

Aplicació Mòbil per Difusió I Outreach de Missions Astronòmiques I Espacials

Joaquim Calvo Gubianas

Resum— Si analitzem el món de les tecnologies mòbils i el seu recorregut al llarg de la història, podrem observar que han crescut de forma clarament accentuada aquests últims anys. Aquestes circumstàncies han ajudat molts sectors i els han ofert la possibilitat de donar-se a conèixer a través d'aquesta tecnologia, fent-ho de diverses maneres, però principalment a partir d'aplicacions informatives. A partir d'aquí sorgeix la necessitat per part de l'IEEC (de l'Institut de ciències de l'espai. Institució depenent del CSIC, que forma part del consorci del Institut d'estudis espacials de Catalunya. IEEC/ICE-CSIC) de desenvolupar una aplicació de difusió de contingut dels seus projectes, tot implicant diverses necessitats directament relacionades amb el funcionament de l'aplicació, com són la fàcil parametrització de contingut, l'optimització de recursos i la mantenibilitat. Tot aplicant l'enginyeria del software en tots els àmbits del desenvolupament de l'aplicació, es presentarà tot seguit l'arquitectura i desenvolupament aportats per tal de donar resposta a aquestes necessitats.

Paraules Clau: Aplicació, Difusió, Contingut, Parametrització, Projectes, Optimització, Mantenibilitat, Dispositiu mòbil, Android

Abstract- If we look at the world of mobile technologies and its journey in history, we can see that they have clearly grown in recent years. These circumstances have helped many sectors and have offered the possibility of showing its technology to the world, in many different ways, but mainly through informative applications. The IEEC (de l'Institut de ciències de l'espai, dependent institution of CSIC part of the consotrium with Institut d'estudis espacials de Catalunya. IEEC/ICE-CSIC) found the need to develop a spread application of the content of their projects, involving various needs directly related to the operation of the application, such as easy parameterization content, resource optimization and maintainability. The software engineering in all areas of application's development will be presented below providing the architecture and development needed in order to meet these needs.

Index Terms: Application, Difussion, Content, Parametrize, Projects, Optimisation, Maintainability, Mobile Device, Android

1 INTRODUCCIÓ

L'objectiu del present article és presentar la solució per a la implementació d'una aplicació de gestió de contingut per al sistema operatiu Android com a resposta a les necessitats de difusió de contingut per part de l'IEEC.

En l'actualitat podem trobar una gran varietat de Frameworks i entorns de treball per tal de facilitar implementacions de certes solucions software, en podrem trobar alguns exemples més endavant, però no existeix cap solució exacte pel problema plantejat com a base del projecte.

Per tant des de l'anàlisi del problema, les necessitats i altres factors es presentarà un solució aplicada en un cas concret i específic.

Des de una visió general els factors principals els quals han de regir sobre l'estructura de la solució es basen en la facilitat de publicació de contingut i la mantenibilitat del propi software per tal de poder aplicar modificacions de forma continuada, per això s'han tingut en compte els diferents recursos els quals compte l'IEEC i la tipologia del contingut a presentar, que en aquest cas es correspon a contingut científic-tècnic dels diferents projectes que s'estan portant a terme en aquest.

Aquest conjunt de factors i a partir del context en el qual ens trobem, fan que se'ns presentin una sèrie de preguntes :

- Quina és la millor manera de publicar contingut a l'aplicació?
- Quina tecnologia serà necessària utilitzar per tal de satisfer l'usuari final?
- Com ha de gestionar l'aplicació aquest contingut publicat?
- Com s'ha de poder organitzar?

Com es pot observar aquest conjunt de preguntes aboquen en gran part al propi contingut i la seva representació en una estructura de dades que permeti publicar, recollir, administrar i permetre una futura visualització. Tot i això les metodologies emprades per a resoldre els anteriors dubtes també han de satisfer l'usuari final, en els diferents aspectes, tot permetent al desenvolupador crear una aplicació que permeti un fàcil manteniment i escalabilitat en quant a creixement de la infraestructura com també en la introducció de nous tipus de contingut.

Tot seguit presentarem la solució plantejada a partir de l'anàlisi anterior, enfocant la resolució del problema des de diferents perspectives relacionades amb l'enginyeria del software.

E-mail de contacte : Joaquim.calvo.gubianas@gmail.com

Menció realitzada: Enginyeria del Software.

Tutors : Lluís Gesa (UAB) I Josep Guardia (IEEC)

Curs: 2015-16

2 OBJECTIUS

Els objectius perseguits es basen en la proposta de resolució del problema presentat anteriorment a partir de l'aplicació de les diferents metodologies de treball que s'expressaran posteriorment les quals es basen en l'enginyeria del software.

Aquests objectius estan directament relacionats amb els diferents entregables resultants del desenvolupament del treball, els quals es poden representar amb els següents entregables :

- Definició arquitectural de la solució.
- Definició del flux de funcionament.
- Definició de les estructures de dades utilitzades.
- Definició de la metodologia de publicació de contingut.
- Definició de les tecnologies i metodologies d'enviament d'informació.
- Definició del tractament de les dades.
- Definició dels diferents mecanismes d'optimització.

Totes aquests entregables seran presentats conjuntament amb una explicació detallada per tal de veure els raonaments emprats en la presa de decisions.

3 ESTAT DE L'ART

Actualment no trobem solucions exactes al problema plantejat anteriorment, però si que ens podem fixar en les metodologies que s'apliquen en solucions semblants per tal de realitzar aproximacions i adaptacions d'aquestes per tal de poder construir la que ens fa falta en aquest cas concret.

En l'àmbit del software es porta terme, de forma continuada, tasques de recerca i implementació de solucions que permeten l'automatització de tasques i que permetin abstracció en la resolució de problemes complexos. Tot això genera plataformes o standards que aporten valors a solucions posteriors basades en aquestes, com per exemple la reducció de costos, la reducció d'errors, la millora de la gestió i l'ampliació a noves funcionalitats o continguts de manera més controlada i senzilla

En l'àmbit de la gestió de contingut podem trobar diferents frameworks o APIs que ho permeten, la gran majoria orientats a tecnologies web com per exemple Prestashop (orientat al e-commerce i per tant a la publicació de productes) o wordpress. Tots aquests es basen en la parametrització de les seves publicacions a partir d'aplicacions o plataformes que permeten un gran nivell d'abstracció, deixant de banda la lògica i les tecnologies que permeten aquestes gestions.

Tot i això aquests frameworks condicionen, parcialment, punts importants relacionats amb la publicació de contin-

gut i la metodologia que s'ha d'aplicar per tal de que això es produeixi, fent que no es puguin modificar de forma senzilla els diferents fluxos i lògiques de funcionament relacionades amb aquest conjunt de tasques. Per alter banda implementen conceptes o idees base que poden ser extrapolades a alter àmbits per tal de generar una solució més òptima al problema en qüestió. En definitiu trobem una forta dependència en quant a les directrius a seguir al utilitzar aquests frameworks, convertint-se en inviablés en certs casos en els que la base del problema s'allunya molt de la solució que aquests aporten.

En el cas de les tecnologies mòbil no trobem una plataforma o marc de treball clar en quant a aplicacions de difusió de contingut pròpiament, però sí que trobem uns estàndards de desenvolupament i llibreries d'ajuda que poden aportar cert valor en la implementació de tot tipus de solucions de software.

Per tan tens trobem davant d'un panorama de solucions aproximades però majoritàriament orientades a plataformes web i que es s'aproximen en algunes parts, a la solució del problema plantejat.

4 METODOLOGIA

Pel que fa al problema plantejat l'IEEC va demanar una solució per a la seva necessitat d'informar als usuaris dels seus projecte, però a l'hora no hi havia una definició clara de les característiques de l'aplicació a desenvolupar. Per tal de poder satisfer aquestes necessitats informatives es va requerir de les expectatives dels usuaris per a una aplicació d'aquestes característiques. Amb aquest context es va realitzar una roda d'enquestes per tal d'obtenir feedback dels possibles usuaris i així generar una idea més clara de la solució final i així poder-la contrastar amb els stakeholders. Algunes de les dades més rellevants al finalitzar l'enquesta es poden consultar en la figura 1 de l'annex ón es fa un recull parcial d'aquests resultats.

A partir d'aquí es van realitzar reunions periòdiques amb els stakeholders per tal de definir de forma clara els diferents requeriments. Aquests es troben plasmats en el document APPICE-TFG-CR.doc.

A partir de la definició dels casos d'ús i requeriments s'ha procedit a realitzar un anàlisi de disseny de l'aplicació, en qual s'ha definit l'estructura de classes i els diagrames de flux amb els quals s'implementarà la lògica de petició i càrrega de contingut de l'aplicació.

Per a la generació de l'estructura de classes s'han aplicat patrons de disseny per tal de millorar la qualitat de codi, mantenibilitat i eficiència. En aquest cas s'han aplicat els següents patrons de disseny en les següents parts de l'aplicació :

- PATRÓ BUILDER : Mòdul consumidor de XML.

Per naturalesa d'aquest mòdul s'ha implementat aquest patró per a la creació de les diferents instàncies que es representaran en les diferents vistes.

- PATRÓ SINGLETON : Diferents mòduls auxiliars (responsiu, URLResolver, Control de versions, etc..)

La necessitate de no necessitar instàncies diferenciades en diferents parts de l'aplicació, és possible implementar el patró singleton en aquestes classes auxiliars.

- PATRÓ TEMPLATE METHOD: Aplicat en els diferents controladors de les vistes per tal de estipular un fluxe de càrrega de contingut per a cada tipus de vista.

La metodologia aplicada durant el desenvolupament del treball es pot subdividir en diferents etapes o processos, que es troben directament relacionats amb les fases del projecte i els entregables a generar en cadascun d'aquests.

Podem determinar les següents etapes segons la metodologia aplicada :

- **1: Etapa d'Anàlisi I Especificació:** En aquesta primera fase es porta a terme un anàlisi del problema i del seu context per tal d'extreure conclusions i poder fer un desglossament d'aquest en petites tasques/problemàtiques més senzilles de resoldre. En aquesta es formalitza una planificació temporal i la definició de les metodologies a seguir durant el transcurs del projecte. Durant la definició de les subtasques es realitza una planificació dels diferents entregables, els quals es relacionen de manera directa amb aquestes. En l'anàlisi del problema i el context es realitza una investigació de tecnologies actuals que ofereixin solucions semblants o que puguin ser interessants des de algún punt de vista per a la implementació de la solució final. Tot generant el desglossament del problema en tasques, s'ens permet generar una prioritització d'aquestes, deixant com més prioritàries les tasques d'anàlisi, disseny i recerca de tecnologies que permetin generar la millor solució possible. És necessari recalcar que en aquesta primera fase es genera un estudi de viabilitat en el qual s'analitza el problema, un Work Breakdown Structures (WBS) el qual s'especifica el desglossament del problema inicial. Aquesta divisió del problema inicial ens permet un millor seguiment del projecte, ja que es pot analitzar el seu progrés de forma separada. Finalment també es genera un diagram de Gannt en el qual s'especifica la durada i les dependències de les diferents tasques conjuntament amb les fites i els entregables relacionats. Aquest diagram es pot veure en

la figura número 4 de l'annex.

- **2: Etapa de desenvolupament:** A partir de la definició de les bases (requeriments, disseny, tecnologies, etc..) per al desenvolupament del projecte, s'aplica una metodologia de treball en espiral, la qual ens aportara dinamisme en els diferents processos tot donant de manera indirecte fiabilitat en els diferents entregables resultants; per contrapartida ens genera una major càrrega de treball. Prèviament a realitzar les tasques de desenvolupament de la solució final, es genera un prototip de l'aplicació amb la finalitat de tenir una primera presa de contacte directa amb la implementació de la solució i les tecnologies involucrades, fent que al mateix temps, es poguessin descobrir bugs o errades en el disseny plantejat inicialment. Durant les diferents iteracions efectuades en el transcurs de projecte s'han dedicat bona part dels recursos en la recerca de tecnologies, metodologies i estàndards per aplicar en el desenvolupament de la solució final aportant més rigor i valor en aquesta. Dins dels entregables els quals es comprenien en la fase de desenvolupament en espiral es trobaven els propis binaris de software i documentació (que incloïa diagrames UML per a la representació gràfica de certs components) generada en el transcurs dels diferents processos.
- **2.1: Etapa de Testing:** Podem tractar aquesta com una sub-etapa del desenvolupament, la qual transcorria de forma paral·lela a aquesta. Es realitzaven conjunts de test unitaris a partir de la llibreria JUnit i també testing dels components de la interfície gràcies a "Espresso". Alhora es procedien a realitzar tests per inspecció amb la finalitat de trobar bugs tant en el propi codi font com en la usabilitat de l'aplicació. Tots els errors o modificacions seran replantejats en les següents iteracions per a la seva posterior implementació. Per tal de tenir un control dels bugs trobats durant les diferents tasques de testing, aquests eren plasmats en un document per tal de planificar la seva futura implementació en la següent iteració de desenvolupament.
- **3: Etapa de revisió:** En aquesta etapa es porten a terme l'avaluació dels diferents resultats obtinguts durant el transcurs de les dues etapes anteriors, fent una avaluació dels graus de satisfacció del problema plantejat inicialment amb contrapartida de les solucions plantejades i els entregables corresponents. Aquests processos d'avaluació es porten a terme de forma separada tot diferenciant els diferents aspectes que engloba el problema inicial.

Per tal de portar a terme de manera controlada i or-

ganitzada el projecte s'ha utilitzat l'entorn de desenvolupament (IDE) android studio i bitbucket (Git) com a repositori o control de versions. Per altra banda tot el material documental generat també ha servit per a l'estructuració i seguiment dels diferents processos o subtasques involucrats en el correcte desenvolupament del projecte.

Seguint el procés d'enginyeria del software i amb el propòsit de controlar els diferents aspectes involucrats en el desenvolupament del projecte s'han generat de manera progressiva els documents pertinents els quals, cadascun d'aquests, s'orienta a propòsits diferents.

Per tal de poder organitzar-los correctament s'ha seguit la següent nomenclatura en el nom d'aquests:

APPICE-TFG-<INICIALS DEL DOCUMENT>

En el cas del document de disseny obtindríem el següent nom : APPICE-TFG-DD.

Els documents més rellevants generats durant el transcurs del projecte són :

- (EV) Estudi de viabilitat.
- (CR) Document de captura de requeriments .
- (DD) Document de disseny .
- (INF1-INF2) Documents de control 1 I 2.

El fet d'implementar un sistema de navegació jeràrquic i aplicar mecanismes que permeten una interfície user-friendly , generen un sistema senzill i intuïtiu d'utilitzar. Fent que sense necessitat d'un manual o tutorial previ, es pugui utilitzar l'aplicació sense cap problema. Aquesta característica o capacitat era un dels requisits extrets a partir de les enquestes realitzades al inici del desenvolupament del projecte. Cal detallar que en la majoria de documents podrem trobar diagrames UML els quals pretenen millorar la lectura i comprensió de les diferents explicacions ja que aquests aporten un suport gràfic molt útil, els més destacables són els següents :

- Diagrama de casos d'ús.
- Diagrames de fluxe.
- Diagrames de components o capes
- Diagrames de classes
- Work Break Down Structure

Aquest últim s'ha utilitzat per tal de fer una avaluació del projecte a partir de les diferents tasques que s'hi defineixen, tot donant a aquestes un nom a partir d'una nomenclatura determinada, la qual es pot trobar definida en el document "APPICE-TFG-INF1".

Tota la documentació, diagrames i altres elements involucrats s'han gestionat en un repositori diferent al utilitzar per al control de versions del codi font del projecte. El fet de tractar-se de dos artefactes o entregables diferents s'ha procedit a utilitzar aquesta metodologia.

5 ENTREGABLES

Com s'ha comentat anteriorment els resultats obtinguts es presenten en forma d'entregables els quals es troben relacionats amb els diferents problemes o necessitats plantejades durant el període d'avaluació del problema inicial. Aquests es comprenen en dos grans conjunts:

- Documentació
- Software

A partir d'aquests es dona una solució treballada a partir de l'anàlisi del context, el qual es pot analitzar des de els següents punts de vista:

- Estructures de dades.
- Publicació de contingut
- Comunicació
- Tractament de les dades.
- Mecanismes d'optimització
- Tecnologia emprada.
- Desplegament.
- Avaluació d'alternatives

5.1 Arquitectura del Sistema

5.1.1 Part Servidor

A partir de l'anàlisi de solucions emprades en altres entorns es va poder observar la necessitat d'emprar mecanismes d'automatització ens les diferents operacions implicades en les publicacions de contingut, fent que a l'hora, aquestes responguin a les necessitats de les persones encarregades o responsables d'aquestes publicacions.

Aquest mecanisme permet posicionar a l'administrador del contingut a un nivell d'abstracció superior, ja que no s'implica en la lògica ni els processos interns els quals permeten la publicació del contingut, possibilitant al mateix temps, la desvinculació de tota aquesta lògica de les tecnologies que ho permeten.

Quan ens referim a la infraestructura del sistema de publicació ens referim de manera clara a una aplicació que permeti, mitjançant una interfície gràfica, a l'administrador de l'IEEC corresponent manipular els diferents aspectes d'aquest.

Com podem observar en la figura 1, aquesta diferenciació ens proporciona un desacoblament entre les diferents parts involucrades en el procés de publicació, fent que es puguin introduir modificacions en certes parts sense tenir afectacions parcials o totals en altres artefactes o parts del sistema. Tot això involucre una millora substancial en quant a mantenibilitat i la implementació de noves funcionalitats o ampliació de nous continguts.

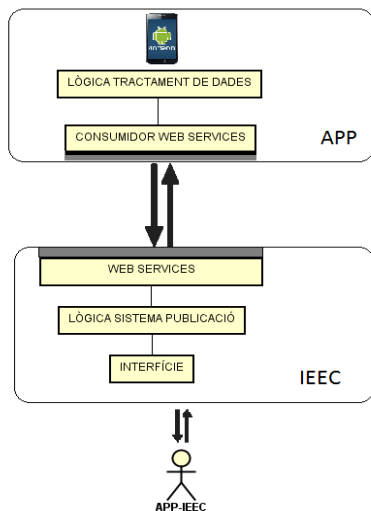


Fig.1: Esquema general arquitectura de la solució

Si fem un seguiment de les capes que conformen el sistema ens trobarem amb la part encarregada de la transmissió de la **informació o contingut**. En aquesta s'ha de tenir en compte les estructures o models de dades emprades per aquesta i com l'aplicació haurà de accedir i consumir aquestes dades. En aquest cas concret es treballarà mitjançant web services que operaran amb fitxers XML els quals ens permetran definir de manera clara i concisa la informació a transmetre. Aquesta tecnologia ens permetrà un fàcil implementació dels processos generadors i administradors de contingut, facilitant la incorporació de noves tipologies de contingut per publicar. Alhora ens permet la generalització en el consum d'aquesta informació, gràcies al gran suport que té aquesta tecnologia en diferents llenguatges i plataformes.

També cal destacar que gràcies al ús de fitxers de XML per a la transmissió, permet l'emmagatzematge d'aquestes de forma directa en memòria secundària per al seu posterior consum, fent que no siguin necessaris altres mètodes de persistència com la serialització.

Tot seguint la línia de l'arquitectura plantejada els web services permeten un desacoplament entre els serveis que s'ofereixen i els consumidors d'aquests, generant menys dependències i permetent aplicar modificacions de manera més senzilles en les diferents parts sense tenir implicacions no desitjades.

En general ens aporta facilitats en el manteniment, escalabilitat, simplicitat en la implementació i suport per a una gran varietat de tecnologies permetent estendre l'ús d'aquest Sistema de publicació. Podem veure de manera gràfica el funcionament anteriorment plantejat en el diagrama de la figura 2.

El fet d'utilitzar aquesta metodologia conjuntament amb peticions sense estat (stateless) permet la implementació de sistemes de balanceig de càrrega en sistemes clusteritzats, oferint una major disponibilitat i una millor resposta en casos d'error o incidències.

Els avantatges que ens aporta el treball amb fitxers XML:

- Pot ser fàcilment interpretat per a persones.
- Separa radicalment el propi contingut de la seva presentació o format.
- Pot ser utilitzat en qualsevol llenguatge o alfabet
- Permet una estructura jeràrquica
- Gran suport en diferents llenguatges de programació API's tractament de fitxers XML

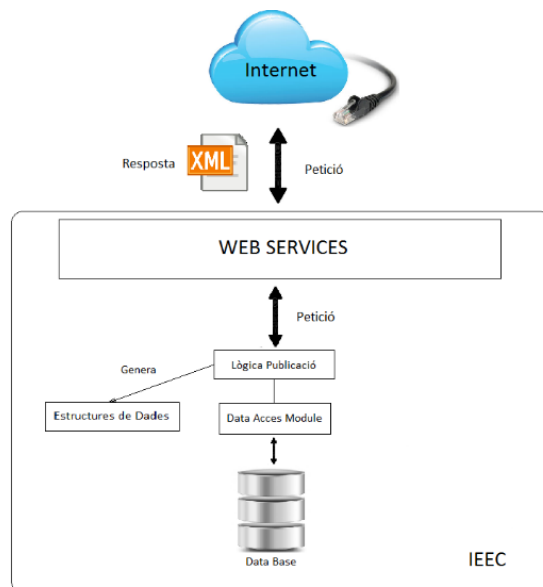


Fig.2: Funcionament lògic publicació mitjançant web services

5.1.2 Part Aplicació

L'aplicació s'ha dividit en diferents pantalles per tal de poder mostrar la informació rebuda de les crides als serveis corresponent. La seva composició es basa en una estructura jeràrquica molt senzilla, per tal d'oferir un sistema de navegació intuïtiu i ja conegut per part de l'usuari. En aquesta podem trobar les següents pantalles:

- Pantalla inicial de projectes
- Pantalla de seccions d'un projecte
- Pantalla d'una secció concreta d'un projecte
- Pantalles derivades de la secció

Podem veure algunes captures de pantalla de les diferents vistes de l'aplicació en la figura 4 de l'annex.

En la primera pantalla podrem trobar de manera diferenciada el conjunt de projectes els qual el IEEC està portant a terme dels quals vol informar. A partir d'aquí podem seleccionar un projecte per tal de veure el seu contingut. Els diferents tipus de contingut publicat es troben definits

en seccions agrupades en un menú amb icones que les representen. Inicialment les possibles seccions que pot tenir un projecte són:

- Descripció del projecte
- Fotos
- Vídeos
- Notícies relaciones

Gràcies a l'arquitectura de l'aplicació, es poden ampliar les tipologies de dades les quals pot difondre l'aplicació.

Cada una d'aquestes seccions pot incorporar diverses pantalles que l'hi seran útils per a representar el seu contingut.

Entrant en la part més tècnica i de gestió de la informació, serà necessari que l'aplicació implementi un mecanisme per consumir les dades del servidor i un sistema que implementi tota la lògica per a mostrar el contingut rebut a l'usuari.

En l'aplicació seguim també, una metodologia modular. En aquesta s'ha implementat tot el model de gestió de les dades en forma d'algoritme modular, a partir del patró de disseny "Template Method" fent que cadascun dels diferents mòduls realitzi una part específica de tota la lògica de funcionament i aquests puguin ser fàcilment substituïbles o modificats. Aquest conjunt de mòduls el conformen :

- Mòdul peticions web service.
- Mòdul consumidorXML
- Mòdul consumidor d'imatges.
- Mòdul de presentació de contingut
- Mòdul de control de versions
- Sistema cache.
- URL resolver

5.2 Definició Mòduls Aplicació

5.2.1 Mòdul peticions web service

Aquest mòdul implementa tot el sistema de peticions als diferents serveis web per tal de rebre el contingut corresponent. A partir de les diferents vistes de l'aplicació es generaran peticions de contingut, condicionades a les accions del usuari. A partir del context de la petició (vista des de la qual es realitza), de l'acció de l'usuari i de les propietats d'idiomes definides, es generarà la petició corresponent que donarà com a resultat un fitxer XML amb el contingut, el qual serà transferit a la vista corresponent per tal de que aquesta pugui generar el seu contingut.

Ahora es pot aprofitar el format XML de resposta, per tal de transferir certes propietats a les vistes com per exem-

ple colors, mides i altres aspectes relacionats amb els diferents elements gràfics que les conformaran.

Referent al tema de l'idioma, aquest serà recollit a partir les propietats definides per l'usuari que es trobaran en el fitxer de propietats corresponent. Aquesta metodologia permet externalitzar la lògica del mult-idioma als propis serveis web i deixant únicament per l'aplicació, el tractament de certes parts dels menús per tal d'implementar aquesta funcionalitat.

5.2.2 Mòdul Consumidor XML

Aquest mòdul implementa tota la lògica de creació d'instàncies que s'utilitzaran per part de la vista per tal de mostrar el contingut publicat.

Com el seu nom indica, aquest rebrà contingut, provinent del mòdul de peticions a web servies, el qual estarà en format XML. Per tal de simplificar la lectura i la creació de les instàncies definides en el fitxer, la lectura es basarà en la tipologia de la petició que hagi donat com a resultat les dades obtingudes. D'aquest manera sabrem de manera precise quina tipologia de dades necessitem instanciar i quins tags hem de reconèixer durant les tasques de parsing. El fet d'implementar aquesta lògica ens permet agilitzar molt el procés de lectura, convertint aquest en un dels processos més lleugers de tota la lògica.

En una primera fase el mòdul recollirà el fitxer XML provinent de la crida als serveis web realitzada en la vista anterior, tot seguit s'identificarà el tipus de petició que pot ser:

- Projectes IEEC
- Seccions d'un projecte
- Fotos d'un projecte
- Vídeos d'un Projecte
- Notícies d'un Projecte
- Altres

Es poden veure screenshots de les vistes anteriorment esmentades en les figures número 5, 6 i 7 de l'annex.

A partir d'aquí s'aplicarà un patró de lectura concret tot depenent de la tipologia. Aquesta lectura es realitza amb l'ajuda de la interfície XMLPullParser que ens ofereix mètodes que faciliten el procés de parsing.

Durant aquestes tasques de parsing es creen i es setegen els valors de les diferents instàncies que es necessitaran per a la representació de les vistes.

Com s'ha comentat anteriorment tota l'estructura d'estructures de dades s'ha construït de manera arborescent per tal de facilitar futures ampliacions de contingut com també la generalització d'algunes tipologies de dades per tal de no ser replicades en totes les seves subestructures.

Podem veure el plantejament inicial per al projecte en el següent diagram de classes:

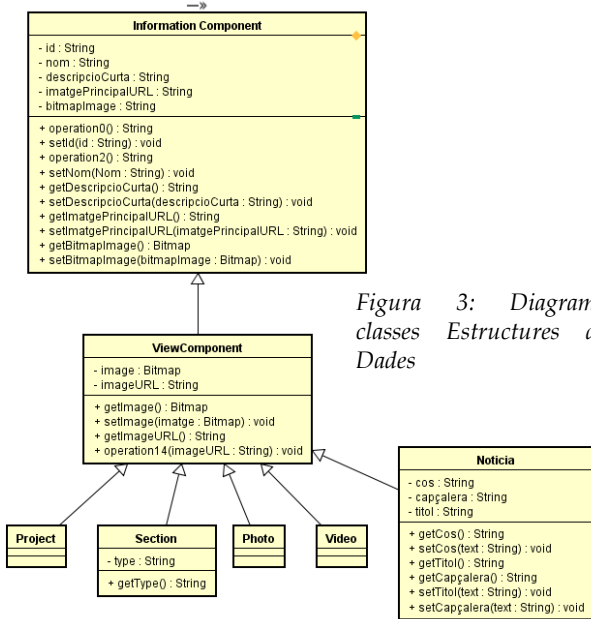


Figura 3: Diagrama Estructures de Dades

El fet d’aplicar mecanismes d’especialització i herència d’atributs ens aporta la capacitat de generar agrupacions per tipologies de contingut a partir de l’abstracció de dades o estructures de dades comunes, alhora ens permet realitzar canvis genèrics de forma més senzilla.

Per tal de poder llençar les tasques consumidores de XML conjuntament amb les de càrrega de vista s’han tingut que definir aquestes com a tasques asíncrones que es puguin executar en altres “threads” per tal de paral·lelitzar i balancejar la càrrega de treball.

5.2.3 Mòdul Consumidor d’Imatges

La gran majoria dels menús de l’aplicació content imatges relacionades amb el contingut publicat. Aquestes estructures de dades s’han definit dins la seva jerarquia de classes com a viewComponent, en canvi la resta com a informationComponent, les quals es poden veure en la figura 3.

Aquestes imatges però, no són enviades a partir de serveis web. Dins dels fitxers XML provinents de les crides a aquests serveis trobarem tota la informació i meta-informació del contingut que s’haurà de publicar com també camps específics per a les imatges, en els quals únicament trobarem URL’s totals o absolutes on podrem trobar les imatges necessàries.

El mòdul consumidor d’imatges entra en joc posteriorment al mòdul consumidor de XML. En aquest es generen les instàncies relacionades amb els elements de la vista, les quals si necessiten d’una imatge tindran variables

relacionades amb la URL i la pròpia imatge. El mòdul rebrà un llistat d’instàncies de la vista pendents d’obtenir la seva imatge, i a partir de tasques asíncrones s’obtindevan les seves imatges mitjançant peticions GET HTTP, seran posteriorment processades i setejades en les seves corresponents instàncies de la vista a partir d’estructura del tipus Bitmap.

Pel que fa a les URL’s trobem el mòdul URLResvoler, el qual ens permetrà validar la URL si ve completament informada en el camp anteriorment comentat o bé generar-la. El procés de generació de la URL dependrà segons la vista des de la qual generem la petició de contingut. Aquest mòdul serà presentat en detall posteriorment en el document.

Per altre banda en quant a gestió d’imatges i millora de l’eficiència en la càrrega de les vistes, ens trobem el mòdul de cache, el qual s’encarrega d’emmagatzemar en una memòria temporal, les diferents imatges que es van carregant Durant l’execució de l’aplicació, tot millorant els temps de càrrega com també el consum de dades i de recursos en general. Aquest serà presentat en detall posteriorment.

En global i amb el conjunt del tres mòduls, tenim un sistema que ens permet consumir, tractar i mostrar les diferents imatges que seran utilitzades en l’aplicació. El fet que no s’utilitzin llibreries o frameworks externs ens permeten gestionar i parametritzar per complet tots els aspectes relacionats amb aquestes tasques.

5.2.4 Mòdul de Presentació de Contingut

Aquesta part de l’aplicació actuarà de controlador si ho miréssim des de la perspectiva MVC. Aquest mòdul el component el conjunt de classes que es troba directament relacionades amb les diferents vistes de l’aplicació i són les que s’encarreguen de realitzar les crides de tota la lògica de flux de càrrega i tractament de dades que ha de seguir l’aplicació per tal de que es pugui mostrar el contingut corresponent de forma exitosa.

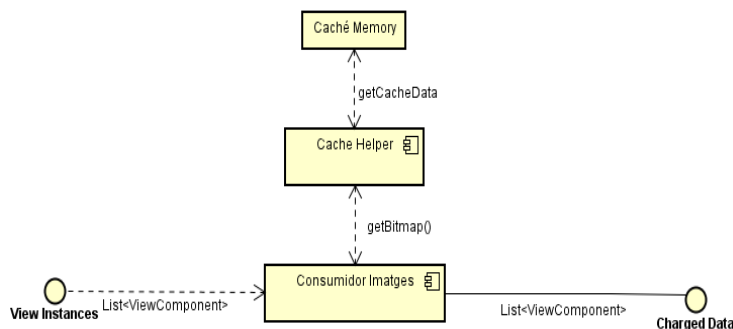


Fig.4: Mòdul consumidor d’imatges

Ja que cada vista requerirà diferents tipologies de d'estructures de dades i de continguts molt diferents, serà necessari que aquestes necessitats siguin contemplades i per tant s'ha implementat tota aquesta lògica directament en les classes que es relacionen amb les diferents "Activities" de l'aplicació, podent diferenciar aquestes segons el context de la vista des de la qual ens trobem.

Podem trobar el funcionament d'aquest mòdul en la figura 2 de l'annex.

Dins d'aquest mòdul també es parametrizaran, de forma parcial, alguns components de la vista per tal de poder adaptar-los a diferents resolucions de pantalla. Per tal de poder portar aquesta tasca trobem l'ajuda d'un mòdul auxiliar que permet obtenir les diferents resolucions i característiques de la pantalla del dispositiu en el qual s'estigui executant l'aplicació.

Cal comentar que totes les classes que només donen suport o funcionalitat parcial en algun altre component, apliquen el patró singleton, ja que no és necessari la creació de diferents instàncies d'aquests donades les seves característiques.

Per tal d'agilitzar els diferents processos que es generen a partir d'aquest mòdul, aquests són llençats de manera asíncrona i en diferents threads per tal de poder paral·lelitzar la càrrega de treball i obtenir un millor resultat, tot reduint els temps d'espera.

Posteriorment s'explicaran algunes variacions en la implementació d'aquest mòdul.

5.2.5 Mòdul de Control de Versions

Aquest mòdul és l'encarregat de controlar el contingut que es publica en l'aplicació a partir d'un control de versions. Aquesta metodologia ens permet generar notificaciones (notificaciones Android) quan trobem que el contingut publicat en el web service és d'una versió diferent (superior) a la que es troba publicada en l'aplicació.

Per tal de que aquesta tasca no afecti al transcurs natural de l'aplicació, s'ha definit tota la funcionalitat en un procés que s'executa en segon plà, el qual va realitzant comprovacions periòdiques. Per a poder procedir amb aquestes comprovacions és necessary que hi hagi un control intern en el propi aplicatiu referent a la versió del contingut, per això ens suportem del mòdul de gestió de memòria, el qual és l'encarregat de gestionar els fitxer de "properties" que es troben emmagatzemats en aquesta. Aquests fitxers bàsicament contenen les diferents propietats que l'usuari pot modificar a partir de la vista corresponent com també el número de versió del contingut publicat fins a l'última actualització d'aquest.

Podem veure el funcionament del sistema de control de

versions en la figura 3 de l'annex.

Pel que fa al sistema de gestió de memòria treballa a partir de tags, permetent l'inclusió de noves funcionalitats, en quant a propietats de control, de manera senzilla. Aquest mòdul s'explicarà més en detall posteriorment.

5.2.6 Mòdul d'emmagatzematge en Caché

En la implementació del sistema de cache s'han volgut abarcar dos objectius, millorar el rendiment de l'aplicació i a l'hora intentar no consumir massa memòria en el dispositiu.

A l'hora aquest sistema de cache es pot aplicar a diferents elements de l'aplicació i aplicar diferents implementacions.

Els elements els quals es podria aplicar serien els que més transit de dades generen. Durant l'execució de l'aplicació com són els fitxers XML de contingut, els quals provenen dels web services i les imatges carregades directament per petició HTTP. Entre aquests dos, com es evident, el que més volum genera són les diferents imatges i tot depenent de la magnitud dels continguts publicats es pot transformar en un problema el qual s'ha de tractar.

A partir d'aquí podem trobar dues possibles implementacions per aquest tipus de problemes. Utilitzar un sistema de cache volàtil, el qual mantingui les dades que es vulguin guardar en estructures de memòria temporal que tenen un scope temporal, fent que aquestes s'omplin de contingut a mesura que aquest es va visitant i s'esborri un cop el sistema operatiu requereixi de més memòria principal.

O bé utilitzant un sistema d'emmagatzematge permanent com la memòria secundària.

En el primer cas s'ha creat un mòdul auxiliar implementat a partir del patró singleton que conté una estructura de dades del tipus LruCache, la qual es basa en una cache que implementa l'algoritme "Last Recently Used" i que permet ser utilitzada amb una interfície semblant a una HashList a partir d'esquemes clau-valor. En el cas de les imatges s'utilitzaria la pròpia URL de la imatge a carregar com a clau i una estructura de dades Bitmap com a valor, permetent una relació inequívoca entre aquestes.

Aquesta metodologia té alguns inconvenients ja que té limitacions amb quant a espai de memòria màxim de la cache i la necessitat de trobar un valor correcte d'aquest per tal de no provocar un excés moviment de dades dins d'aquesta ni tampoc errors per faltes de memòria.

5.2.7 Sistema de Memòria Secundària

Aquest mòdul ens permet gestionar fitxers en la memòria interna del dispositiu, en aquest cas ens permet gestionar el fitxer de propietats en el qual, com el seu nom indica, permet emmagatzemar les propietats que pot parametritzar l'usuari des de la vista correspondent.

Dins d'aquests paràmetres hi podem trobar :

- Llenguatge del contingut publicat i dels menus.
- Mida de la lletra
- Versió del contingut que s'està publicant en el dispositiu. (Control de versions)

Aquest sistema està implementat a partir de les funcions que ens ofereix les classes de gestió de context del propi entorn Android, aquestes permeten el maneig del directori assignat a l'aplicació, aquest es troba dins del mateix directori d'instal·lació de l'aplicació, més concretament en la següent ruta dins del propi dispositiu:

```
/data/data/<package de l'aplicacio*>/files/*
```

La classe implementada utilitza el patró singleton i ens permet la creació, borra i obtenció de les estructures de dades necessàries per a la seva posterior modificació.

El fet de que el mòdul sigui general i no especialitzat en una tipologia de dades concreta, ens permet la seva futura utilització per a altres afers, com per exemple caching d'imatges a nivell de persistència en memòria, persistència dels fitxers XML de publicació de contingut o altres elements que es puguin incorporar en futurs desenvolupaments.

5.3 DESPLEGAMENT

Per a poder utilitzar, depurar i testejar l'aplicació s'han utilitzat dues metodologies de desplegament. Inicialment mitjançant el mòdul "Android Virtual Devices" (ADV) que es troba integrat en el entorn de desenvolupament "Android Studio" i que ens permet generar instàncies virtualitzades d'un dispositiu Android, puguent desplegar dispositius mòbils, tablets amb diferents mides de pantalla, diferents versions del sistema android, etc.. La principal problemàtica que presenta aquest mètode és la lentitud del propi desplegament, tot i que aquest pot variar segons l'equip en el qual es realitzi. El fet d'utilitzar un sistema amb processador Intel ens permet l'utilització el motor HAXM millorant els temps desplegament.

Posteriorment el desplegament es va fer a partir de dispositius reals configurats per admetre depuracions d'aplicacions mitjançant una connexió USB. En aquest cas l'únic contratemps relacionat és la instal·lació dels drivers MTP corresponent al dispositiu on es realitza el desplegament per tal de que el IDE el reconegui com a tal. El fet

d'utilitzar aquest segon mètode ens aporta més rapides en el desplegament i per tant més agilitat en el desenvolupament associat.

5.4 Testing Unitari i Interfície Mitjançant Espresso

Per tal de provar tal d'assegurar que la definició de la lògica i el flux de funcionament en la càrrega de contingut és correcte s'han generat diferents casos de test intentat involucrar les diferents situacions que es poden generar en aquest context. Per això s'ha utilitzat la llibreria JUnit fent test unitaris als mòduls corresponents :

- Mòdul de càrrega d'imatges.
- Mòdul consumidor XML
- Mòdul generador de URL's
- Mòdul de vistes responsives.

Per la part de generació i comportament de les vistes s'ha utilitzat la llibreria "Espresso", aquesta ens estalvia la complexa necessitat de controlar els diferents fils d'execució i prepara el context general pel testeig de l'interfície. Per tal d'utilitzar-la únicament necessitem afegir la seva dependència en el fitxer de configuració Gradle corresponent.

En aquests test s'ha comprovat si la lògica de creació de les vistes es correspon al comportament plantejat en el seu disseny.

Els tests es creben paral·lelament a les tasques de desenvolupament permetent assegurar la correcta implementació de la lògica definida en el disseny, alhora donant seguretat a les futures implementacions o modificacions.

Pel que fa referència a les dependències entre els serveis web i la pròpia aplicació s'ha tingut que resoldre de diferents maneres.

En les primers etapes del projecte i del propi desenvolupament de l'aplicació el consum de les dades es realitzava de manera local ja que no es disposava de cap servidor en el qual dipositar les dades, per tant es va decidir realitzar la recollida de forma directa en fitxers definits dins d'una estructura d'assets. Per tal d'implementar-ho es va implementar un mock object que simulava la implementació del mòdul consumidor dels serveis webs. Aquest mock object únicament realitzava una lectura d'aquests fitxers de continguts definits en l'estructura d'assets del projecte.

Posteriorment la recollida del contingut es va portar a terme a partir d'un servidor, però la falta de serveis en aquest va portar a la modificació del mock object prèviament introduït per tal de que, en aquest context, recollits els fitxers de contingut directament del servidor amb una petició HTTP tot simulant una petició al servei.

En línies futures del desenvolupament del projecte seria interessant la introducció de mètriques que permetessin evaluar de manera numèrica certs aspectes dels diferents test com per exemple el percentatge de cobertura, efectivitat (test completats satisfactòriament), total d'errors.

5.5 VARIACIONS EN LA IMPLEMENTACIÓ

Per a la implementació de la càrrega d'imatges hi ha diverses API que ens faciliten aquesta tasca. El fet de no utilitzar-les es basa en la meua pròpia curiositat per aprendre més respecte el desenvolupament Android i pel fet de no dependre de altres llibreries per als processos de càrrega de les imatges del contingut.

Per tant amb la implementació local de la lògica de càrrega de fotos ens permet realitzar modificacions d'aquesta de manera més senzilla i sense tenir en compte dependències externes.

Algunes de les APIs més conegudes per a realitzar aquestes tasques són les següents:

- Picasso
- Volley (gestió de peticions HTTP)
- Universal Image Loader

Una altra variació la podem trobar en el sistema notificacions, en aquesta podem optar per notificacions "push". Aquestes basen en la figura d'un servidor per tal d'iniciar la comunicació i notificar a l'usuari. En aquest cas és necessari un servidor per tal de que s'encarregui de portar a terme aquesta tasca. Aquest tipus de notificació ens permet implementar mecanismes més òptims de notificació els quals consumeixen menys recursos.

En aquest cas no s'ha implementat aquest tipus de solució ja que inclou més complexitat en el desenvolupament i és necessari registrar un servei específic en un servidor extern.

En futures línies de l'aplicació seria interessant implementar aquest mecanisme per tal de millorar el rendiment de l'aplicació,

6 CONCLUSIONS

La solució generada com a resultat del projecte compleix els diferents requeriments i fites proposades en les fases inicials del projecte. El fet de que aquest solució s'hagi implementat seguint una filosofia modular i generalitzant tots els seus components permet que pugui ser utilitzada en altres aplicacions de difusió de tot tipus de contingut.

Pel que fa al seguiment del projecte s'han complert totes les fites i objectius proposats al inici del projecte, però seria necessari invertir més temps per tal de poder acabar de perfeccionar alguns dels mòduls de l'aplicació i poder implementar un estil de vistes més curos i detallat.

Personalment ha sigut una satisfacció poder portar a

terme un projecte d'aquestes característiques ja que m'ha permès aprendre les tecnologies emprades en el projecte i veure la dificultat implicada en l'estimació de les diferents tasques d'un projecte de software.

Tot i això m'haguess agraït completar, en la seva totalitat, el desenvolupament de l'aplicació i tot el software de la part de back-end per tal de veure els resultats aplicats en un entorn de producció, però per limitacions temporals no ha sigut possible. Com a motivació personal i mirant el projecte en línies futures, seria interessant portar a terme aquesta tasca, ja que pot resultar útil per a resoldre necessitats semblants en altres entorns, fent que realment es pugues comprovar la possible utilització de la solució plantejada en aquest article en altres àmbits.

7 BIBLIOGRAFIA

[1] Tutorial Desenvolupament aplicació Android, Documentació Oficial [Online] Disponible : <http://developer.android.com/intl/es/training/basics/firstapp/index.html>

[2] Patrons de disseny Java [Online] Disponible : <http://migranitodejava.blogspot.com.es/search/label/Destructorator>

[3] Teoria de patrons de disseny, Assignatura Disseny del Software [Online] Disponible: <https://cv.uab.cat/portada/index.html>

[4] Funcionament LRU Cache Android, Documentació Oficial [Online] Disponible : <http://developer.android.com/intl/es/reference/android/util/LruCache.html>

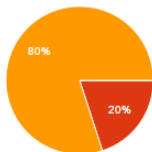
[5] XMLPullParser Documentació Android [Online] Disponible: <http://developer.android.com/intl/es/training/basics/network-ops/xml.html>

[6] Testing Android Oficial [Online] Disponible: http://developer.android.com/intl/es/tools/testing/testing_android.html

[7] Test Interfícies Android amb Espressos [Online] Disponible: <http://www.stevenmarkford.com/android-ui-testing-with-espresso-basics-tutorial/>

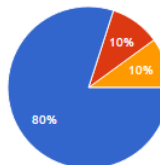
ANNEX

Com creus que hauria de ser el tipus de contingut?



Tipus de contingut	Quantitat	Porcentatge
Únicament científic	0	0%
Adaptat a tots els públics	2	20%
Una mica de tot	8	80%
Otro	0	0%

Esperaries trobar alguna secció amb contingut del propi centre ?



Resposta	Quantitat	Porcentatge
Sí	8	80%
No	1	10%
Otro	1	10%

Quins creus que serien els apartats o seccions que hauria de tenir l'aplicació

aquests que surten a l'exemple i també un apartat on s'expliqui la informació que es mostra

Curiositats, moltes curiositats.

Notícies, agenda, participa, curiositats, informació de l'institut.

Notícies, vídeos, curiositats, imatges, novetats, enquestes

Notícies Curiositats Entrevistes a científics Projectes (antics i de futur) Vídeos científics Articles científics d'alt impacte

XAFARDERIES, VIDEOS

Novetats Notícies Espai personal Concursos Investigació i recerca ...

Quin tipus d'aplicació t'esperes?

Una aplicació de fàcil ús.

Una aplicació que realment representi l'espai.

Una bastant visual on poder consultar fàcilment els projectes.

Resolutiva i molt categoritzada

Una aplicació relacionada amb el món de la ciència adaptada a tots els públics, tant científics com no, on la gent pugui conèixer els projectes duts a terme i els futurs projectes, on la gent pugui veure notícies relacionades i pugui consultar les preguntes freqüents.

Aplicació simple. No carregar en excés de botons, apartats. La informació i més accessible possible. Bona jerarquia i ordre que permeti trobar allò que busques sense coneixement previ del que vols trobar. Que aportés alguna cosa diferent a la que es pot consultar a la web. P.ex Un seguiment de projectes o participació interactiva en projectes o informacions dels projectes per part dels usuaris que se l'han descarregat

Figura 1: respostes més representatives de l'enquesta realitzada en les primeres etapes del projecte.

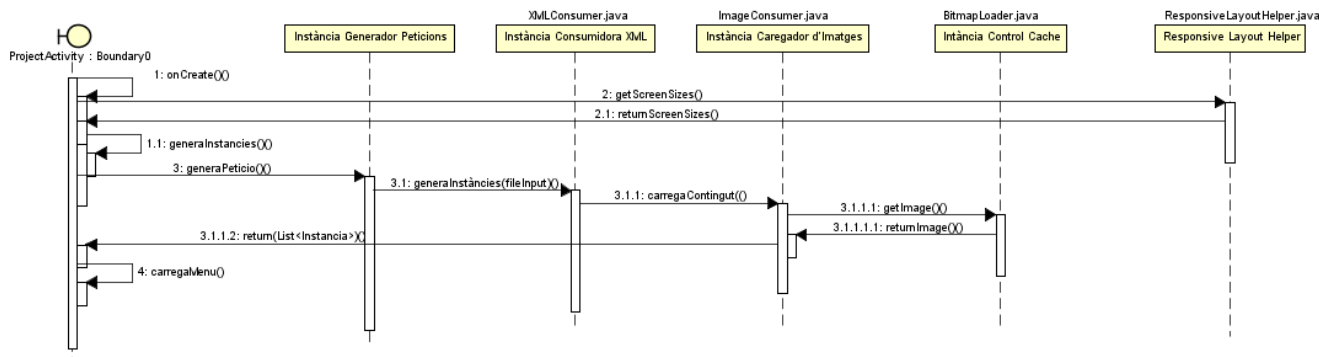


Figura 2 : Diagrama de fluxe que defineix el funcionament la part de presentació de contingut.

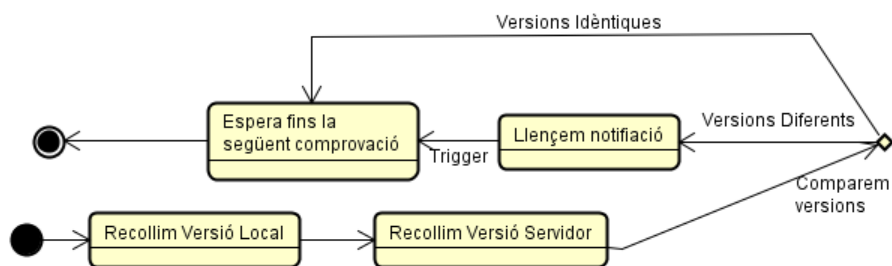


Figura 3: representació del fluxe de funcionament del mòdul de control de versions.

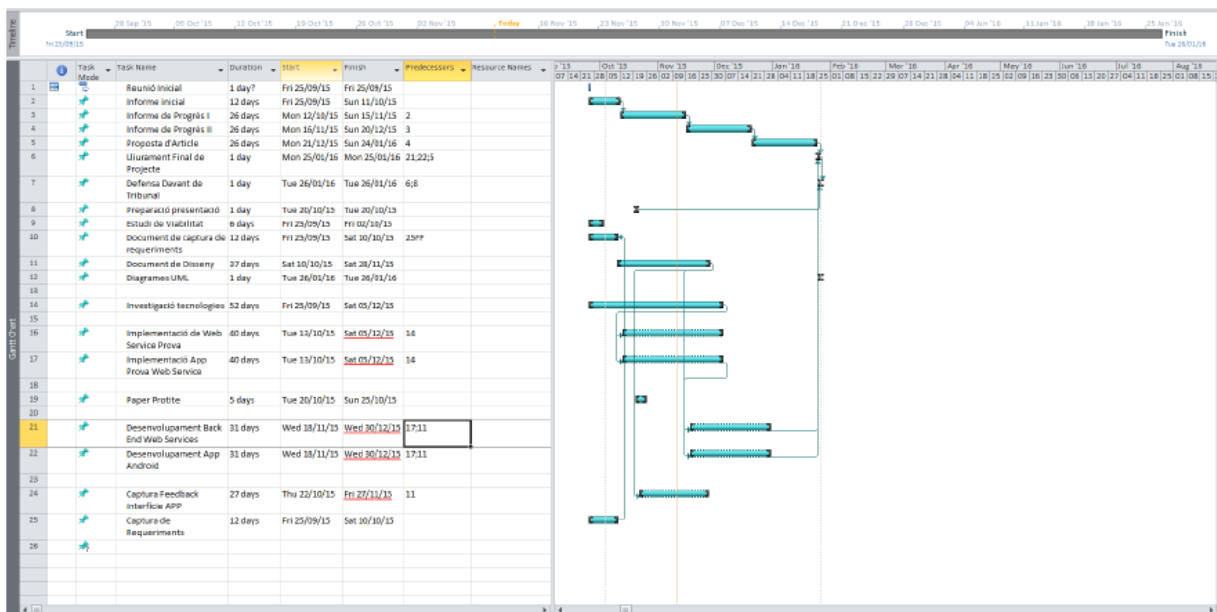


Figura 4: Diagrama de Gannt del projecte.

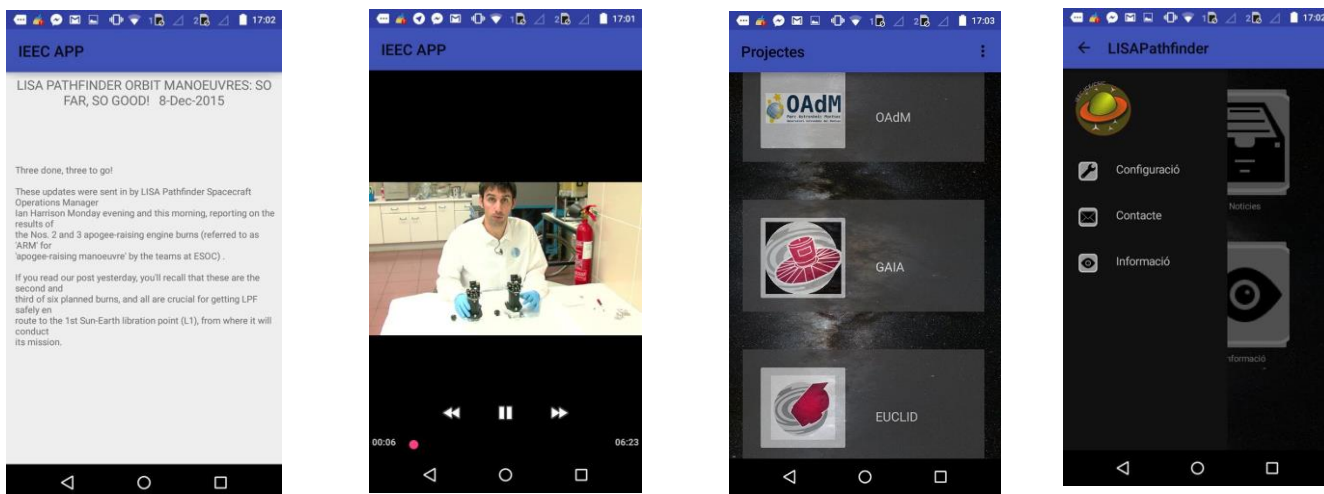


Figura 5: Screenshots corresponents al menu de projectes, secció de videos I finalment noticies.