


Pantalla 6.1

Igual que a la pantalla 2.2, en aquest cas veurem un llistat amb les últimes notificacions. Això ens permetrà navegar entre elles per a veure-les amb detall. En aquest cas no adjunto la interfície ja que serà igual que en la 2.2 (ListView) però canviant els noms pel títol de les notificacions.

Pantalla 6.2

Aquesta pantalla correspon a una notificació. Les notificacions són un punt molt important segons les guies de disseny de Google, motiu pel qual m'he adaptat totalment al disseny que proposen. És per això que surt el logo de l'aplicació (per a que en cas de no estar dins l'aplicació l'usuari la pugui relacionar), i just a sota el missatge corresponent.

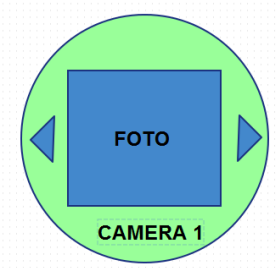


Pantalla 7.1

Igual que a la pantalla 2.2, en aquest cas veurem un llistat amb totes les càmeres disponibles. Això ens permetrà navegar entre elles per a veure les seves fotografies. En aquest cas no adjunto la interfície ja que serà igual que en la 2.2 (ListView) però canviant els noms pels noms de les càmeres.

Pantalla 7.2

Finalment, aquesta pantalla mostrarà les fotografies corresponents a la càmera seleccionada per l'usuari. Les fletxes al costat de la foto indiquen que és possible navegar entre les fotos disponibles. En aquest cas he dubtat força si incloure-les o no (ja que en teoria això ja es fa amb el moviment swipe), però crec que en aquest cas concret cal remarcar-li a l'usuari que té disponibles més fotografies i no únicament una.



Avaluació

Per a l'avaluació del prototip dissenyat a la fase anterior, s'ha optat per a fer un test d'usuari. La idea és intentar passar aquest test a usuaris que compleixin amb els perfils detectats a la fase 1, ja que d'aquesta manera podem veure si els requeriments que hem trobat són correctes (en el cas de que no ho siguin, podríem canviar-los a la propera iteració).

Abans de respondre el test, es demanarà als usuaris que facin una sèrie de tasques de dificultat creixent. S'espera que depenent del perfil pugin completar-les totes o només una part (evidentment l'ideal seria que tots poguessin realitzar la totalitat, però ens conformarem col·locant un nombre de tasques mínimes per al perfil d'adult reticent a canvis). Les tasques demanades seran:

Grup 1: Tots els usuaris han de completar aquestes tasques:
- Veure l'estat de seguretat de la casa i activar l'alarma.
- Canviar l'estat d'un interruptor.
- Observar el consum diari d'energia.
- Observar el nivell d'humitat de la casa.
- Rebre i descartar una notificació.
- Descartar notificació mitjançant veu.

Grup 2: Tasques de nivell mitjà. Com a mínim els usuaris joves i adults no reticents a canvis han de ser capaços de portar-les a terme.
- Rebre notificació de canvi de temperatura i realitzar l'acció immediata per a tornar-la a canviar.
- Rebre notificació de seguretat i observar la fotografia associada.
- Veure una notificació del llistat
- Veure l'última fotografia del llistat de càmeres.

Grup 3: Tasques de nivell alt. Com a mínim els usuaris joves han de ser capaços de portar-les a terme.
- Rebre notificació d'alarma, observar la fotografia i canviar l'estat de seguretat.
- Veure temperatura de la casa mitjançant comanda de veu.
- Canviar l'estat de l'alarma mitjançant comanda de veu.

En quant a les preguntes del test, s'ha intentat principalment centrar l'estudi en els requeriments que els usuaris han demanat durant l'enquesta d'indagació, però també qüestions bàsiques d'usabilitat, les quals és necessari no oblidar. Per tant, l'objectiu ha estat principalment obtenir l'opinió dels usuaris en tres aspectes:

1) Facilitat d'ús i ajuda de la intuïció per a resoldre les tasques habituals.

2) Compliment dels requeriments (funcionalitats) demanats pels usuaris

3) Usabilitat general del disseny, sobretot centrant-nos en si els protocols que hem seguit per part de Google s'adapten al que volen els usuaris (per exemple, navegació habitual mitjançant swipe).

A continuació es mostra el test que els usuaris hauran de respondre:

1) Presentació (es demana una presentació al usuari per a ubicar-lo en el perfil adequat)

2) Navegació aplicació (Pantalla inicial i pantalles amb les icones d'àrea):
    a. És possible identificar la companyia?
    b. Consideres que el disseny de les icones és bonic i amigable? Creus que les icones són prou entenedores de l'àrea que representen?
    c. Creus que el "swipe" és un bon mètode de navegació? O preferiries veure les àrees en format llista?
    d. Trobes a faltar algun element entre aquestes icones principals?

3) Navegació aplicació (Accions concretes dins de cada àrea):
    a. Has trobat a faltar alguna acció o visualització que estigui present a la web o a l'aplicació mòbil? Quina?
    b. Creus que sobra alguna de les accions disponibles? Perquè?
    c. Són les icones suficientment indicatives del que representen?
    d. Consideres que el disseny és bonic i amigable? Creus que les icones són prou entenedores del que representen?
    e. En aquests apartats, en molts casos es presenten els elements en format llista. Et sembla correcte? O preferiries navegar entre ells mitjançant "swipe"?

4) Notificacions:
    a. Et sembla útil rebre les notificacions al rellotge?
    b. Que n'opines del nombre de notificacions rebudes? Escàs o excessiu?
    c. Creus que són útils les accions immediates que es poden realitzar després de rebre la notificació? N'has trobat a faltar alguna?
    d. Que et sembla el disseny de les notificacions? Creus que està bé seguir amb l'estàndard de Google (mateix format que les notificacions d'altres aplicacions) o preferiries que aquestes estiguessin clarament diferenciades?

5) Reconeixement de veu:
    a. Quina és la teva valoració sobre el sistema de reconeixement de veu de l'aplicació?
    b. Has pogut realitzar les accions desitjades mitjançant comandes de veu?
    c. En el futur, utilitzaries aquest sistema per a canviar l'estat dels elements d'enControl?

6) Avaluació final (Quantitativa)
    a. De l'1 al 10 indica el teu nivell de satisfacció amb l'aplicació.
    b. De l'1 al 10 indica el nivell de dificultat trobat per a realitzar les accions.
    c. De l'1 al 10 indica el nivell del disseny gràfic de l'aplicació (si és bonic/amigable, si t'hi has sentit còmode).
    d. De l'1 al 10 indica la probabilitat de que en cas de voler realitzar algun canvi, sigui mitjançant el rellotge enfront del mòbil.

Resultats Avaluació

En quant als resultats de l'avaluació, el test està englobat dins del procés iteratiu de desenvolupament del projecte, i per tant, seria necessari realitzar-lo després de cada sprint o conjunt de sprints. Tot i això, per falta de temps només l'he pogut realitzar al final del projecte i amb només 2 usuaris, però els resultats obtinguts poden ser útils de cara a propers desenvolupaments de l'aplicació.

A continuació es mostra un resum de les respostes obtingudes en les diverses preguntes:

1) Presentació dels usuaris. Els dos usuaris compleixen els perfils desitjats (joves de 25 i 29 anys).

2) Consideren que la navegació de la pantalla i el menú principal de l'aplicació és bona, especialment el fet que amb el swipe a l'esquerra es torni sempre enrere.

3) També consideren que la navegació a cadascuna de les àrees és bona. El disseny és amigable i n'estan molt satisfets. Prefereixen el format llista per a escollir àrees i opcions. No troben a faltar cap element respecte a la versió mòbil. Només un dels usuaris ha considerat que es trenca una mica el flow quan s'accedeix al llistat de les notificacions (perquè surt de l'aplicació i quan es torna a entrar és a la pantalla inicial).

4) Pel que fa a les notificacions, les han trobat correctes, tot i que ambdós usuaris han demanat ampliar les accions disponibles per a fer a partir d'elles. En quant al disseny consideren lògic seguir els estàndards de Google.

5) La valoració respecte al reconeixement de veu no ha estat gaire bona. Un dels usuaris no ha estat capaç de que el rellotge l'entengués (l'accent anglès no era del tot correcte), i l'altre usuari ha considerat que la utilitat d'aquesta funcionalitat no aporta valor afegit.

6) Avaluació final (mitja de la puntuació donada pels dos usuaris):

    - Nivell de satisfacció amb l'aplicació: **8,5**
    - Nivell de dificultat per realitzar les accions: **5**
    - Nivell de disseny gràfic: **9**
    - Probabilitat d'usar el rellotge enfront del mòbil: **6,5**

3) DISSENY TÈCNIC

Abans de començar a dissenyar tècnicament la futura implementació del projecte, cal remarcar de nou que la realització de l'aplicació està molt pautada per les guies de Google, i això fa que com veurem a continuació, no s'apliquin alguns dels paradigmes habituals en el món de la programació. A més, al tractar-se d'una extensió d'un projecte ja existent (enControl), hi haurà elements que ja venen donats, i sobre els quals s'haurà d'adaptar el disseny més que "crear-lo".

Per tant, en aquest apartat s'exposa els tipus de dades que s'emmagatzemaran i com ho faran, el disseny del software a nivell general (diagrama de classes UML) i el disseny de l'arquitectura del sistema, posant especial èmfasi en les dues transferències de dades possibles: del smartphone al smartwatch i viceversa.

Base de dades

El primer exemple que trobem respecte a aquests dos factors comentats és el fet que l'aplicació no compta amb base de dades pròpia. Més concretament, la base de dades sí que existeix, però es troba a la plataforma Cloud on està ubicat enControl. A aquesta base de dades no es pot accedir directament, sinó que s'accedeix a través d'una API pròpia.

En el cas de l'aplicació per al wearable, ni tan sols s'accedeix a aquesta API directament, ja que com s'explicarà més detalladament al tercer apartat (Arquitectura del Sistema), és l'smartphone el que es comunica amb la API, i el wearable només intercanvia missatges (en les dues direccions) amb el mòbil.

Per tant, tot i que no existeix una base de dades pròpia de l'aplicació, sí que és cert que cal consultar, guardar i intercanviar informació entre els dos elements (rellotge i mòbil). Com podem veure a la documentació oficial, hi ha diverses maneres de realitzar aquestes operacions (https://developer.android.com/training/wearables/data-layer/index.html), i en aquest cas, s'haurà de fer ús de gairebé totes.

En primer lloc tenim l'element bàsic, els "DataItems", que permeten guardar dades mantenint la sincronia entre l'smartphone i el wearable. Per tant, els DataItems seran emprats en els casos en que mostrem informació per pantalla i aquesta pugui ser sotmesa a canvis externs. Dintre d'aquesta categoria trobem els diversos estats i valors que es mostren per pantalla (per simplicitat considero que les llistes, de càmeres o sensors per exemple, no seran modificades durant la seva visualització).

En segon lloc trobem els "Messages", que tal i com el seu nom indica són missatges que s'envien entre el mòbil i el rellotge. En aquest cas, s'empraran per als casos on vulguem enviar una ordre des del wearable al smartphone. Per exemple, si l'usuari decideix canviar l'estat de seguretat, es crearà un missatge que anirà al mòbil, i el mòbil decidirà que fa amb el missatge (haurà de realitzar l'acció que correspongui per a comunicar-se amb la API).

Finalment, els "Assets" serveixen per a adjuntar "blobs" de dades als DataItems. Són molt útils per a l'enviament de fotografies al rellotge.

Disseny software aplicació

En el cas del disseny del software de l'aplicació, també trobem força diferències amb els casos habituals. La primera diferència i que també és comuna amb Android, són les classes anomenades "Activitiy". Aquestes classes són les principals encarregades de mostrar i operar amb la UI. Per tant, serà necessària una Activity per a cadascuna de les pantalles que volem mostrar.

Dins de les Activities, hi ha algunes pantalles que contenen llistes. Aquestes llistes són creades mitjançant "ListViews", les quals per a tenir un correcte funcionament necessiten una classe "Adapter" (per a crear-les correctament i omplir-les de contingut), i una classe "WearableListItemLayout" (per a crear la forma de l'element de la llista, i en cas necessari una animació).

Addicionalment, també trobem dues classes més, encarregades de les notificacions i de l'intercanvi d'informació entre mòbil i rellotge:

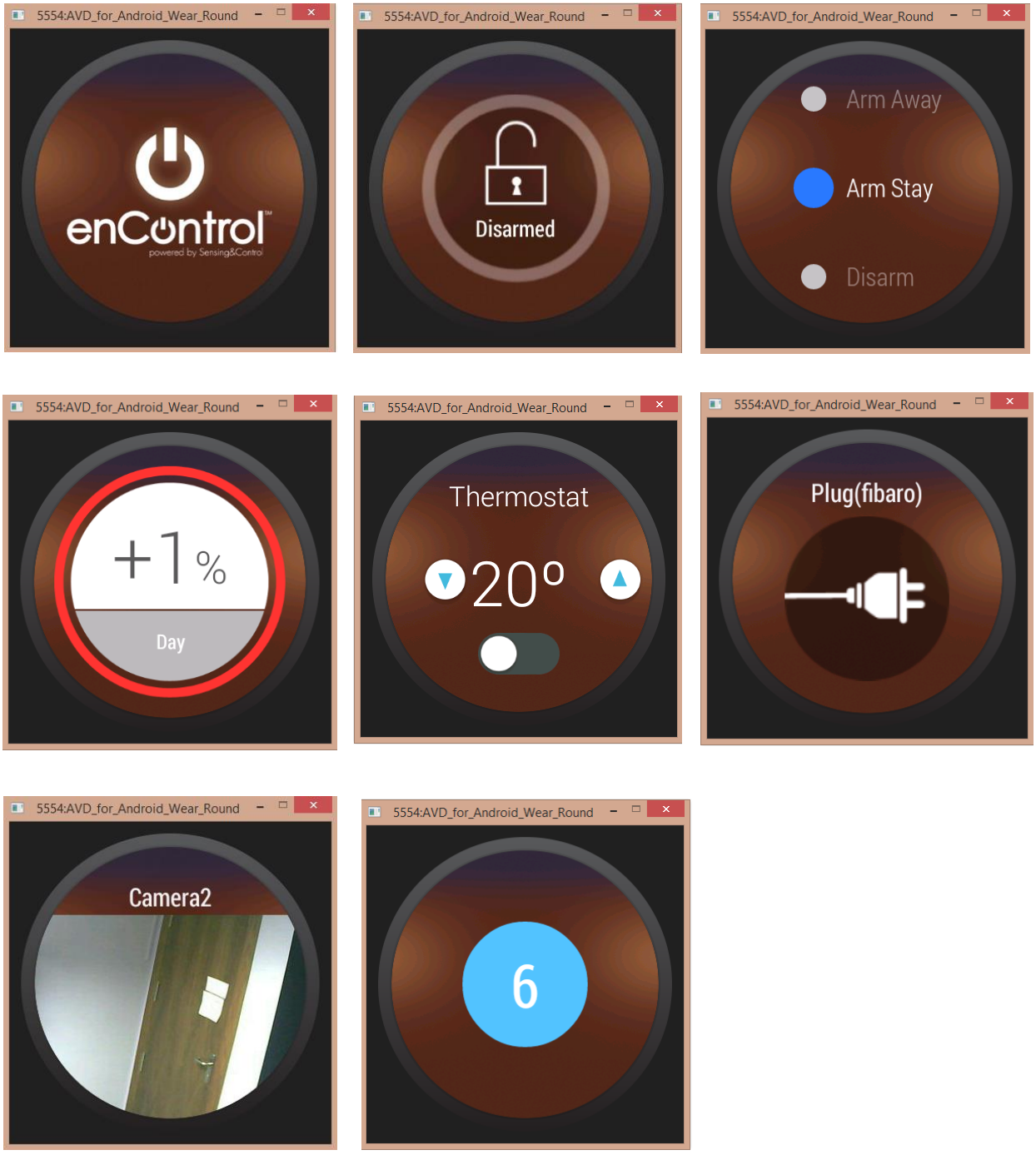
- NotificationsHub: Encarregat de rebre les notificacions del mòbil i donar la funcionalitat que els hi correspongui en cada cas.

- MessageSender: Aquesta classe servirà per a poder enviar missatges des del wearable al smartphone, i també per a rebre la resposta corresponent.

Em sembla important comentar que en el disseny tècnic inicial, també havia afegit una classe anomenada ***WearableListener***, encarregada de centralitzar la comunicació entre Smartphone i Wearable (en aquesta direcció). Després de començar la implementació, és a dir, posteriorment a la realització d'aquesta fase, vaig veure que em resultava més convenient utilitzar el mètode onDataChange() de DataApi.DataListener, ja que d'aquesta manera a cada Activity activa es pot comprovar si els canvis corresponen a la seva interfície i pot aplicar-los directament. Per tant, ha estat necessària una petita actualització del disseny tècnic a posteriori (els canvis també han estat reflexats als diversos elements gràfics).

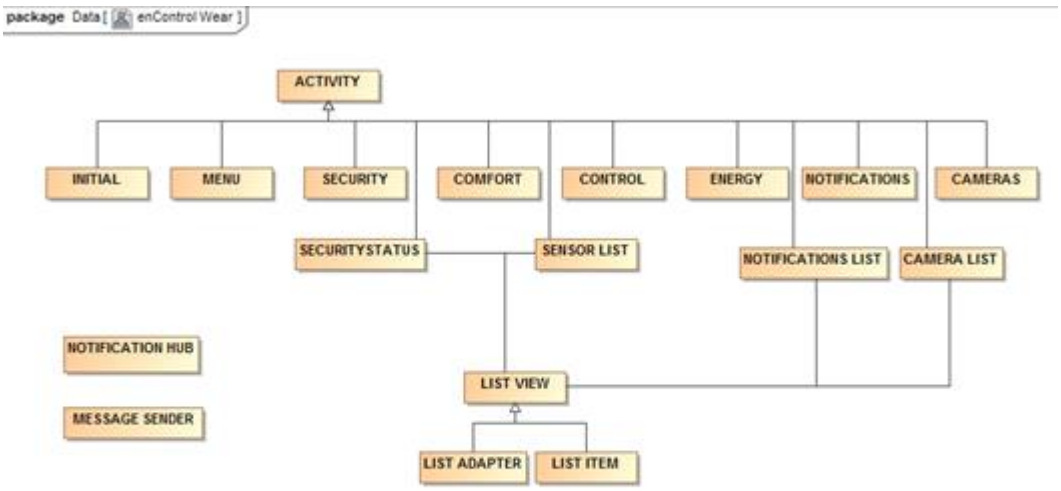
Cal dir que arribats a aquest punt i després d'estudiar més a fons la documentació de Google i començar la implementació, va ser necessari realitzar un petit canvi respecte al disseny inicial: en comptes de fer una pantalla per a cadascuna de les àrees principals, finalment he decidit realitzar un menú principal, a partir del qual l'usuari pot escollir l'àrea que desitgi (es correspon a la classe Menú del diagrama). A continuació es pot observar la captura de pantalla del simulador corresponent al nou disseny del menú, així com altres captures d'algunes de les altres pantalles definitives:

Menu Activity



Respectivament es tracta de les pantalles (Activity) corresponents a:
Initial, Security, SecurityStatus, Energy, Comfort, Control, Cameras, Notifications.

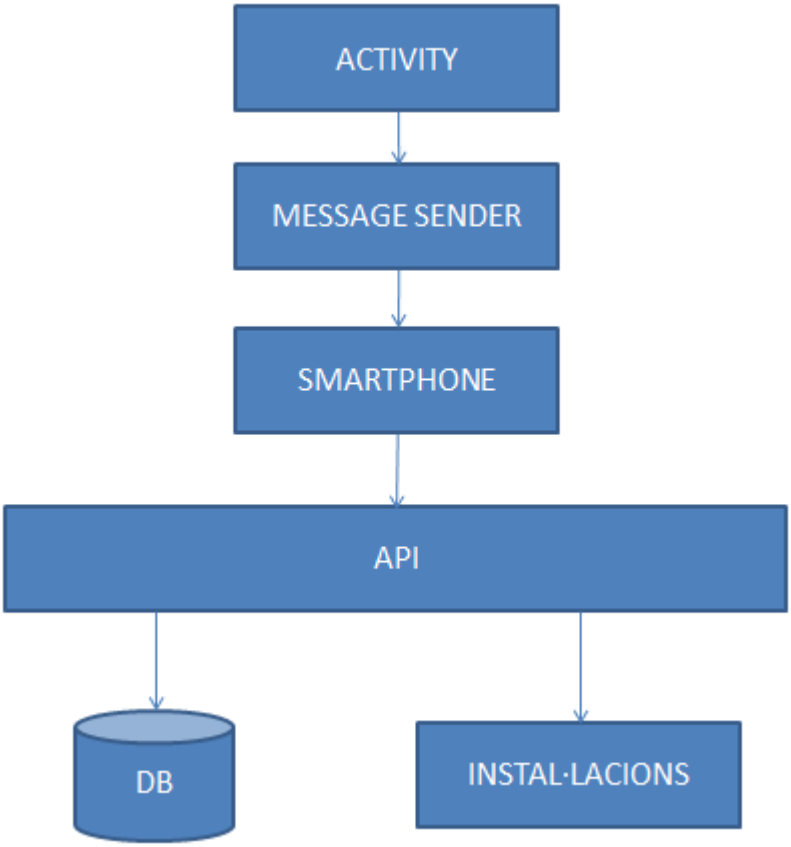
Per tancar aquest apartat, a continuació es mostra un disseny UML que intenta reproduir al màxim l'estructura que tindrà el software.



Arquitectura del Sistema

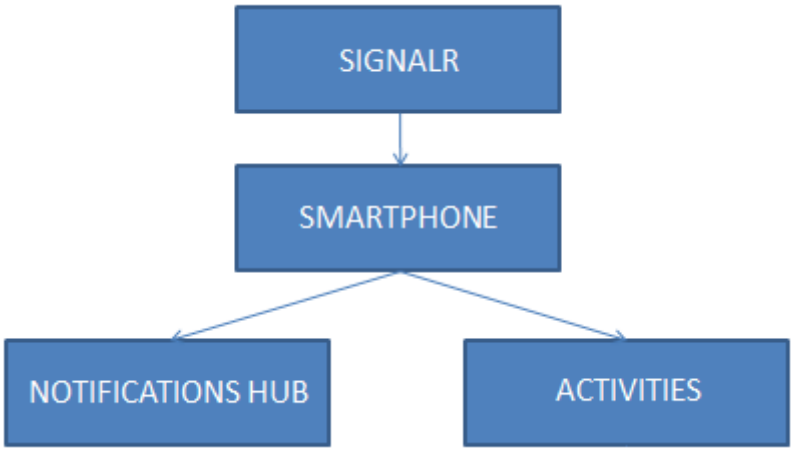
Pel que fa a l'arquitectura del sistema, novament vull remarcar que també es tracta d'un cas molt específic i especial. El primer que hem de tenir en compte és que el rellotge únicament es comunica amb el mòbil. És per aquest motiu que les dues classes per a enviar/rebre missatges són claus, ja que són les que comuniquen el rellotge amb tota la informació. Un cop la informació arriba a l'smartphone, el que passa ja no entra dins l'abast d'aquest projecte, tot i que en aquest apartat està inclosa un petit esquema per a poder tenir una visió global.

Podem diferenciar entre dos branques de l'arquitectura del sistema: la part encarregada d'enviar missatges al smartphone i la part encarregada de rebre dades. En la primera, l'usuari estarà observant una UI mitjançant una Activity, i al prémer un botó per exemple, això provocarà que la classe MessageSender enviï un missatge al smartphone. L'smartphone enviarà la comanda corresponent a la API d'enControl. A partir d'aquí la API realitzarà la operació que correspongui, que pot significar algun canvi en les instal·lacions i/o a la Base de Dades d'enControl.



Pel que fa a la segona part, encarregada de rebre les dades provinents de l'smartphone, consisteix en el sistema "SignalA", llibreria per a Android de SignalR (https://github.com/erizet/SignalA), que proporciona esdeveniments de canvis en temps real per a poder modificar dades sense tenir que cridar a la API. Aquest sistema ja està implementat a l'aplicació Android, i per tant, s'ha implementat que cada arribada de noves dades modifiqui el corresponent DataItem, i mitjançant el mètode onDataChange() cada Activity activa modifiqui l'element de la UI si li correspon.

Un altre cas que hem de contemplar en quant a arribada de dades són les notificacions. En aquest cas, el NotificationHub s'encarregarà de rebre-les i tractar-les com correspongui.


SIGNALR
SMARTPHONE
NOTIFICATIONS HUB
ACTIVITIES

4) DESENVOLUPAMENT

En aquest apartat s'explicarà amb més profunditat els aspectes relacionats amb el desenvolupament de l'aplicació. D'una banda tenim les diverses eines, editors i APIs emprades. També es comentarà l'estructura del codi, ja que al tractar-se d'una plataforma prou innovadora, crec que són necessàries algunes clarificacions abans de començar a endinsar-se en el codi, ja que tal i com em va passar a mi en el seu moment, a vegades costa adaptar-se a aquesta nova tecnologia i la seva arquitectura (tot i ser molt semblant a Android).

Un altre aspecte important és que el conjunt de l'aplicació està dividit en dues, una per a cadascun dels dispositius smartphone i smartwatch (per aquest motiu també hi haurà dos .apk, un pel mòbil i un pel rellotge). L'aplicació mòbil hauria de ser la ja existent d'enControl Android, afegint algunes classes addicionals, però des de l'empresa m'han recomanat que no envies tot el codi sencer, i que només posés les classes i elements totalment imprescindibles. És per això que la app al mòbil funciona correctament pel que fa a la interacció amb el rellotge, però he fixat que a la pantalla simplement surti el logo de l'aplicació. Tot i això, per al vídeo de presentació final del projecte sí que podré fer servir l'aplicació real.

Eines utilitzades

Pel que fa a les eines utilitzades, hi ha una que destaca per sobre de les altres: Android Studio. Ha estat la meva primera experiència amb aquest editor (anteriorment utilitzava Eclipse amb el plugin de desenvolupament d'Android), i la valoració no pot haver estat més positiva. Tot i l'habitual resistència que tenim a fer canvis en el nostre mode de treball, alguns articles d'experts ja fa temps que recomanaven aquest canvi (https://www.infinum.co/the-capsized-eight/articles/eclipse-is-dead-for-android-development-and-i-helped-kill-it), i la veritat és que es noten moltes millores respecte a Eclipse. A més, durant els primers dies de desembre ha sortit la primera release estable, Android Studio 1.0.

Les millores són moltes, i Internet està replet d'articles explicant-les. Des de la meva experiència, he trobat dos aspectes clau:

- ***Representació immediata i fidedigne de les UI***. Aquesta característica també estava present a Eclipse, però la veritat és que la millora és brutal. Tots els canvis realitzats als layouts xml els pots veure instantàniament a una finestreta al costat, fet que millora molt el disseny de les interfícies.

- ***Ús per defecte de Gradle i Maven***. En aquest cas, es tracta de dos coses que també es podien afegir a Eclipse, però en aquest cas ja venen per defecte al IDE, i per això vaig decidir donar una oportunitat per a provar-les. La compilació i afegir llibreries és molt més fàcil amb aquests sistemes.

Una altra eina emprada molt important ha estat l'Emulador d'Android Wear. En aquest cas és important remarcar que no és possible (o no ho va ser per a mi) realitzar les proves amb dos emuladors, un pel mòbil i un pel rellotge. Per això vaig fer les proves connectant un Nexus 4 amb l'emulador Android per a una interfície rodona (no especifica el model, tot i que evidentment està inspirat en el Moto 360). L'emulador m'ha servit per a poder provar tots els desenvolupaments i veure que realment funcionaven.

En quant al propi codi, s'han utilitzat algunes llibreries externes. El seu ús és herència directa de les llibreries emprades en l'aplicació Android, així que aquest ha estat el motiu de la seva elecció. Aquestes llibreries (es poden trobar els arxius .jar a la carpeta ***libs***) són:

- **HttpClient Async**. Llibreria del client asíncron utilitzat per a les comunicacions de l'smartphone amb l'API d'enControl.

- **Gson**. Aquesta llibreria és emprada per a les transformacions d'estructures de dades a JSON i viceversa.

- **SignalA (llibreria Android de SignalR)**. S'utilitza per a rebre en temps real les modificacions d'estat dels diversos elements. Bàsicament consisteix en uns hubs als quals el mòbil es subscriu, i que informen de qualsevol canvi que es produeixi. En aquest cas utilitzo una estructura existent al back-end d'enControl. Més informació de la llibreria es pot trobar a: https://github.com/erizet/SignalA.

- **Notification-Hubs**. S'utilitza per a poder rebre notificacions push des de la plataforma Cloud Azure. Bàsicament, al codi s'indica una clau única per a identificar que el programa ha de rebre un identificador de notificacions push determinades (Id d'usuari), i Azure s'encarrega d'enviar-les quan n'hi hagi.

Addicionalment, i tal com s'ha explicat en apartats anteriors, s'utilitzen estructures existents de la plataforma enControl, com ara:

- **API enControl**. Aquesta API és utilitzada per a comunicar-se amb el back-end d'enControl.

- **Estructura SignalR**. Tal i com comentava anteriorment, existeix tota una estructura corresponent a SignalR al back-end d'enControl que ha estat utilitzada per a l'aplicació.

- **Estructura Notificacions Push**. Igual que en el cas de SignalR, també hi ha una estructura potent al back-end d'enControl que es comunica amb el servidor Azure per a la gestió de les notificacions push. Aquest servidor es troba a la plataforma cloud Azure (de l'empresa Microsoft), i bàsicament s'encarrega de rebre des del servidor de S&C les diverses notificacions que s'han de traslladar als dispositius mòbils dels usuaris. Per a saber on les ha d'enviar, prèviament els dispositius s'han hagut de registrar al hub amb el "userId", per tal que el servidor pugui filtrar quines notificacions li ha d'enviar.

Aplicació mobile

A continuació es presenta una breu descripció dels elements més importants del codi corresponent a l'aplicació mòbil:

MyActivity.java:

Aquesta és l'Activity de l'aplicació mòbil més important, ja que és la que realitza la connexió entre la API enControl i l'smartwatch. A continuació comentaré el seu flux de funcionament:

- Abans de començar, voldria remarcar que els Id d'usuari i instal·lació apareixen fixats per a facilitar les proves, però la idea és que siguin dinàmics depenent de l'usuari que hagi accedit al mòbil.

- En primer lloc, l'Activity connecta amb el GoogleApiClient, que serà el client per a comunicar-se amb el rellotge. Un cop s'ha connectat, s'afegeixen dos listeners (DataApi i Messages), i es fa una crida al mètode "getInitialData()", per a obtenir el bloc de dades inicials.

- Aquest mètode conté diversos mètodes, cadascun dels quals crida a la corresponent funció del WebApiConsumer. També inicialitza el SignalR, el GCM Messaging i el BroadcastReceiver, utilitzat per a enviar la foto un cop descarregada o recuperada de la memòria del telèfon. Finalment, prepara la seva re execució per al cap de 30 min, per a actualitzar els elements que no ho fan per SignalR.

- Una vegada l'API retorna resposta, dins el callback del mètode que l'ha cridat (onSuccess) es crida als mètodes d'update, que són els encarregats d'actualitzar els DataItems.

- L'estructura d'aquests mètodes és molt semblant en tots els casos: s'assigna a diverses claus (aquestes claus es poden trobar a la classe Constants) el valor que posteriorment serà recuperat pel wear (en el cas de les estructures de dades, aquestes són prèviament serialitzades). Un cop s'ha creat el DataMapRequest correctament, aquest es comparteix a la DataApi.

- Addicionalment també podem trobar en aquesta classe el MessageListener (onMessageReceived), que és l'encarregat d'escoltar les accions que arriben des del wear. En tots els casos recupera les dades associades al missatge per a saber l'acció exacta que ha de realitzar. Finalment, realitza la crida al mètode que correspongui (postControlStatus, postThermostatStatus, etc), que serà l'encarregat de comunicar-ho a la API via WebApiConsumer.

- També podem veure que hi ha certs elements específics per a les càmeres. Això és degut a que l'esquema d'acció en aquest cas és:
    - Recuperar via API la URL d'Azure on està guardat el blob amb la foto.
    - Cridar la classe DownloadImage, encarregada de descarregar i guardar en memòria aquesta foto. Es tracta d'una classe a part per a poder realitzar-la en segon pla i que no bloquegi la resta de les accions. Les fotos descarregades, es guarden a la memòria externa del telèfon, i per tant, la propera vegada que es torni a demanar, ja no serà necessari descarregar-la.
    - Un cop s'ha obtingut la foto (memòria o descàrrega), es retorna a l'activitat mitjançant un BroadcastReceiver.
    - Finalment, dins el BroadcastReceiver, es comparteix mitjançant la Google API un DataItem amb la fotografia com a asset associat.

- Finalment, tenim la secció corresponent a SignalR, la qual té el següent funcionament:
    - En primer lloc, és necessari connectar-se al Hub.
    - Un cop connectat, és necessari unir-se als grups dels que volem rebre la informació.

- A partir d'aquí, és necessari unir-se als mètodes de cada Hub que desitgem, i implementar les corresponents accions a realitzar quan arribin noves dades. En aquest cas, la norma habitual en tots els casos és actualitzar les variables globals de l'aplicació de l'smartphone per a posteriorment poder reutilitzar els mètodes "Update" anteriorment comentats, els quals actualitzen el DataItem. Com que el wear estarà escoltant el DataItem contínuament, això fa que finalment s'aconsegueixi traslladar el canvi des del servidor fins al rellotge gairebé immediatament.

- Finalment, tenim la subscripció a Google Cloud Messaging (GCM) Notifications, que es connecta al hub de notificacions Push d'enControl a Azure, per a rebre les notificacions push corresponents al Id d'usuari.

WebApiconsumer.java:

El WebApiConsumer es tracta d'una classe Singleton, la qual utilitza aquesta única instància per a comunicar-se amb la API d'enControl. Corresponen en la seva majoria a les mateixes crides de l'aplicació original Android.

Més concretament, observant les crides es pot observar que sempre segueixen el mateix patró:

- Mitjançant el client es fa una crida amb mètode (GET-POST-PUT) i a la URL que correspongui.

- Si s'obté una resposta exitosa, es deserialitza l'string de resposta a l'estructura de dades corresponent. Es pot observar que degut a problemes que encara no entenem de la API, a vegades les respostes de l'onSuccess no poden ser correctament deserialitzades per Gson, i és per això que també s'ha incorporat en aquesta part el sistema de retries.

- Si la resposta és errònia, normalment el motiu és un error d'autenticació. És per això que s'autentica el client de nou i es torna a fer retry de la crida (si encara no s'ha arribat al nombre màxim de retries).

NotificationsHub.java:

Aquesta classe és una extensió del BroadcastReceiver, i és l'encarregada de rebre les notificacions Push. Un cop les rep, crea una notificació Android extensible a Android Wear, i per tant, la mateixa notificació es rep al mòbil i a l'smartphone, tot i que en formats diferents (fins i tot es podrien fer accions diferents per a cadascun d'ells).

DownloadImage.java:

Aquesta classe comprova en primer lloc si la imatge que volem descarregar ja està present a la memòria del telèfon. Si és així, la retorna a l'activitat principal. En cas de que sigui necessari descarregar-la, inicia una task en segon pla que ho farà. Una vegada s'ha descarregat la nova imatge correctament, aquesta és guardada en memòria, i enviada a l'activitat principal mitjançant el BroadcastReceiver.

Constants.java:

Aquesta classe conté les claus per a insertar/recuperar els diversos DataItems. També apareix a l'aplicació wear.

La resta de classes:

Totes les altres classes presents en aquest paquet són classes de dades, necessàries per a poder transformar les estructures JSON que arriben o que volem enviar des de la API.

- ActuableSensorsResponse

- Camera
- CameraLastPhotoResponse
- ComfortValues
- ComfortValuesResponse
- Event
- EventBatchResponse
- EventsCountResponse
- Installation
- InstallationDetailsResponse
- ResponseError
- ResponseStatus
- SecurityStatusResponse
- Sensor

Aplicació wear

A continuació es presenta una breu descripció dels elements més importants del codi corresponent a l'aplicació del rellotge:

MyActivity.java:

És la primera Activity que l'usuari veu quan entra a l'aplicació. Bàsicament controla que el rellotge i el mòbil estiguin connectats (en cas que no ho estiguin mostra un missatge per pantalla, ja que l'aplicació no pot continuar). Si ho estan, mitjançant un click a la pantalla comença l'Activity següent, el menú per a escollir l'àrea a tractar.

MenuActivity.java:

Aquesta activity presenta el menú per a escollir l'àrea sobre la qual l'usuari vol accedir. Aquest menú s'ha extret dels samples que proporciona Google, i utilitza també la classe "WearableListItemLayout". Un cop es clica en un dels elements de la llista, es redirigeix a l'usuari a l'activity que pertoqui.

Activities que representen les diverses pantalles de la UI:

Aquestes Activities són aquelles que corresponen a les pantalles sobre les que l'usuari podrà navegar, veure els valors actuals i realitzar les accions. Més concretament són:

- ***SecurityActivity***: Mostra l'estat de l'alarma de seguretat, i amb un clic sobre ella s'accedeix a SecurityStatesListActivity, on es pot canviar l'estat.

- ***SecurityStatesListActivity***: En aquesta Activity es pot canviar l'estat de l'alarma. Les diverses opcions apareixen en format llista.

- ***EnergyActivity***: Presenta els diversos valors d'energia per a dia, setmana i mes. Cadascun d'ells es mostra sol per pantalla, i es pot navegar entre ells mitjançant un clic.

- ***ControlActivity***: Mostra el llistat de sensors disponibles sobre els que es pot actuar.

- ***ControlDeviceActivity***: Mostra l'estat d'un sensor en particular, i permet modificar el seu estat entre ON/OFF alternativament, mitjançant un clic a la pantalla.

- ***ComfortActivity***: Mostra un llistat amb tres opcions de confort: Inside, Outside i Thermostat.

- ***ComfortInsideActivity***: Permet visualitzar la temperatura i humitat interior de la casa.

- ***ComfortIOutsideActivity***: Permet visualitzar la temperatura i la previsió de l'exterior de la casa.

- ***ComfortThermostatActivity***: Permet visualitzar, i controlar els valors i l'estat del termòstat.

- ***NotificationsActivity***: Aquesta pantalla mostra el nombre de notificacions pendents de l'usuari (pantalla modificada respecte al disseny previ). Al clicar el número s'accedeix a la següent activity on es pot escollir el tipus de notificacions a visualitzar.

- ***NotificationsByTypeActivity***: Mostra un llistat amb els quatre tipus de notificacions. Un cop es clica sobre un d'ells, s'obren les 10 últimes notificacions d'aquest tipus. Si es clica en una de les notificacions i es fa swipe a l'esquerra, podrem navegar entre les diverses opcions que es poden realitzar a partir de la notificació. Aquestes opcions són:
    - Action: Al clicar sobre aquesta opció, s'obrirà una UI especial per al reconeixement de veu, que estarà pendent de rebre dues paraules clau,

"Read" i "Delete". Si el que diu l'usuari coincideix amb alguna d'aquestes accions, es procedirà a realitzar-la.
  - Open: Obre l'àrea corresponent a la notificació, per si l'usuari vol realitzar alguna acció més específica.

- ***CamerasActivity***: Mostra el llistat de càmeres disponibles.

- ***CamerasPhotosActivity***: Per a la càmera seleccionada, mostra la última fotografia realitzada. Tal i com s'ha comentat anteriorment, he decidit no implementar el llistat de fotografies.

Després d'aquest anàlisi a grans trets de cadascuna de les pantalles, és necessari comentar més concretament alguns elements que apareixen en elles:

- WatchViewStub: Aquest element s'utilitza per a poder programar per als dos tipus de pantalles que hi ha a Android Wear, round i square. S'utilitza de manera que a l'Activity es referencia el WatchViewStub, i dins d'aquest s'indica quin layout s'utilitzarà en cas de dispositiu quadrat i quin en cas de dispositiu rodó.

- GoogleApiClient. En aquestes Activity és necessari recuperar DataItems, i per tant, és necessari connectar amb la GoogleApiClient. Un cop connectat, es procedeix a cridar el mètode getInitialValues.

- getInitialValues(). Aquest mètode recupera els valors inicials per a mostrar-los per pantalla. En el cas de les notificacions i les càmeres és una mica diferent, ja que en aquest mètode s'envia un missatge al mòbil per a indicar-li que comenci la descàrrega de la foto o les notificacions.

- onDataChanged(). Aquest mètode correspon a DataApi.DataListener i és de vital importància. Mentre l'Activity està activa, aquest mètode s'activarà cada cop que un DataItem canviï. Per tant, per a cada activity es comprova si el canvi l'afecta, i en cas de que així sigui es procedeix a actualitzar la UI.

- Finalment, també hi ha algunes activities amb format de llista. El funcionament és exactament el mateix que per a MenuActivity, tot i que en aquests casos els items de la llista no són fixats, i s'ha de cridar primer a getInitialValues per a obtenir-los.

SenderService.java:

Aquest servei és cridat quan des de l'smartwatch volem enviar una acció cap a l'smartphone. Aquesta acció s'envia per mitjà d'un missatge via el GoogleApiClient. És important veure que al missatge se li pot afegir informació addicional, com ara l'estat desitjat, Id de sensor, etc.

Classes per a representar elements concrets de la UI:

Aquestes classes serveixen per a representar elements de la UI complexos, és a dir, conjunts d'elements gràfics natius d'Android agrupats per a realitzar una estructura més gran. La majoria les he extret del programa existent de l'aplicació Android, ja que d'aquesta manera l'usuari pot fer servir els mateixos patrons perquè el disseny és força semblant. Tot i això, degut a la mida més reduïda de la pantalla, en molts casos no ha estat trivial realitzar aquesta adaptació.

Aquestes classes d'elements concrets de la UI són:

- EnergyButtonView

- HomeButtonView

- ImageLoader (marc negre amb spinner d'animació, prèvia a la càrrega de la imatge).

- InfoView
- InfoViewBigIcon
- RoundedSwitch
- StateAttr

Constants.java:

Aquesta classe conté les claus per a insertar/recuperar els diversos DataItems. També apareix a l'aplicació mòbil.

La resta de classes:

Totes les altres classes presents en aquest paquet són classes de dades, necessàries per a poder transformar les estructures JSON que arriben o que volem enviar des de la API.

- Camera
- ComfortValues
- Sensor

5) PROVES

Pel que fa a les proves, bàsicament es realitzaran testejos reals per a veure que les accions realitzades des del dispositiu tenen els efectes desitjats en els dispositius reals. En aquest cas, al tractar-se de dispositius físics, és fàcil comprovar que tant les dades mostrades són correctes, com veure que si s'acciona a la UI el botó per a apagar una llum, aquesta s'apaga. Per tant, més que realitzar proves a nivell de software intern, es centraran els esforços en veure que el software compleix les funcionalitats desitjades a nivell real.

Lamentablement, degut a falta de temps per a la realització del projecte, no m'ha estat possible endinsar-me en el camp de les proves unitàries (desconegut per mi abans del projecte). A més, tot i la múltiple documentació existent per a Android i JUnit, no hi ha documentació explícita per a Android Wear, fet que va complicar una mica més l'apartat.

Procés de proves

En quant al procés de proves, cal dir que durant el desenvolupament del programari ja s'ha testejat força mitjançant l'emulador natiu d'Android Wear. Aquest simulador connectat un smartphone Nexus 4, permet simular el comportament d'un dispositiu real, i per tant, s'ha pogut comprovar que efectivament el comportament dels diversos elements és l'esperat.

En aquest moment, disposo d'un dispositiu real (Moto 360) adquirit durant el Nadal. He tingut el temps just per a poder comprovar que tot funcionava correctament, exactament igual que a l'emulador. Tot i això vull destacar que degut a un motiu desconegut per mi (i del que no he pogut trobar documentació a Internet), alguns elements de la UI que estaven posicionats en posicions absolutes no estaven al mateix punt de la pantalla a l'emulador i al dispositiu real. Tot i això, he modificat lleugerament aquests valors per a que de cara al vídeo amb el dispositiu real, tot estigui al lloc correcte.

Pel que fa a les proves a nivell de software intern, es realitzaran proves unitàries, ja que és el que demana l'exercici. El framework utilitzat és el que proporciona Android per a testing, **JUnit**.

Des del meu punt de vista, considero que en aquest cas serien molt més útils proves d'integració, ja que són molts els elements interconnectats dins l'aplicació, i amb les proves d'integració asseguraríem el seu funcionament. Tot i això, per falta de temps m'ha estat impossible implementar-les, i per tant, és clarament un dels punts a tractar en una futura ampliació del projecte.

Proves unitaries

Malauradament per falta de temps, només s'ha pogut realitzar una prova unitària i el seu abast és força limitat. D'una banda, la falta de temps per a la realització de proves, ja que s'han allargat altres parts del desenvolupament, ha fet que no pogués dedicar-li tot el temps necessari. A més, mai havia utilitzat JUnit abans, i m'ha costat adaptar-me més de l'esperat.

D'altra banda, cal dir que la documentació de JUnit per a Android Wear és totalment inexistent. He intentat utilitzar la documentació per a Android, i tot i que al principi em va semblar possible realitzar una adaptació, finalment no ha estat així. En general m'he trobat amb dos grans problemes:

- Les activitats que mostren elements per pantalla que no són llistes (SecurityActivity, ControlDeviceActivity, etc.) tenen l'element "WatchViewStub", i no he pogut aconseguir accedir-lo des del test un cop "inflated", motiu pel qual no era possible testejar botons i les seves funcionalitats internes.

- A diferència dels ListView d'Android, el WearableListView no permet fer "performItemClick()", i per tant, no és possible simular el que passaria quan pitges un element de la llista.

Per aquests motius, només he pogut fer un test molt simple. Bàsicament llança la ComfortActivity (llistat amb Inside, Outside i Thermostat, que quan cliques redirigeix a cadascun d'ells), i quan està llançada mira les següents precondicions:

- Que existeixi l'Activity, és a dir, que s'hagi creat correctament.
- Que existeixi la WearableList corresponen al menú d'aquesta pantalla.

A partir d'aquí, realitza un es realitza un "Assert" sobre el nombre de items que té la llista (és l'únic valor que he pogut recuperar del WearableListView), el qual sabem que ha de ser 3. El test funciona correctament, però és evident que no aporta massa als objectius que habitualment tenen els test. Tot i això, estic satisfet perquè almenys he pogut endinsar-me breument a JUnit, i la propera vegada ja no hauré de començar des de 0.

6) CONCLUSIONS

En aquest apartat es presenta la valoració global del projecte, així com l'estat en que ha quedat respecte a les funcionalitats inicials desitjades, identificant les desviacions realitzades i els propers passos a realitzar.

Valoració global del projecte

Per a realitzar la valoració global del projecte, crec que pot ser un bon exercici recuperar els objectius inicials del projecte, i veure en quina mesura s'han completat:

- *Conèixer l'entorn de programació d'Android Wear, i adquirir els coneixements necessaris per a ser capaç de realitzar una aplicació totalment funcional.*

Aquest objectiu era molt engrescador, ja que suposava una nova experiència per a mi en una de les tecnologies més prometedores, ja que els wearables està clar que seran part del nostre futur proper. Estic satisfet amb el resultat obtingut, ja que s'ha obtingut una aplicació totalment funcional, i des del meu punt de vista amb un nivell molt acceptable. L'aplicació a nivell de disseny compleix amb els estàndards i realitza les accions per a les que ha estat dissenyada correctament.

- *Ser capaç d'enfrontar-me a una tecnologia desconeguda, obtenint un resultat positiu al final del projecte.*

Pel que fa a aquest objectiu, està clar que era el gran repte del projecte. En un principi tenia molts dubtes, i fins i tot vaig proposar al consultor un altre projecte alternatiu de menys dificultat. Tot i això vull agrair l'empenta donada per a que em decantés per aquest projecte, ja que si és ben cert que hi han hagut moments difícils (sobretot per la falta de documentació, que feia que problemes trivials costessin una mica de resoldre), la satisfacció final també és més gran del normal al haver obtingut un bon resultat.

- *Millorar les meves habilitats referents al disseny d'aplicacions. Aquest punt engloba tant el disseny previ de l'aplicació (estructura, components, "flow process", usabilitat, etc.), com el propi disseny gràfic i amb codi de la interfície d'usuari.*

Amb el resultat d'aquest objectiu estic especialment content, ja que no és un camp que domini especialment, i després d'esforçar-me crec que el resultat obtingut és molt correcte. Els programadors despreciem moltes vegades aquesta part dels projectes, ja que el que a nosaltres ens agrada és programar i volem començar l'abans possible. Amb aquest projecte he pogut comprovar que un bon disseny facilita enormement la feina posterior, i això fa que la qualitat obtinguda sigui molt superior en un menor temps.

- *Conèixer, tot i que sense entrar en profunditat, noves tecnologies punteres (reconeixement de veu, "geofencing", per exemple).*

De tots els objectius, potser aquest és amb el que he quedat una mica més descontent. D'una banda, he pogut implementar alguns elements de reconeixement de veu, però és cert que a un nivell molt baix (els motius són diversos, i es podria millorar/ampliar en el futur). Pel que fa a altres tecnologies punteres, m'he trobat amb problemes aliens a mi, com per exemple que la majoria d'smartwatch al mercat encara no incorporen GPS, i per tant, no valia la pena dedicar el temps si després no ho podria provar. Tot i això, és evident que aquest punt té un gran atractiu, i m'agradaria explorar-lo en el futur.

- *Augmentar les capacitats de "management" d'un projecte, i realitzar correctament cadascuna de les fases que el composen.*

Finalment, aquest objectiu té un abast molt genèric, però també crec que s'ha assolit correctament. Estic content amb la meva organització durant el projecte, ja que he completat el

projecte seguint pràcticament el calendari exposat a l'inici, i això ha fet que el resultat obtingut fos l'esperat. Un dels aprenentatges que m'emporto sobre management, és que els milestones (en aquest cas eren les PACs) són molt útils, ja que donen un plus de motivació i serveixen com a referencia temporal.

Per tant, resumint tot l'exposat fins ara, la meva opinió personal sobre la valoració final del projecte és molt positiva. A més d'haver obtingut un resultat satisfactori, crec que el que ha estat més important és el que han millorat les meves habilitats intangibles, que de ben segur influenciaran positivament el meu futur professional.

Estat del projecte

En quant a l'estat actual del projecte, crec que el resultat obtingut és molt satisfactori. En primer lloc, cal considerar el gran repte al qual m'enfrontava abans de començar el projecte: la tecnologia Android Wear és molt nova i gairebé no disposa de documentació extensa al respecte, i a més jo no tenia cap experiència prèvia. Per aquest motiu, crec que era una mica complicat poder realitzar una previsió acurada. Tot i això, s'han assolit tots els punts bàsics (com detallaré a continuació), així com la majoria dels punts de nivell avançat. Els objectius de dificultat més avançada no s'han pogut assolir, però en la majoria de casos ha estat per motius aliens a mi, i a més ja s'esperava que no entressin dins de l'abast del projecte.

Més concretament, si revisem el llistat de funcionalitats presentat a l'apartat de **Disseny**, podem veure que:

- ***Funcionalitats bàsiques***: S'han assolit en la seva totalitat.
    - Es reben notificacions des de l'smartphone, i es poden realitzar actuacions/visualitzacions a partir d'elles. També s'han implementat accions directes sobre les notificacions (marcar com a llegida o esborrar), les quals poden fer-se mitjançant veu o tocant la pantalla del rellotge.
    - Es visualitzen els diversos paràmetres de la instal·lació des del rellotge. És possible consultar l'estat de l'alarma, paràmetres de consum d'energia (diari, setmanal i mensual), temperatures actuals (dintre i fora de casa), valor del termòstat, humitat i estat dels sensors elèctrics (ON/OFF).
    - Es pot actuar sobre els diversos dispositius de les instal·lacions. Es pot canviar l'estat de l'alarma, l'estat dels sensors elèctrics i la temperatura objectiu del termòstat.
- ***Funcionalitats avançades***: S'han assolit una gran part d'aquests objectius, i en els casos que no ha estat possible, ha estat per motius aliens a mi. Per exemple, la funcionalitat de Geofencing no l'he codificat de moment, degut a que actualment només un de tots els smartwatch amb Android Wear disponibles incorpora funcionalitats de GPS. És per això que clarament podem considerar aquest aspecte com un dels propers passos del projecte en el futur.
    - Es visualitzen les fotografies al rellotge. En aquest cas, he modificat lleugerament la funcionalitat per a adaptar-la a obtenir la millor usabilitat per a l'usuari. Finalment tan sols es mostra la última fotografia de la instal·lació. Això és degut a que la càrrega de diverses fotografies era massa pesada, i provocava temps d'espera massa llargs que tenien un efecte nociu en la usabilitat (ha estat testejat durant el desenvolupament). A més, si l'usuari desitja això, ja té el mòbil per a consultar-ho, el rellotge està destinat a accions més ràpides i immediates.
    - Es realitzen algunes accions mitjançant reconeixement de veu. En aquest cas ho he limitat a esborrar i marcar com a llegides les notificacions. Tot i això,

també es podria ampliar en el futur a altres parts de l'aplicació, com ara els actuadors.

- ***Funcionalitats fora de l'abast***: Del tercer grup de funcionalitats, les quals ja apareixien fora de l'abast al pla de treball, no se n'ha pogut realitzar cap. Els motius són diversos, entre els quals destaquen la impossibilitat tecnològica, la falta de temps i la gran dificultat. Tot i això, és evident que es tracta de funcionalitats interessants i que es podria considerar la seva incorporació en futurs passos del projecte.

Propers passos

Finalment, i després del que s'ha comentat en aquest apartat, està clar que és molta la feina feta, però no es tracta d'una aplicació tancada i finalitzada, sinó que en el futur pot rebre moltes millores i l'ampliació amb noves funcionalitats.

Des del meu punt de vista, la primera ampliació futura seria afegir funcionalitats GPS ("GeoFencing" i "HomeFencing"). Un cop els smartwatchs incorporin GPS pot ser una funcionalitat molt interessant d'afegir, tot i que com he comentat anteriorment, actualment no l'incorporen.

En segon lloc, crec que també seria interessant millorar les comandes de veu, ja que en aquesta versió són una mica reduïdes (marcar com a llegides i esborrar notificacions). Tot i això, cal dir que no les he incorporat en aquesta aplicació, ja que crec que encara queda camí per recórrer en aquest camp (ampliar els idiomes a més de l'anglès, que és l'únic per al qual funciona de moment), i en aquest moment no m'ha acabat de convèncer com estava muntat tot el relacionat amb el reconeixement de veu. És per això que està molt acotat com a acció després de rebre les notificacions, de manera que siguin només dues opcions a escollir d'una llista.

Finalment, també es podria considerar afegir les funcionalitats que estaven fora de l'abast, ja que és evident que són funcionalitats molt interessants. Des de la meva opinió, poder visualitzar vídeo des del rellotge seria espectacular, tot i que encara l'ecosistema Android Wear no està preparat per a fer-ho fàcilment (he trobat algunes persones que ho feien, però era massa complicat per al resultat obtingut, el qual consumia excessius recursos).

7) ANNEX

Bibliografia

A continuació es presenten els enllaços a la documentació emprada en el projecte:

- https://developer.android.com/training/wearables/notifications/index.html
- https://developer.android.com/training/wearables/apps/index.html
- https://developer.android.com/training/wearables/ui/index.html
- https://developer.android.com/training/wearables/data-layer/index.html
- https://developer.android.com/design/wear/principles.html
- https://developer.android.com/design/wear/patterns.html
- https://developer.android.com/design/wear/style.html
- https://developer.android.com/design/wear/creative-vision.html
- https://developer.android.com/design/wear/structure.html
- https://www.infinum.co/the-capsized-eight/articles/eclipse-is-dead-for-android-development-and-i-helped-kill-it
- https://github.com/erizet/SignalA
- http://azure.microsoft.com/en-us/documentation/articles/mobile-services-android-get-started/