



Plug-in «Manage Games» (KPAX2)

Jairo Sarabia Torres

Màster Oficial de Software Lliure (UOC)

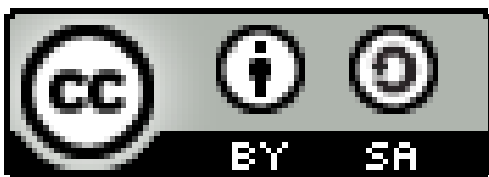
Administració de web i de comerç electrònic

Professor col·laborador: Francisco Javier Noguera Otero

Professor tutor: Daniel Riera i Terrén

1 de Juliol de 2018

LLICÈNCIA



Aquesta obra està subjecte a la llicència de Reconeixement-CompartirIgual 4.0 Internacional de Creative Commons. Si voleu veure una còpia d'aquesta llicència accediu a <http://creativecommons.org/licenses/by-sa/4.0/> o envieu una carta sol·licitant-la a Creative Commons, PO Box 1866, Mountain View, CA 94042, USA.

FITXA DEL TREBALL FINAL

Títol del treball:	<i>connector (plug-in) de gestió de jocs a la plataforma kPAX2</i>
Nom de l'autor:	<i>Jairo Sarabia Torres</i>
Nom del consultor/a:	<i>Francisco Javier Noguera Otero</i>
Nom del PRA:	<i>Daniel Riera i Terrén</i>
Data de lliurament (mm/aaaa):	<i>06/2018</i>
Titulació o programa:	<i>Màster Oficial de Software Lliure (UOC)</i>
Àrea del Treball Final:	<i>Administració de web i de comerç electrònic</i>
Idioma del treball:	<i>Català</i>
Paraules clau	<i>open social networks, e-learning, games</i>
Resum del Treball (màxim 250 paraules): <i>Amb la finalitat, context d'aplicació, metodologia, resultats i conclusions del treball</i>	
<p>El present treball té com a principal objectiu fer un estudi teòric i pràctic sobre la xarxa social oberta per a jocs d'aprenentatge, kPAX2. L'esmentada xarxa social té com a principal públic els desenvolupadors i els amants dels jocs seriosos.</p> <p>El propietari d'aquesta plataforma i client del present treball és la Facultat d'Estudis d'Informàtica, Multimèdia i Telecomunicacions (FUOC-EIMT) de la Universitat Oberta de Catalunya (UOC).</p> <p>La tecnologia que utilitza la plataforma és tecnologia web. D'una banda, està el front-end basat en el framework ELGG (per desenvolupament de xarxes socials), i desenvolupada en entorn LAMP (Linux, Apache Web, MySQL i PHP). D'altra banda, tenim el back-end desenvolupat (versió actual de kPAX2) amb Node.js i Mongo DB.</p> <p>El treball exposat en aquest document tracta de l'aprenentatge i desenvolupament d'un connector pel front-end de kPAX2, iniciat anteriorment per altre alumne del Màster de Software Lliure.</p> <p>La metodologia que inicialment es va fer servir és «àgil». Més endavant es va canviar a un desenvolupament en «cascada». També es van realitzar algunes reunions amb el client.</p>	

El desenvolupament va començar amb l'aprenentatge de les tecnologies en que es basa la plataforma kPAX. Va seguir amb l'aprenentatge del framework ELGG, i la plataforma kPAX. Va finalitzar amb l'aprenentatge i desenvolupament del connector «Manage Games».

Per concloure, ha estat una experiència enriquidora, on s'han après diverses tecnologies, i s'ha tingut l'oportunitat de contribuir a la plataforma kPAX. Finalment, m'hagués agradat disposar de més temps per finalitzar el desenvolupament del connector i corregir alguns bugs de la plataforma.

Índex

1. Introducció.....	iv
1.1 Context i justificació del Treball.....	iv
1.2 Objectius del Treball.....	v
1.3 Enfocament i mètode seguit.....	vi
1.4 Planificació del Treball.....	vi
1.5 Breu sumari de productes obtinguts.....	vii
1.6 Breu descripció dels altres capítols de la memòria.....	vii
2. Estudi de viabilitat.....	ix
2.1. Establiment de l'abast del sistema.....	ix
2.2. Estudi de la situació actual.....	xi
2.3. Definició dels requisits del sistema.....	xii
2.4. Estudi de les alternatives de solució.....	xiii
2.5. Valoració de les alternatives.....	xiii
2.6. Selecció de la solució.....	xiv
3. Anàlisi del sistema.....	xv
3.1. Definició del sistema.....	xv
3.2. Establiment de requisits.....	xvi
3.3. Definició d'interfícies d'usuari.....	xvii
3.4. Especificació del pla de proves.....	xxiii
4. Disseny.....	xxv
4.1. Arquitectura.....	xxv
4.2. Revisió de casos d'ús.....	xxvii
5. Desenvolupament.....	xxxí
5.1. Planificació de les activitats de desenvolupament i integració de sistema.....	xxxí
5.2. Desenvolupament.....	xxxii
5.3. Metodologia de desenvolupament.....	xxxiii
5.4. Desenvolupament del programari.....	xxxiii
5.4. Pla de proves revisat.....	xxxiv
5.5 . Principals problemes i solucions aportats.....	xxxiv
6. Implantació.....	xxxvi
7. Manteniment.....	xxxvii
8. Conclusions.....	xxxviii
9. Glossari.....	xl
10. Referència bibliogràfica.....	xli
11. Annexos.....	xliii
Annex I. Planificació temporal inicial.....	xliii
Annex II. Planificació temporal desenvolupament final.....	xliv
Annex III. Repositoris Git/GitHub del projecte.....	xliv
Annex IV. Repositoris O2 de kPAX.....	xlvi

1. Introducció

1.1 Context i justificació del Treball

La plataforma **kPAX** és una xarxa social oberta, basada en el framework **ELGG** pel desenvolupament de xarxes socials, i que té com a principal objectiu difondre l'ús i promoure el desenvolupament de «jocs seriosos» destinats a l'aprenentatge.

La idea i el projecte van néixer l'any 2012, i va ser creat per la Facultat d'Estudis d'Informàtica, Multimèdia i Telecomunicacions de la Universitat Oberta de Catalunya, en endavant **FUOC-EIMT**. Cal dir que la FUOC-EIMT també és el client del present treball.

L'any 2016, l'estudiant *Miquel Muntaner* va dur a terme una migració de la plataforma amb la finalitat de millorar la seva arquitectura i el rendiment de la plataforma. La nova i actual versió es va anomenar kPAX2, on es va separar la part del client (front-end) i la part del servidor (back-end).

Es va millorar l'arquitectura del back-end emprant tecnologies amb millor rendiment en entorns concurrents i de temps real, com és el cas de *Node.js* (motor back-end de *Javascript*) i *MongoDB* (base de dades No-SQL). Tot i que el front-end va continuar ser un entorn *LAMP* basat en ELGG, cal dir que es va actualitzar a la versió d'ELGG a la Elgg 2.2.2 i de *PHP* a *PHP7*. Aquesta actualització també va comportar una petita millora en el rendiment a la part del client (front-end).

També, d'ençà de la creació del projecte, s'han desenvolupat alguns connectors (o mòduls Elgg) pel front-end de la plataforma kPAX2. Alguns d'ells són: *kPAX_core*, mòdul core de l'aplicació; *theme_kPAX*, mòdul per canvia el tema (estils) d'ELGG; *loginrequired*, connector que manega l'autenticació (login) a la plataforma de kPAX; *apiadmin*, connector per gestionar les l'accés i permisos a les dades través d'API keys; *html5*, mòdul que permet fer servir les funcionalitats de *HTML5* i *CSS3*; *likeKpax*, connector que permet fer votacions dels jocs mitjançant likes/dislikes (m'agrada/ no m'agrada). Per últim, i amb especial menció ja que és l'objectiu principal del treball, «**manage_games**», mòdul destinat a la gestió dels jocs propis de desenvolupadors dins la plataforma kPAX.

Com ja s'ha dit, aquest últim mòdul per gestionar jocs propis de desenvolupadores a la plataforma és l'objectiu principal del present treball.

El connector es va iniciar el curs 2017/2018 per l'alumne *Francesc Pons Aróztegui*. En aquest treball es fa una nova iteració per introduir algunes millores al connector desenvolupat.

L'objectiu principal era treure la funcionalitat de la gestió de jocs fora del mòdul principal de l'aplicació (*kPAX_core*), ja que són funcionalitats molt concretes per la gestió de jocs propis dels desenvolupadors.

D'aquesta manera, es treu funcionalitat innecessària del core, refactoritzant-lo de manera que esdevengui en un mòdul més lleuger, més net i més fàcil de mantenir. D'altra banda, extreure aquestes funcionalitat en mòdul independent, també esdevé en unes funcionalitats més modulars, més mantenibles i escalables.

Per desenvolupar el connector s'ha fet també un model *CRUD* dels jocs, amb l'objectiu de manegar d'una manera senzilla la creació i modificar de les dades al servidor mitjançant una API REST.

Finalment, cal dir que el model *CRUD* ha quedat a «*grosso modo*» desenvolupat, a falta de polir alguns detalls i fer una bateria de test adequada abans de procedir amb la migració final.

1.2 Objectius del Treball

A continuació, s'exposen els objectius principals del present treball:

- Estudi i comprensió de les tecnologies que hi ha darrera del desenvolupament de la plataforma kPAX: PHP, javascript, MySQL, el framework ELGG, Node.js i Mongo DB
- Estudi i comprensió de l'estructura de la plataforma kPAX
- Desenvolupament del connector «*manage_games*» que implementa les funcionalitats de gestió de jocs dins la plataforma kPAX
- Integració del connector a la plataforma kPAX
- Tests funcionals del connector integrat a la plataforma kPAX

Més detalladament, els objectius del connector «*manage_games*» són els següents:

- Migració del *CRUD* de gestió de jocs desenvolupat al mòdul principal de l'aplicació (*kPAX_core*)
- Refactorització del cas d'ús de creació d'un joc integrat dins d'un mòdul independent
- Refactorització i integració del cas d'ús de llistar els jocs d'un usuari desenvolupador a la plataforma kPAX
- Desenvolupament i integració en un connector independent del cas d'ús de modificació d'un joc propi a l'aplicació
- Desenvolupament i integració en un connector independent del cas d'ús de visualització del detall d'un joc a l'aplicació
- Desenvolupament i integració mitjançant connector del cas d'ús d'eliminació d'un joc propi a l'aplicació kPAX
- Tests funcionals de totes les funcionalitats desenvolupades al connector

- Integració del connector desenvolupat a la plataforma kPAX

1.3 Enfocament i mètode seguit

Inicialment l'enfocament va consistir en una reunió amb el client on explicava l'estat actual del projecte i exposava les necessitats i requeriments principals a assolir.

A partir d'aquí, i amb el rol d'enginyer i desenvolupador, vaig gaudir de plena llibertat per escollir desenvolupar les funcionalitats i requeriments que més interès em suscités.

Vaig escollir continuar amb el desenvolupament del connector de gestió de jocs que havia quedat a mitges el curs anterior. A banda de que era l'opció que més m'agradava, crec que era un desenvolupament important i que era una necessitat acabar-lo el més aviat possible.

Inicialment, la metodologia seguida pel desenvolupament var ser una mena de «*desenvolupament en espiral*» (metodologia àgil), on havia de reunir-me periòdicament amb el client per tenir feedback del desenvolupat dut a terme durant un període de temps curt i acordat (sprint) i on prèviament s'havien acordat les tasques a desenvolupar en aquell sprint.

La idea era promoure l'agilitat del desenvolupament del projecte amb iteracions curtes i el feedback continu del client sobre les tasques desenvolupades i lliurades en cas d'acceptació del client.

1.4 Planificació del Treball

Els recursos requerits per realitzar el projecte han estat els següents:

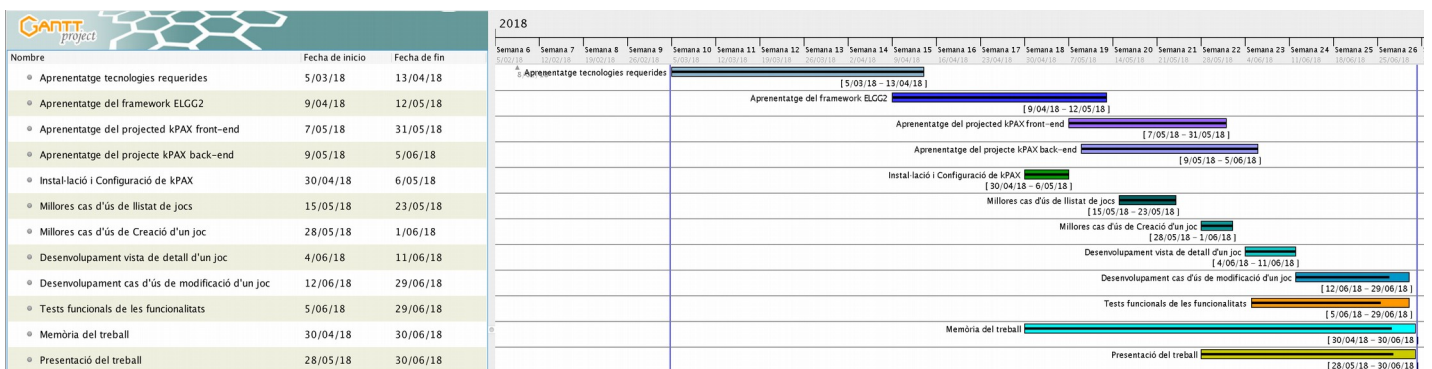
- un ordinador personal amb internet per poder desenvolupar i comunicar-me amb el client,
- la instal·lació i configuració de una màquina virtual amb la virtualitzadora *VirtualBox*, ja que els repositoris *GitHub* dels projectes (front-end i back-end) tenien l'entorn preparat amb Linux i el meu ordinador personal és un *Macbook (macOS)*,
- un *IDE (PHPStorm)* pel desenvolupament del connector de gestió de jocs,
- un navegador web, ja que la interfície d'usuari actual de kPAX és web,
- un paquet d'ofimàtica (*LibreOffice*) per la realització de la memòria i de la presentació del projecte,
- correu electrònic i Skype per consultar el client i reunir-me amb ell,

- *GanttProject*, software per fer diagrames de la planificació de les tasques del projecte,
- *QuickTime Player* per la gravació de la presentació del projecte,
- i *Trello* per la planificació i gestió de les tasques del projecte

Les principals tasques a realitzar són exposades a continuació:

- ◆ Aprenentatge de les tecnologies utilitzades: PHP, javascript, Node.js, MySQL, MongoDB i Git (GitHub)
- ◆ Aprenentatge del framework ELGG per desenvolupament de xarxes socials, en el qual es basada la plataforma kPAX (front-end)
- ◆ Aprenentatge dels projectes front-end i back-end de la plataforma kPAX
- ◆ Milliores en el desenvolupament del cas d'ús per llistar jocs propis
- ◆ Milliores en el desenvolupament del cas d'ús per crear jocs propis
- ◆ Desenvolupament del cas d'ús per modificar un joc propi
- ◆ Desenvolupament del cas d'ús per visualitzar el detall un joc propi
- ◆ Desenvolupament del cas d'ús per esborrar un joc propi
- ◆ Tests funcionals de les funcionalitats desenvolupades
- ◆ Integració de les funcionalitats desenvolupades a la plataforma kPAX
- ◆ Realització de la memòria i documentació del treball
- ◆ Realització de la presentació del treball

La planificació temporal del treball es pot veure al diagrama de Gantt següent:



1.5 Breu sumari de productes obtinguts

Tal i com s'ha esmentat prèviament, el principal producte construït és el connector d'ELGG «*manage_games*» per la gestió de jocs propis seriosos per a desenvolupadors de jocs dins la plataforma kPAX.

Resumint, el producte permet fer un CRUD sobre els jocs propis d'un desenvolupador. És a dir, permet crear, llegir o visualitzar els jocs propis, actualitzar-los i esborrar-los de la plataforma.

1.6 Breu descripció dels altres capítols de la memòria

A continuació es descriuen breument la resta de capítols del treball:

- 2. Estudi de viabilitat: és la fase del projecte on s'exposen els conjunt de necessitats que el client o usuari vol resoldre, i s'estudien les possibles solucions, tenint en compte les avantatges i desavantatges de cadascuna. Finalment s'escull la solució més adequada tenint en compte el context (aspectes econòmics, tècnics, legals, etc.) i les condicions d'acceptació del client
- 3. Anàlisi del sistema: aquí es descriu el sistema de la solució escollida, sense entrar en detalls tecnològics ni d'implementació, ja que són qüestions que pertanyen a la fase de disseny. Sobretot es descriuen els casos d'ús del sistema i els actors que intervenen
- 4. Disseny: es detalla la solució descrita a la fase anterior d'anàlisi. Es on definim tecnològicament els models i especificacions de la solució. També es defineixen els components i interfícies a implementar en la fase de desenvolupament, així com l'entorn de proves i implantació del sistema
- 5. Desenvolupament: es la fase d'implementació o construcció de la solució definida a les fases anteriors d'anàlisi i disseny. És una fase que es pot començar una vegada finalitzades les fases anteriors o es pot fer simultàniament amb les fases d'anàlisi i disseny. En aquesta fase també s'implementen les proves definides a la fase de disseny
- 6. Implantació: és la fase on es defineix i es posa el sistema en un entorn de producció. Es defineix la posada en marxa d'un sistema nou, o també la migració o substitució d'un sistema ja existent. En aquesta també es planifiquen i es realitzen proves amb usuaris reals del sistema
- 7. Manteniment: es on es planifiquen i es preveuen totes les tasques relacionades amb el manteniment del sistema, de manera que es garanteixi la seva operabilitat i estabilitat al llarg del temps
- 8. Conclusions: en aquest apartat s'analitzen els resultats obtinguts, s'exposen les conseqüències i deduccions lògiques a les que s'arriba amb el projecte
- 9. Glossari: és la llista amb les explicacions dels mots tècnics
- 10. Referències bibliogràfiques: la llista de la bibliografia, material i documentació consultat durant la realització del treball
- 11. Annexos: documentació i material important relacionat amb el projecte i que ajuda a entendre'l i conèixe'l més detalladament

2. Estudi de viabilitat

2.1. Establiment de l'abast del sistema

Partim ja d'un sistema existent, ja que la plataforma *kPAX* és un projecte que va néixer l'any 2012. Cal dir que és una xarxa social destinada a l'aprenentatge basat en jocs seriosos, i que el seu client i creador és la *FEIMT* de la *UOC*.

És una plataforma destinada a usuaris que desenvolupen jocs d'aprenentatge així com usuaris que volen aprendre jugant a jocs digitals.

És un projecte amb continua evolució ja que tant el client com els alumnes de la *UOC* desenvolupen noves funcionalitats o milloren les ja existents. Al ser un projecte amb llicència de software lliure, qualsevol persona interessada, podria contribuir al desenvolupament del projecte.

És un projecte separat en 2 repositoris, el front-end (client) i el back-end. L'objectiu d'aquest projecte és estudiar i acabar d'implementar un dels connector o mòdul de la part del front-end. Més concretament es tracta del mòdul de gestió de jocs anomenat «*manage_games*».

2.1.1. Requisits econòmics

Els costos econòmics per desenvolupar el projecte són molt baixos. D'una banda, el projecte té fa servir llicències de software lliure (*GPL v2*, *MIT*) que no tenen cap cost. D'altra banda, el desenvolupament és fa amb l'ajuda de la comunitat que no cobra, i amb la contribució dels alumnes de la *UOC* que fan projectes de final de grau i de màster basats en la plataforma *kPAX*.

També, cal dir que actualment els costos d'implementació i manteniment són mínims o nuls, ja que la plataforma no està allotjada en cap entorn de producció i els repositoris on s'allotja el codi del projecte també és gratuït, ja que *GitHub* no cobra pels repositoris públics ni pels repositoris privats amb comptes destinades a l'educació i la formació.

2.1.2. Requisits tècnics

Els requeriments tècnics per poder desenvolupar el projecte són els exposats més avall:

- ◆ Un ordinador per desenvolupar (preferiblement amb 8 Gb de memòria RAM)
- ◆ Internet per consultar documentació i comunicar-se amb el client
- ◆ Una distribució del sistema operatiu *Linux* o una virtualitzadora (per exemple, *VirtualBox*) amb una màquina virtual amb SO *Linux*

(preferiblement *Ubuntu 16*, ja que al repositori es facilita un «*shell script*» per instal·lar i configurar l'entorn en aquest sistema operatiu)

- ◆ *Git* i compte de *GitHub* per baixar-se els repositoris del projectes front-end i back-end de kPAX, i per poder actualitzar-los amb les funcionalitats desenvolupades
- ◆ Correu electrònic i chat (*Hangouts*, *Skype*) per consultar i comunicar-se amb el client
- ◆ Entorn *LAMP* per muntar el projecte front-end de la plataforma, amb SO *Linux*, *PHP7*, *MySQL* (en aquest cas *MariaDB 10*) i servidor web *Apache2*.
- ◆ Framework *Elgg*, versió *ELGG 2.2.2*, ja que és el framework en què està basat el projecte web client (front-end) de kPAX
- ◆ Tot i que no és estrictament necessari, es preferible muntar-se també l'entorn del servidor (back-end) kPAX. Aquí també és preferible muntar-se una VM Linux (*Ubuntu 16*) amb *Node.js* i *MongoDB*. També existeix al repositori Git del projecte un «*shell script*» per instal·lar i configurar l'entorn
- ◆ Un editor o *IDE* per desenvolupar. En aquest cas he fet servir *PHPStorm*, ja que és un dels *IDE* més complets per desenvolupar web que existeix a l'actualitat
- ◆ Un navegador web per desenvolupar (ja que kPAX té interfície d'usuari web) i per buscar i consultar informació a la web
- ◆ Paquet d'ofimàtica de software lliure (*LibreOffice*) per fer la documentació del projecte
- ◆ Programa per gravació de presentacions (en aquest cas *QuickTime Player*)
- ◆ Programari de software lliure per fer els diagrames de la fase d'anàlisi i de de disseny (*GanttProject* i *Visual Paradigm 15 CE*)

2.1.3. Requisits legals

El projecte està lligat a software lliure, i per tant, a llicències de software lliure.

Per una banda, tenim el front-end de kPAX que està basat en un entorn *LAMP* i en el framework *ELGG*, que té doble llicència *GPLv2 – MIT*. També, cal dir que tots el plugi-in del projecte front-end tenen llicència *GPLv2*, i més concretament el connector «*manage_games*» que és el connector objecte d'aquest treball.

També, cal dir que el propietari del projecte i del desenvolupament és la *FUOC* de la *UOC*, i que pot prendre lliurament i amb total dret la decisió de canviar de llicència.

2.1.4. Requisits operatius

Kpax és una xarxa social amb una interfície web, per tant els principals requisits d'operabilitat, tot i que és un projecte que encara no està implantat en un entorn de producció, són força elevats.

D'una banda, és important que la tecnologia utilitzada i el software desenvolupat sigui adient per un entorn amb possibilitat d'una alta concurrència i una elevada transmissió de dades, com és el cas d'una xarxa social.

D'altra banda, i com és normal en qualsevol sistema informàtic seriós i amb pretensions de publicar i implantar en un entorn de producció, el sistema ha de ser estable i escalable. Per tant, ha d'estar ben dissenyat i desenvolupat tenint cura dels tests de funcionals i unitaris per tal de garantir aquests requeriments.

Aquests requisits abans esmentats, s'aconsegueixen per una banda dissenyant una arquitectura client-servidor on tots dos són totalment independents, com és el cas de *kPAX2*, i d'altra banda amb un disseny del servidor que utilitza tecnologies preparades per suportar una elevada càrrega de dades i de connexions simultànies. Com és també el cas de la plataforma *kPAX 2*, que utilitza *Node.js* com a servidor web d'aplicació i *MongoDB* (base de dades no relacional) per la persistència de dades.

2.2. Estudi de la situació actual

Com ja hem comentat en capítols previs, la situació actual del sistema *kPAX2* és una situació de desenvolupament on no tenim encara entorn de producció i on encara queden moltes tasques per desenvolupar segons la visió final que té el client del producte.

L'any 2016, es va fer una migració del sistema per tal de millorar l'arquitectura, com ja hem exposat prèviament. Es va separar el client i el servidor en dos projectes amb repositoris independents. També es redissenar l'arquitectura del servidor, que estava implementada amb tecnologia web Java, i es va passar a tecnologia *Javascript* per servidor, *Node.js*; i a una base dades No SQL, *Mongo DB*. Finalment, també es va actualitzar la versió del framework *ELGG* a la part del client (front-end), concretament es va introduir la versió d'*Elgg 2.2.2*.

D'ençà de la migració, s'han desenvolupat diversos mòduls o connector pel projecte front-end de la plataforma. D'entre els quals tenim: ***kPAX_core***, mòdul core de l'aplicació; ***theme_kPAX***, mòdul per canvia el tema (estils) d'*ELGG*; ***loginrequired***, connector que manega l'autenticació (login) a la plataforma de

kPAX; **apiadmin**, connector per gestionar les l'accés i permisos a les dades través d'API keys; **html5**, mòdul que permet fer servir les funcionalitats de HTML5 i CSS3; **likeKpax**, connector que permet fer votacions dels jocs mitjançant likes/dislikes (m'agrada/ no m'agrada).

Finalment, i amb una menció especial, ja que és l'objecte principal d'aquest treball, tenim el connector **«manage_games»**, mòdul destinat a la gestió de jocs propis per desenvolupadors de jocs dins la plataforma kPAX.

L'estat actual d'aquest connector és inacabat i inestable. Només s'havien fet la funcionalitat bàsica de creació d'un joc i de la llista de jocs propis, i aquestes no estaven ben acabes ja que contenien **«bugs»** (errors).

2.3. Definició dels requisits del sistema

És un projecte ja iniciat amb un recorregut suficientment llarg i amb una arquitectura ben definida, com ja hem esmentat anteriorment, per tant es pot dir amb força seguretat que és un projecte viable.

Tal i com hem descrit en apartats anteriors, el projecte client està basat en el framework *Elgg*. Per tant, ens hem d'adaptar als requisits i a la manera de desenvolupar d'aquest framework per tal de poder implantar nous mòduls al sistema.

A continuació, s'explica amb més detall els requisits bàsics per tal d'implementar un mòdul nou al sistema.

Primerament, es necessiten tres components bàsics:

- ◆ el fitxer **«manifest.xml»**: és el fitxer (en format XML) que conté la informació descriptiva i metadades del connector (o connector). Aquí es posen dades com la versió d'*Elgg* requerida, les dependències (p. ex. altres connector), la llicència, etc.
- ◆ el fitxer **«start.php»** (script PHP): és el punt d'entrada del mòdul o connector. Aquí es configuren les accions i components principals que tindrà el mòdul, així com les llibreries que utilitzarà. Dos elements imprescindibles són:
 - l'inicialitzador del connector, on es defineixen els directoris de llibreries, interfícies d'usuari (HTML, CSS, imatges, etc), i les accions o controladors del mòdul
 - gestor de pàgines dins del mòdul, que depenent de la ruta (URL) definida es cridarà un script PHP determinat on s'executarà una acció del mòdul i es carregarà una pàgina concreta associada
- ◆ **«estructura de directoris»** del mòdul: consisteix en l'esquelet principal o estructura general del mòdul. A l'arrel es troba el punt d'entrada principal, és a dir, l'script **«start.php»** i la resta de subdirectoris que componen el connector:

- **actions:** directori troben les accions que es criden quan s'envia un formulari des d'*Elgg*. Són accions *CRUD* que no mostren res a l'usuari. Gestionen les entitats del model que s'envia i es reben. Les accions, com ja hem comentat, estan definides a l'script d'arranc del connector (fitxer *start.php*)
- **pages (controlador):** són controladores en un model arquitectònic *MVC (Model-Vista-Controlador)*. Aquí es defineixen les pàgines web que tindrà el mòdul i amb les que l'usuari podrà interaccionar. Cada «*page*» tindrà una ruta (URL) única associada, a la qual es cridarà per tal d'executar la pàgina i finalment mostrar la vista corresponent a l'usuari
- **lib (lliberies):** aquí s'han d'introduir les lliberies i s'han de definir les entitats del model (classes, interfícies, etc.)
- **views (vistes):** és el directori on es defineixen les vistes per cada pàgina (*page*) definida a la carpeta de «*pages*». Són les vistes web (HTML+CSS) que es visualitzaran a l'executar-se una controladora determinada del connector. Aquest directori està format per altres:
 - *forms*: aquí es defineixen els formularis web *HTML* del connector
 - *js*: aquí es posen les funcionalitats desenvolupades amb certa lògica de les vistes en llenguatge *Javascript*. És important però, tenir en compte que han de ser scripts amb format *PHP*
 - *css*: aquí van tots els fitxer *CSS* amb els estils per les vistes del mòdul
- **graphics:** directori posar les imatges per les pàgines del connector
- **languages:** aquí es defineixen els fitxers d'internacionalització del connector. Ha d'haver un fitxer per cada idioma suportat amb les traduccions corresponents

2.4. Estudi de les alternatives de solució

Com que estem parlant d'un mòdul ja iniciat en cursos anteriors, les alternatives donades pel client estaven relacionades evidentment amb la plataforma *kPAX*, però no amb aquest mòdul concret.

Altres alternatives van ser:

- Millorar l'estabilitat de tota la plataforma i el conjunt de mòduls desenvolupats
- Millorar la part de seguretat de la plataforma, sobretot l'enviament de dades sensibles i que han d'estar protegides o són propietat de tercers

Cal dir, que una altra alternativa no exposada pel client, podria ser la refactorització i migració de la part del client (front-end) de la plataforma a una tecnologia i arquitectura més actualitzada i flexible (*Minds*, *HumHub*, *BuddyPress*)

2.5. Valoració de les alternatives

La primera alternativa és força complexa, perquè requereix una comprensió i un nivell de coneixement de la plataforma molt elevat per poder-la dur a terme amb cert èxit. L'estabilitat d'un sistema és un tema important i a la vegada força complicat, per tant no era l'alternativa més idònia donada la situació en la que hem trobava inicialment, ja que no sabia res sobre la plataforma i molt poc sobre les tecnologies que utilitza. I tampoc disposava del temps suficient per aprendre tot el codi desenvolupat al projecte des de l'any 2012.

Les raons principals per les quals també vaig descartar la segona opció són les mateixes. La seguretat d'un sistema és un tema delicat i molt complexe, i si a més a més tenim en compte que no es tenien coneixements previs sobre el tema ni sobre el projecte, amb el temps del que disposava era una feina pràcticament impossible de dur a terme amb èxit.

I finalment, la possibilitat de migrar el projecte front-end a una plataforma i tecnologia més moderna i flexible és una alternativa força interessant. Però, un vegada més, per les mateixes raons que les donades per les altres alternatives, era una opció que tampoc es podia dur a terme amb èxit tenint en compte el temps disposat.

2.6. Selecció de la solució

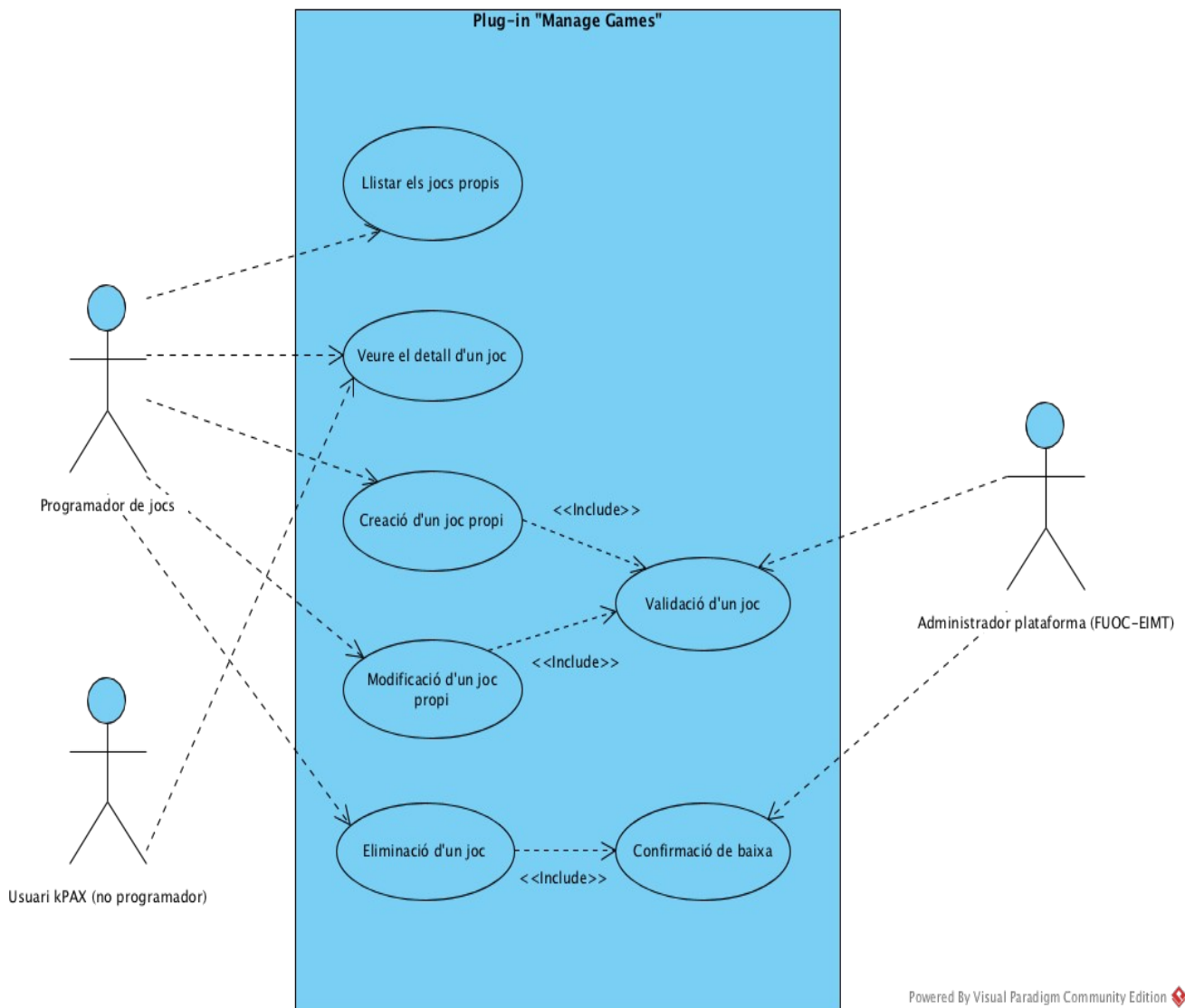
Per concloure, crec que la selecció de la alternativa de continuar amb el desenvolupament del connector «*manage_games*» era l'opció més encertada, donat els coneixements previs del projecte i de les tecnologies que utilitza, i donat el curt temps del que disposava per dur a terme el treball.

3. Anàlisi del sistema

3.1. Definició del sistema

La definició del connector estudiat i desenvolupat durant el present treball, és a dir, del connector «*manage_games*», ja estava feta prèviament. Ja que, com s'ha explicat anteriorment, és un connector ja iniciat per altre alumne el curs passat.

Tot i així, a continuació s'exposen els casos d'ús definits pel connector:



Actors del sistema (connector «*manage_games*»):

- ◆ **Programador de jocs:** és l'usuari o actor del sistema que desenvolupa jocs, i per tant, és l'actor principal en aquest connector. És l'actor que crea, modifica, i elimina jocs propis al sistema

- ◆ **Administrador del sistema (FUOC-EIMT):** és l'administrador de la plataforma kPAX, i per tant, té permissos per fer tot tipus d'acció al sistema. És l'actor encarregat de validar un joc nou o les modificacions sol·licitades sobre un joc. També és l'encarregat de confirmar la baixa d'un joc a la plataforma
- ◆ **Usuari kPAX ordinari (no desenvolupador de jocs):** usuari de kPAX, que no és desenvolupador de jocs seriosos, però que pot llistar i veure informació pública sobre els jocs introduïts al sistema o fer comentaris

Els casos d'ús que implementarà el connector i els seus estàndards són els següents:

- ◆ el connector ha de permetre crear un joc propi a un desenvolupador de jocs de la plataforma
- ◆ el connector ha de permetre modificar les dades d'un joc propi al seu creador o desenvolupador
- ◆ el connector ha de permetre llistar tots els jocs propis a un usuari desenvolupador de jocs
- ◆ el connector ha de permetre sol·licitar la baixa d'un joc propi al seu creador
- ◆ el connector ha de permetre visualitzar les dades d'un joc al seu creador i a altres usuaris de la plataforma
- ◆ el connector ha de seguir l'estructura d'un mòdul d'*Elgg*
- ◆ el codi i el connector en general, han de disposar de documentació necessària pel seu ús i comprensió
- ◆ les pàgines i vistes han de seguir, sempre que sigui possible, els estàndards web de la W3C
- ◆ les baixes i modificacions sobre un joc hauran de ser confirmades per motius de seguretat i per evitar accidents
- ◆ el mòdul ha d'estar adaptat i donar suport a diversos idiomes (internacionalització)
- ◆ les vistes del connector han de ser *responsiva*, han d'adaptar-se adequadament a les característiques i pantalles de diferents dispositius
- ◆ un joc podrà ser categoritzat dins d'un grup específic, i amb unes propietats determinades
- ◆ l'usuari creador de jocs ha de poder donar un format ric en continguts als jocs creats. Per tant, és important que l'usuari disposi d'un «*rich editor*» (editor de contingut enriquit) per donar format al contingut d'una forma amena
- ◆ el connector es comunicarà mitjançant l'*API REST* amb el servidor del sistema per enviar i rebre les dades dels jocs creats
- ◆ s'han de poder crear versions diferents d'un mateix joc a la plataforma

3.2. Establiment de requisits

Els requeriments necessaris per dur a terme el desenvolupament del connector ja estaven definits abans comences el present treball. D'una banda, perquè

molts dels requisits són globals i compartits amb la resta de la plataforma, i de l'altra, perquè el connector desenvolupat ja ha estat definit i iniciat prèviament.

Tot i així, crec que és important esmentar els requisits essencials per desenvolupar aquest connector:

- requisits funcionals: el connector desenvolupat a de complir amb els requisits, pre-condicions i post-condicions dels casos d'ús definits prèviament
- requisits de seguretat: el connector només permetrà fer accions a un desenvolupador de jocs i a l'administrador de la plataforma. Només un propietari o l'administrador podrà modificar o donar de baixa un joc. Qualsevol acció crítica haurà de ser confirmada abans de ser dut a terme pel sistema
- requisits de rendiment: tot i que el rendiment del sistema ja bé donat en gran part per l'arquitectura actual de la plataforma, el connector desenvolupat a de garantir el ús mínim possible de recursos del sistema per dur a terme les accions sol·licitades
- requisits d'operabilitat: el connector ha de definir detalladament els requisits i dependències per poder estar operatiu. El connector ha de garantir uns barems mínims d'estabilitat per garantir la seva disponibilitat i la del sistema en general
- requisits d'implantació: el codi del connector desenvolupat s'integrarà fent servir els repositoris que hi ha disposició del desenvolupador al servidor git de *GitHub*. Es sol·licitarà la integració de tot codi desenvolupat als administradors del projecte mitjançant «*pull request*».

3.3. Definició d'interfícies d'usuari

La interfície d'usuari definida a la plataforma és web, per tant l'usuari podrà accedir a la plataforma i fer ús d'ella només per mitjà d'un dispositiu amb internet i amb navegador web.

Els usuaris que faran ús del connector són normalment usuaris amb un nivell mig-alt d'experiència en entorns web (desenvolupadors de jocs web, i els administradors del sistema, amb una ampla experiència en diversos camps de sistemes de la informació).

Tot i que el servidor sempre ha de validar les accions sol·licitades i el contingut de les dades, la interfície d'usuari també ha de garantir un mínim de validacions per tal de donar feedback a l'usuari i fer-li més amena l'ús de la plataforma.

Per tant, uns principis bàsics que el connector ha de garantir s'exposen a continuació:

- l'usuari enviarà les dades al sistema per crear o modificar un joc per mitjà de formularis web que han de donar feedback constant a l'usuari i garantir una interacció fàcil amb la plataforma

- els estils del connector mantindrà els estàndards ja implantats a la plataforma i altres connector desenvolupats prèviament. Els missatges i les notificacions mostrades a l'usuari tindran el mateix format que es fa servir a la resta de la plataforma
- s'ha de garantir que la pàgina d'entrada al connector contingui tota la informació i les instruccions necessàries per fer ús adequat del connector
- les accions sobre els jocs es faran per mitjà de botons a les pàgines del connector, i hauran de ser confirmades mitjançant una finestra de confirmació (pop-up)
- l'usuari tindrà a la seva disposició el llistat de jocs propis a la pàgina principal del connector. Des d'aquest llistat podrà fer totes les accions permeses sobre un joc: consultar les dades del joc (fitxa del joc), editar les dades d'un joc propi, i sol·licitar la baixa d'un joc propi. Totes les accions enllaçaran i portaran a una pàgina independent del connector
- les pàgines de creació i edició seran molt semblants. De fet, mostraran el mateix formulari de creació d'un joc, però en cas d'edició, el formulari serà emplenat amb les dades actuals del joc a editar
- la pàgina de fitxa de detall d'un joc ha de mostrar una pagina web amb les dades del joc i ha de ser només amb permisos de lectura
- la pàgina per sol·licitar la baixa d'un joc ha de mostrar una pagina web com la de consulta de la fitxa però ha d'afegir un botó per poder sol·licitar la baixa del joc, i ha de mostrar una finestra d'avís per amb confirmació de la baixa del joc
- Les imatges que són portada d'un joc han de ser també enllaços a la pàgina web oficial del joc. Aquí s'han de tenir en compte aspectes legals relacionats amb l'administrador de la plataforma *kPAX* i el creador o desenvolupador del joc

A continuació es mostren algunes imatges sobre la interfície d'usuari al mòdul de gestió de jocs (*manage games*).

Pàgina principal del connector, on es mostra informació i instruccions del connector:

The screenshot shows a user profile page for 'jairo' on the Elgg platform. The main heading is 'My New kPAX'. Below the heading is a navigation menu with 'Activity', 'Games', 'Develop', 'Blogs', 'Bookmarks', and 'More'. The 'Games' section is active, showing 'manage_games' with 'My games' and 'Add game' buttons. The page content includes 'Instructions for Games Developers' and 'Games development' sections. The footer mentions 'Powered by Elgg' and 'EIMT.UOC.EDU'.

Account »

My New kPAX

Activity Games Develop Blogs Bookmarks ▾ More

manage_games

My games Add game

Instructions for Games Developers

kPAX philosophy is based in free software and open source. All the code is available under the [GNU General Public License, version 2 \(GPL-2.0\)](#). For those developers interested in participating in the kPAX project, there are two ways to do it: On the one hand, it is possible to **add functions/improvements to the platform**. On the other hand, new external **games, apps, etc.** to offer to users (players) are welcome. Let us see how to do it.

Games development

If you want to develop a game (do not forget kPAX is only for educative ones) you only need to insert in your code the necessary calls to kPAX engine which validate users, the game itself, and send/receive information to/from the database.

The most simple relationship between a game and kPAX might imply user and game validation, and at the end, send the score to be incorporated to the user's game information to update the database.

To do this, some security has to be added to communication between games and the platform. Security issues in kPAX are dealt with by means of: the validation of games using a public/private key pair, and users autorisation, using [OAuth](#). This means the developer has to create the game's public/private key pair and include the public one in the game's form. The private is used to sign every message sent to kPAX, which can certify it corresponds to the corresponding game.

Powered by Elgg

EIMT.UOC.EDU

Pàgina del llistat de jocs d'un desenvolupador de jocs seriosos a la plataforma de *kPAX*:

My New kPAX

Activity Games Develop Blogs Bookmarks ▾ More

My developed games

ID	Name	Description	Cat	Platform	Actions
65	INVENTAT	Nom duplicat	1	2016-12-18T18:53:01.731Z	View Edit Delete
66	Prova actions	1	1	2016-12-21T06:03:16.918Z	View Edit Delete
72	La prova	test	5	2016-12-28T11:58:42.895Z	View Edit Delete
49	The legend of Zelda	Great game	1	2017-01-09T15:19:29.685Z	View Edit Delete
87	Prova amb camp ulikes	Prova amb camp ulikes	1	2017-01-13T18:03:28.105Z	View Edit Delete
88	Resident Evil	Resident Evil	1	2017-01-15T19:02:39.897Z	View Edit Delete
54	INVENTAT	test	1	2017-10-14T14:18:40.663Z	View Edit Delete
50	mijuego	hola carcola	1	2017-10-19T14:37:20.362Z	View Edit Delete
59	joc administrador fponzar 2	test 2	1	2017-10-20T11:26:16.391Z	View Edit Delete
67	Prova actions	1	5	2016-12-21T06:06:32.955Z	View Edit Delete
68	Prova actions 2	Segona prova on redirigeix	6	2017-10-21T16:35:44.738Z	View Edit Delete
70	pons3	A veure si surt	1	2017-10-22T17:23:03.705Z	View Edit Delete
75	onvaejoc?	A veure quina base de dades alimenta, mongo o mySQL	1	2017-11-01T15:00:48.127Z	View Edit Delete
76	testbons0210	Prova ara	2	2017-11-	View Edit Delete



Search


Pàgina de visualització de la fitxa d'un joc propi:

My New kPAX

Activity Games Develop Blogs Bookmarks More

Games > Develop > Games

Jairo test game

By  jairo 17705 days ago
e-learning, social-game

Logged in users Edit

My test game. That is a new social game and related to e-learning.

Category:
Date of creation: 2017-10-13T18:01:56.512Z
Game id: 53

Leave a comment [Edit HTML](#)

Search

Powered by Elgg

[EIMT.UOC.EDU](#)

Pàgina de creació o edició d'un joc propi:

My New kPAX

ActivityGamesDevelopBlogsBookmarks▼ More

Add game

Game name (*)

Description [Edit HTML](#)

That's a ~~serious~~ game.

body p

Link to thumbnail in game site web

License applicable to the use of the game (*) [Edit HTML](#)

GPLv2

body p

Game category/ies

Creation date

Available platforms

- web
- android
- iOS
- Nintendo DS
- PSP
- Nintendo Wii
- XBox

Skills (*)

Related tags

Security

Search

3.4. Especificació del pla de proves

Les proves que cal dur a terme per garantir l'estabilitat i el correcte comportament del sistema són diversos, i m'he trobat que no existeix actualment cap mena de test automatitzat a la aplicació. Doncs, dit això, actualment cal fer les proves funcionals de la plataforma i dels connector manualment.

Tot i això, crec que val la pena descriure els tests que hauria de tenir el projecte per tal de garantir uns mínims barems en quant a funcionalitat, operabilitat, estabilitat, rendiment i seguretat.

3.4.1. Proves funcionals

En aquestes proves s'han comprovar la correcció i l'acceptació per part del client dels requisits de definits als apartats anteriors. Per tant, cal garantir amb aquestes proves que les funcions i casos d'ús desenvolupats funcionen correctament i compleixen realment amb les especificacions definides prèviament pel client.

Aquestes proves actualment s'han de dur a terme manualment, però estaria bé automatitzar-les amb eines i programari destinat a tal finalitat. Algunes eines força conegudes per fer aquestes proves de caixa negra són *Selenium HQ Web*, *Behat* o *Codeception*.

Més concretament, el connector de gestió de jocs «*manage games*» ha de fer les crides a la *API REST* del servidor *kPAX* correctament. És a dir, s'ha de garantir la correctesa del model *CRUD* definit pels jocs: donar d'alta un joc propi, consultar un joc o una llista de jocs, editar un joc propi i donar de baixa un joc propi.

També cal garantir amb aquestes proves la robustesa i la precisió de les dades transmeses al model *CRUD* de gestió de jocs seriosos. És a dir, cal garantir que les dades introduïdes per l'usuari són les dades que es guarden, i que les dades guardades són les dades que es mostren a l'usuari sempre que tingui els permisos per veure-les.

3.4.2. Proves d'usabilitat

El client ha d'acceptar les funcionalitats desenvolupades i lliurades, i per dur a terme tal acceptació, primer cal fer les proves d'usabilitat pertinents.

Actualment, són proves que ha de fer a mà el client, però existeixen eines que segueixen una metodologia *BDD* (*Behaviour Driven Development*) i que es podrien utilitzar per implementar proves automatitzades d'acceptació i usabilitat. Algunes d'aquestes eines són *Behat* o *Codeception*.

Com ja hem descrit en capítols anteriors, la usabilitat és un tema important i és necessari fer un bon disseny de les interfícies d'usuari del connector per garantir uns mínims d'usabilitat.

3.4.3. Proves de rendiment

Assolir els requisits de rendiment en una xarxa social és important i del tot necessari. Per tant, s'han de fer proves d'estrès (*stress test*) per garantir un rendiment adequat del connector desenvolupat i de la plataforma global.

Tampoc existeixen proves automatitzades per garantir el rendiment de la plataforma, però val la pena esmentar alguna eina per poder fer les proves necessàries abans d'implantar la plataforma en un entorn de producció.

Algunes d'aquestes eines per fer proves de rendiment i càrrega són: *JMeter*, *Applause*, *Tsung*, *The Grinder* o *Tank*.

3.4.4. Proves de seguretat

Les proves per tal de garantir la seguretat d'un sistema sempre són delicades i més o menys complexes. El sistema només ha de permetre fer les accions que estan definides pel client i només als usuaris autenticats i autoritzats a tal efecte.

En aquest cas, només un desenvolupador de jocs i l'administrador del sistema pot crear, modificar i eliminar jocs de la plataforma. A més, un desenvolupador només pot executar aquestes accions contra els seus propis jocs.

Aquestes proves queden fora de l'abast d'aquest treball, però cal dir que existeixen en l'actualitat tècniques i eines per minimitzar els possibles forats de seguretat d'una aplicació web, tant a la fase de desenvolupament com a la fase d'implantació i de manteniment.

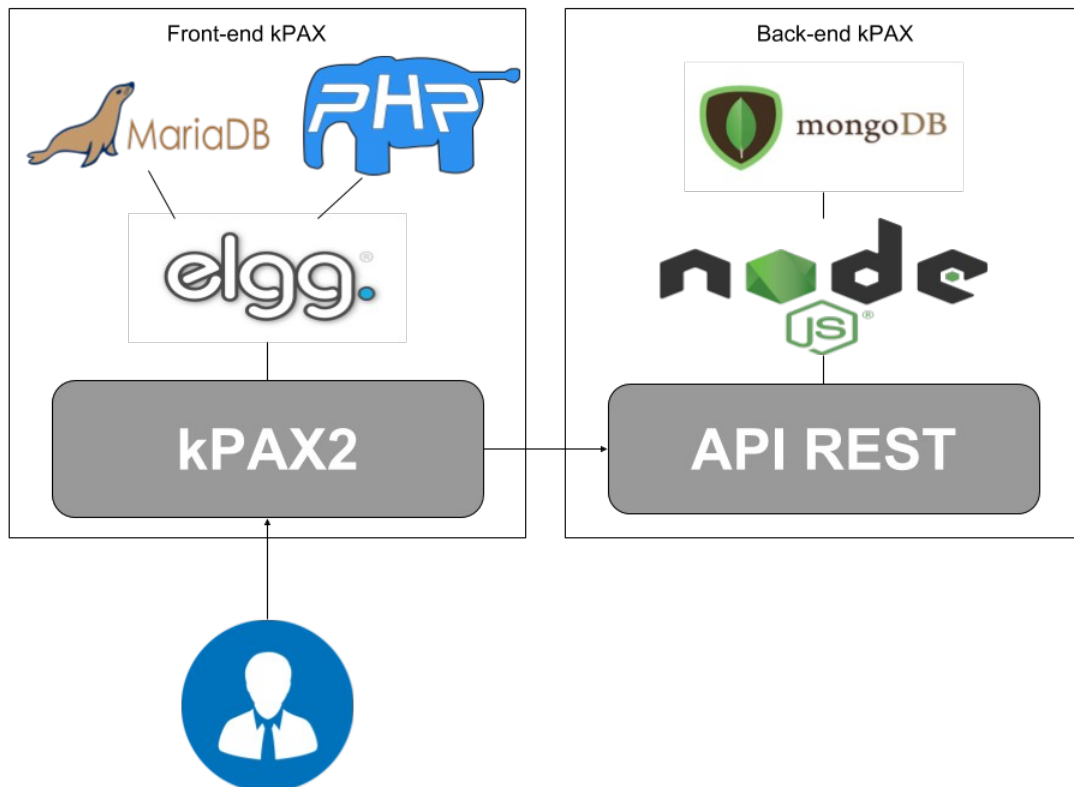
També, és necessari monitorar els accessos i l'ús que es dona de la plataforma, per tal d'evitar abusos o usos malintencionats, i per prendre les mesures adequades en cas que succeeixin aquests usos indesitjats.

4. Disseny

4.1. Arquitectura

4.1.1. Definició de nivells d'arquitectura

A continuació, mostrem un diagrama de l'arquitectura de la plataforma *kPAX* a grans trets.

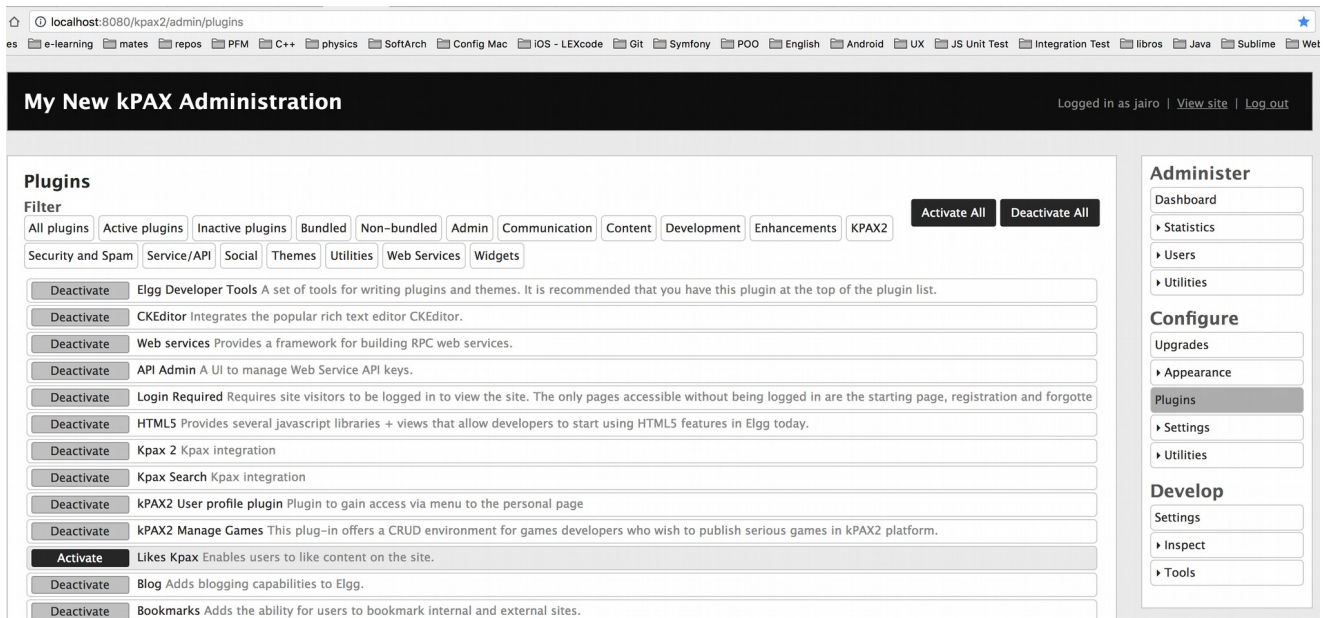


Com ja s'ha esmentat, la plataforma *kPAX* està composta per dos grans projectes.

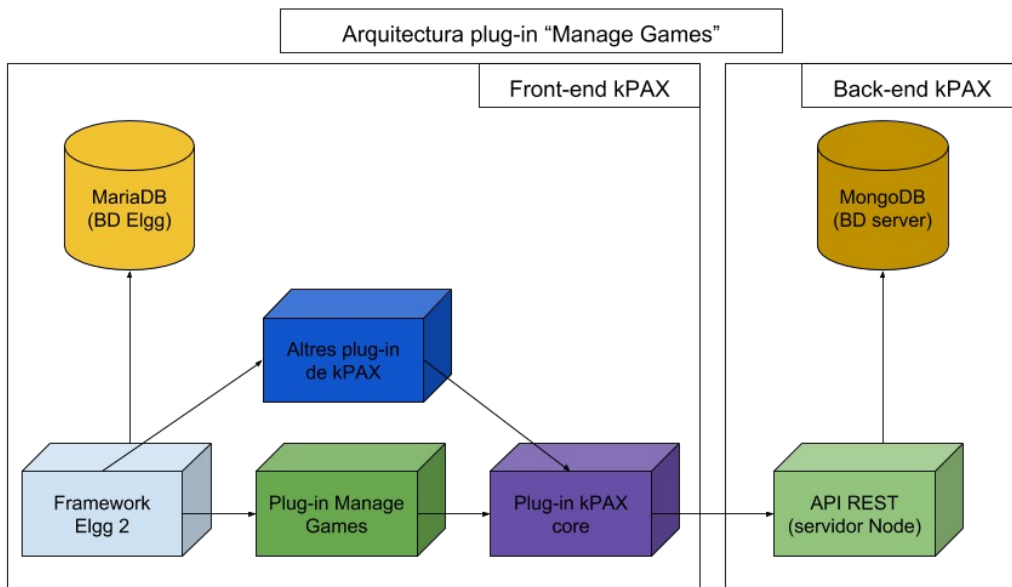
D'una banda, tenim la part client (front-end) desenvolupada en entorn *LAMP* (*Linux, Apache, MySQL/MariaDB, i PHP*) i basada en el marc per desenvolupament de xarxes social *Elgg2*. I d'altra banda, tenim la part del servidor (o back-end) amb una arquitectura dissenyada i basada en un model *API REST* amb tecnologies *Node.js* (servidor de Javascript) i *MongoDB* (base de dades NoSQL).

Cal dir, que el front-end és força escalable i ampliable gràcies al desenvolupament de mòduls nous (connector) basat en el sistema de connectors d'*Elgg*.

A continuació es mostren alguns del mòduls desenvolupats i activats en l'ordre adequat pel seu ús:



A continuació, també mostrem un diagrama a grans trets de l'arquitectura del connector «*manage_games*» per la gestió de jocs propis a la plataforma front-end:



L'usuari es comunica amb el connector per mitjà de la plataforma *Elgg*. D'altra banda, el connector es comunica amb el servidor per mitjà del connector *core* d'*ELGG* per enviar i rebre dades mitjançant l'*API REST* que existeix al servidor *Node.js* de la plataforma *kPAX*. Altres connector es comuniquen amb l'usuari i el servidor *kPAX* també de manera molt semblant.

4.1.2. Especificació d'estàndards, normes de disseny i construcció

Els estàndards i les normes de disseny per a desenvolupar el connector de gestió de jocs seriosos, en realitat ja ha estat definit i exposat en apartats anteriors.

Cal insistir en que s'han de seguir les normes de disseny i construcció de connector del marc *ELGG*, que és el marc de desenvolupament de xarxes social en que es basa la plataforma client del sistema *kPAX*. La seva estructura i els fitxers més rellevants ja s'han sigut explicats en apartats anteriors del present document.

En aquest cas no s'han de desenvolupar noves crides a l'*API REST* del servidor de la plataforma *kPAX*. Ja que el *CRUD* per jocs i els «*endpoints*» (punts d'entrada) corresponents de l'*API* ja van ser definits i desenvolupats en treballs previs a aquest.

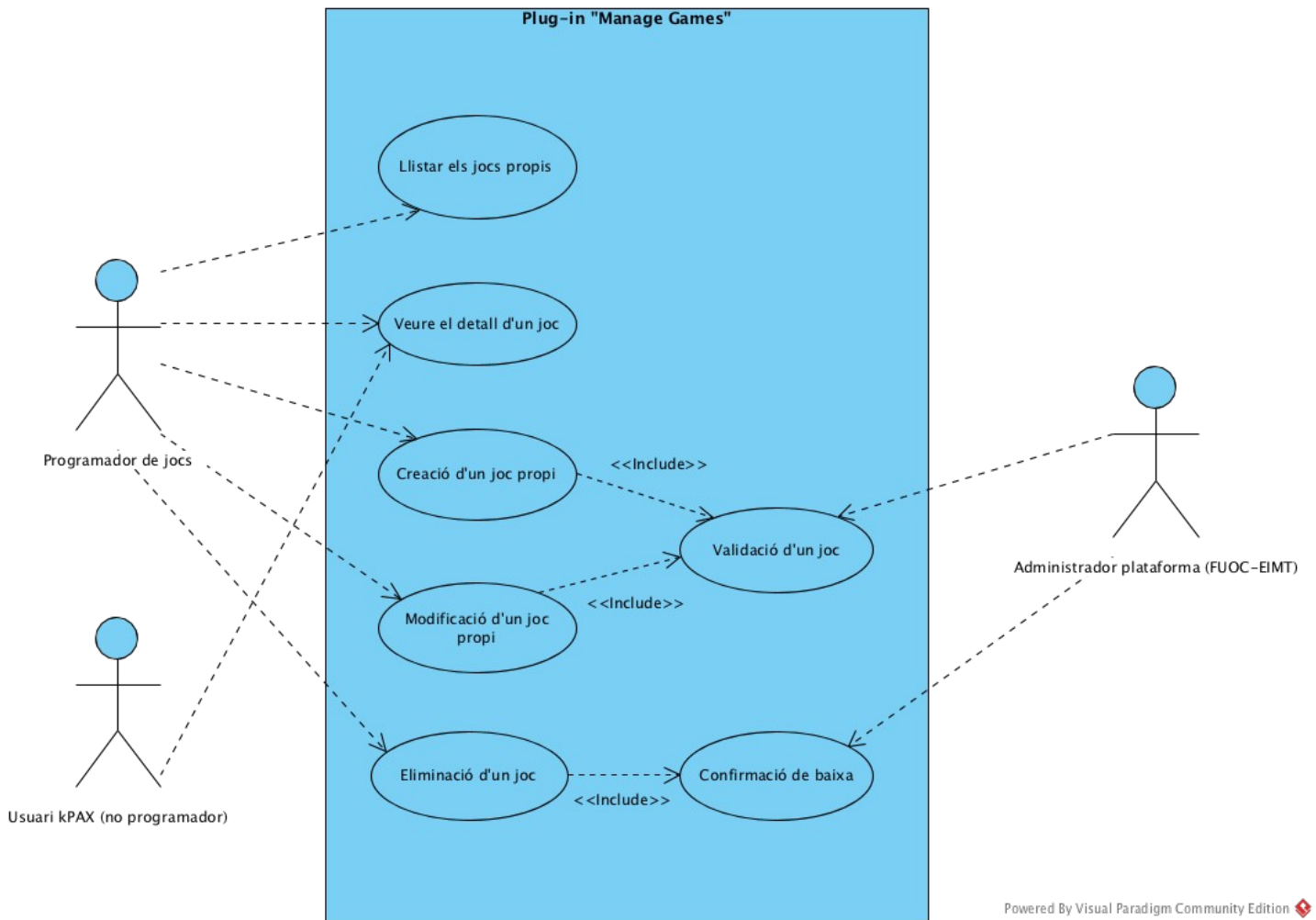
4.1.3. Identificació de subsistemes

Els subsistemes ja han estat definits i exposats varies vegades en capítols de fases anteriors del projecte.

Però, val la pena insistir en que estem parlant d'una arquitectura client-servidor, i per tant, existeixen dos subsistemes principals a la plataforma *kPAX*: el front-end *kPAX*, basat en *Elgg* i desenvolupat en un entorn *LAMP*; i el back-end *kPAX*, basat en una arquitectura *API REST* i un model *CRUD*, i desenvolupat amb tecnologia *Node.js* i *MongoDB*.

4.2. Revisió de casos d'ús

Els casos d'ús del sistema i del connector desenvolupat ja han estat definits en treballs previs al present document. Per tant, l'únic que cal fer es una revisió dels principals casos d'ús que ha d'implementar el connector «*manage_games*» de gestió de jocs seriosos.



A continuació es descriuen més detalladament els principals casos d'ús que implementa el connector desenvolupat «*manage_games*».

Cas d'ús 1: Llistar jocs propis

Resum	Un usuari autènticat a la plataforma, i amb rols de desenvolupador (o administrador) ha de poder accedir al seu llistat de jocs propis
Actors	Desenvolupador de jocs, administrador del sistema
Casos d'ús relacionats	
Pre-condicions	L'usuari ha d'estar autènticat i tenir el rol de desenvolupador de jocs o d'administrador de la plataforma
Post-condicions	Mostrar a l'usuari desenvolupador el seu llistat de jocs propis

Cas d'ús 2: Sol·licitar la fitxa d'un joc

Resum	Un usuari autènticat a la plataforma, i amb rols de desenvolupador (o administrador) ha de poder accedir a la d'un joc propi
Actors	Desenvolupador de jocs (o administrador del sistema)
Casos d'ús relacionats	
Pre-condicions	L'usuari ha d'estar autènticat i tenir el rol de desenvolupador de jocs (o d'administrador de la plataforma). També ha de tenir algun joc propi creat prèviament.
Post-condicions	Mostrar a l'usuari la pàgina de la fitxa del seu joc creat prèviament.

Cas d'ús 3: Crear un nou joc propi

Resum	Un usuari autènticat a la plataforma, i amb rols de desenvolupador (o administrador) ha de poder crear la fitxa d'un joc propi desenvolupat
Actors	Desenvolupador de jocs (o administrador del sistema)
Casos d'ús relacionats	L'administrador del sistema ha de validar que el joc creat és un joc seriós vàlid per la plataforma.
Pre-condicions	L'usuari ha d'estar autènticat a la plataforma i amb rol de desenvolupador de jocs (o d'administrador de la plataforma) ha d'haver desenvolupat o està desenvolupant un joc seriós.
Post-condicions	Mostrar a l'usuari la pàgina de confirmació amb la fitxa del joc creat. Posteriorment s'ha de notificar a l'usuari de la validació del joc.

Cas d'ús 4: Modificar un joc propi

Resum	Un usuari autènticat a la plataforma, i amb rols de desenvolupador (o administrador) ha de poder editar la fitxa d'un joc propi creat anteriorment.
Actors	Desenvolupador de jocs (o administrador del sistema)
Casos d'ús relacionats	L'administrador del sistema ha de validar que les noves dades del joc continuen sent vàlides per la plataforma.
Pre-condicions	L'usuari ha d'estar autènticat a la plataforma i

	amb rol de desenvolupador de jocs (o d'administrador de la plataforma) ha d'haver creat un joc seriós prèviament.
Post-condicions	Mostrar a l'usuari la pàgina de confirmació amb la fitxa de les dades del joc modificat. Posteriorment, s'ha de notificar a l'usuari de la validació de les noves dades del joc.

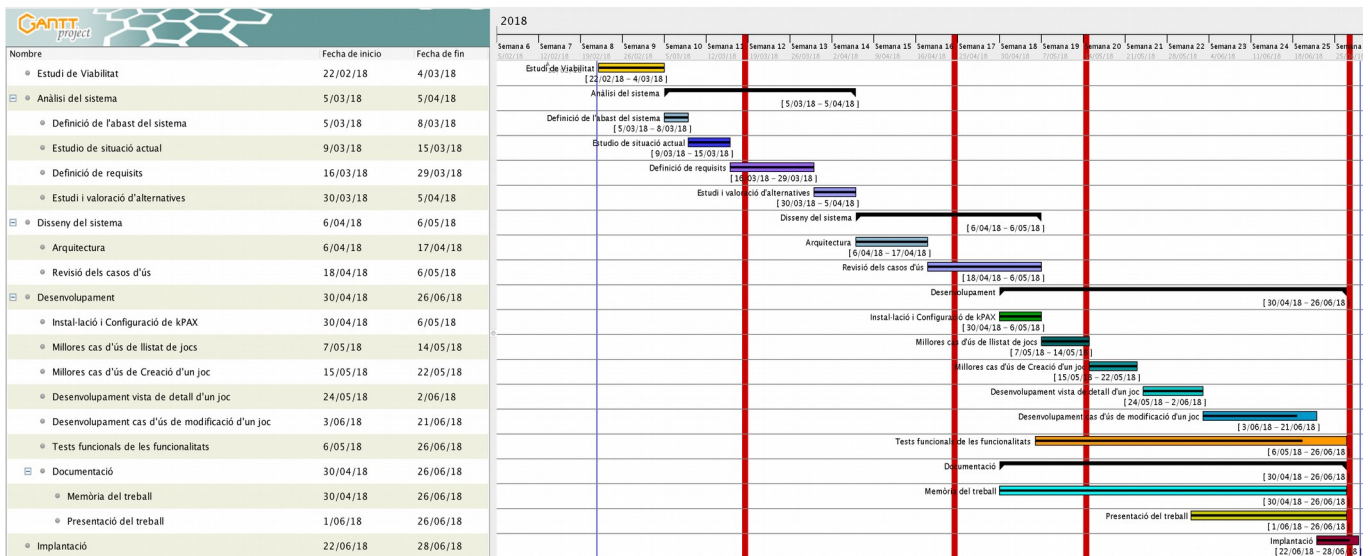
Cas d'ús 5: Esborrar un joc propi

Resum	Un usuari autènticat a la plataforma, i amb rols de desenvolupador (o administrador) ha de poder esborrar la fitxa d'un joc propi creat anteriorment.
Actors	Desenvolupador de jocs (o administrador del sistema)
Casos d'ús relacionats	L'administrador del sistema ha de validar que la sol·licitud de donar de baixa el joc és vàlida i de que ha estat sol·licitada pel seu autor.
Pre-condicions	L'usuari ha d'estar autènticat a la plataforma i amb rol de desenvolupador de jocs (o d'administrador de la plataforma) ha d'haver creat un joc seriós prèviament. S'ha de demanar a l'usuari la confirmació per la sol·licitud de donar de baixa un joc.
Post-condicions	Mostrar a l'usuari la pàgina de confirmació notificant que la sol·licitud de donar de baixa el joc ha estat correctament enregistrada. Posteriorment, s'ha de notificar a l'usuari de la validació de la sol·licitud de donar de baixa el joc.

5. Desenvolupament

5.1. Planificació de les activitats de desenvolupament i integració de sistema

A continuació, es mostra el plan temporal inicial amb les principals tasques del projecte, així com la integració del sistema.



Cal dir que la fase de desenvolupament del projecte s'ha vist alterat per una sèrie de raons que conseqüentment han produït que el *planning* (planificació temporal) inicial previst no s'hagi complert:

- El temps dedicat a l'estudi i aprenentatge a la plataforma i a les tecnologies que fa servir ha estat força més elevat del que s'havia previst inicialment. Degut als escassos coneixements d'algunes tecnologies i marcs de desenvolupament utilitzats a la plataforma i a l'escassa documentació que existeix
- El temps previst per fer la documentació ha estat una mica optimista. Finalment, s'ha dedicat més temps del previst per escriure i revisar la documentació que requereix el projecte
- Durant la fase de desenvolupament del projecte han hagut imprevisions personals i dificultats per fer el projecte de manera continua, així com per reunir-se amb el client i el consultor. Els retards han estat deguts a imprevistos personals i horaris laborals incompatibles

Tot plegat, ha produït que el temps final per desenvolupar el projecte hagi estat molt més reduït. En conseqüència, al no poder-se ajornar més la data de lliurament final amb el client, el desenvolupament d'algunes tasques finalment han quedat fora de l'abast del projecte.

Més concretament, el casos d'ús de modificació i eliminació d'un joc propi han quedat a mig desenvolupar. També s'ha dedicat menys temps a la fase de test i d'implantació.

5.2. Desenvolupament

5.1 . Preparació de l'entorn de desenvolupament

Per poder desenvolupar el codi del projecte és necessària la instal·lació d'un entorn local del projecte front-end i back-end de la plataforma. El codi dels dos projectes està allotjat en dos repositoris de *GitHub* independents. Aquests repositoris són propietat del client, tot i que són públics, per tant, es poden descarregar i contribuir al projecte sense haver de demanar permisos especials.

Cal remarcar que els dos projectes tenen un «*shell script*» d'instal·lació i configuració per la distribució de *Linux «Ubuntu 16.04»*. Això facilita bastant la fase inicial d'instal·lació i configuració dels projectes.

Concretament, l'entorn del front-end consta d'un entorn *LAMP (Linux, Apache Web Server 2, MariaDB 10 i PHP 7)* per suportar el framework *ELGG 2.2.2*, marc de programació de xarxes socials en *PHP*. D'altra banda, l'entorn de desenvolupament del back-end consta d'un servidor de *Javascript NodeJS*, i una base de dades *NoSQL (MongoDB)*, dins la distribució d'*Ubuntu*.

5.2. Eines de desenvolupament

Les eines principals emprades pel desenvolupament són les següents:

- *VirtualBox*: virtualitzadora per crear les màquines virtuals dels projectes
- *Vagrant*: per la instal·lació i configuració de les màquines virtuals dels dos projectes
- *MongoChef* i *Mongoose* per l'accés i gestió de la base de dades de MongoDB
- *MySQLWorkbench* per l'accés i gestió de la base de dades *MySQL* del front-end (*ELGG*)
- *Git* i *GitHub* per la gestió dels repositoris dels projectes
- *Postman* per la gestió de peticions a l'*API REST* del back-end
- *PHPStorm* com *IDE* per desenvolupar
- *iTerm* (consola de *MacOS*) per connectar i gestionar les màquines virtuals
- Paquet d'ofimàtica *LibreOffice* per la documentació del projecte
- Navegador web (*Chrome* i *Firefox*) per acabar de configurar l'entorn, desenvolupar i fer les proves a la interfície web de la plataforma
- *GanttProject* i *Trello* per la planificació i gestió de les tasques del projecte

5.3. Metodologia de desenvolupament

Com a metodologia de desenvolupament inicialment es va implantar una «*metodologia àgil*» semblant a **SCRUM**. És una metodologia on s'usa un desenvolupament de les tasques iteratiu amb un termini curt fixat per desenvolupar determinades tasques acordades entre el client i l'equip de desenvolupament. Al final de cada iteració o *sprint* es fa una revisió de la feina desenvolupada per donar feedback al client i lliurar les tasques acabades o per establir fer modificacions a les subseqüents iteracions del projecte.

Per motius d'incompatibilitat horària laboral i discontinuïtat en el desenvolupament del projecte degut a altres imprevisions, finalment es va canviar a una metodologia de «*desenvolupament en cascada*» (**Waterfall**).

5.4. Desenvolupament del programari

Per desenvolupar el projecte primerament es descarrega el desenvolupament iniciat del connector «*manage_games*» del repositori de *GitHub* https://github.com/fponsar/kPAX2_elgg_frontend_core.git. Després es mou el directori del connector '*kPAX2_manage_games*' al repositori principal actualitzat del front-end de *kPAX* https://github.com/drierat/kPAX2_elgg_frontend_core.git, a la branca *devel*. Finalment, es sincronitza el contingut del repositori local amb el codi allotjat a l'entorn de desenvolupament (màquina virtual).

Per la implementació del connector, es copia el mòdul (directori) '*kPAX2_manage_games*' al directori de mòduls d'*ELGG* (*/var/www/html/elgg-2.2.2/mod*). Sent el directori final a la *VM* de desenvolupament:

- */var/www/html/elgg-2.2.2/mod/kPAX2_manage_games*

A continuació, es resumeix l'estructura del connector desenvolupat seguint les directrius del marc de desenvolupament *ELGG*.

Els fitxers més rellevants són els següents:

- *manifest.xml*: aquí es troben les metadades del connector. La informació inclou les dependències, versió mínima requerida, i la llicència del connector
- *start.php*: aquest script *PHP* és el punt d'entrada on s'inicialitza el mòdul d'*ELGG*. Realitza les funcions d'inicialització i configuració com registrar les accions permeses, les controladores *MVC* de les pàgines web del connector (*pages*), les llibreries i entitats (*lib*), el menú, etc:
 - *elgg_register_action*: funció d'*Elgg* per registrar una acció al connector
 - *elgg_register_event_handler*: funció *hook* (*gantxo*) d'*ELGG* per registrar els *callbacks* (funcions de retorn) dels esdeveniments d'*ELGG* definits al connector

- *elgg_register_entity_url_handler*: funció d'ELGG per registrar les rutes (URL) que managen les entitats al connector
- *elgg_register_page_handler*: funció d'ELGG per registrar les controladores (*page controllers*) de les pàgines web del connector controlador
- *actions*: controladores PHP on es defineixen les accions del CRUD que es criden des del marc ELGG
- *languages*: aquí es defineixen les traduccions per cada idioma suportat al connector (*internacionalització*)
- *lib*: es troben les llibreries i entitats del model per generar les dades i la interfície d'usuari del connector
- *pages*: controladores de les pàgines vistes (*views*) del connector. Carreguen les dades i mostra les vistes definides al mòdul d'ELGG
- *views*: aquí es troben les vistes del connector i els estils

També s'ha tret i modificat codi del connector principal '*kPAX_core*' (*/var/www/html/elgg-2.2.2/mod/kPAX_core*), degut a que les funcionalitats de gestió de jocs inicialment van ser desenvolupades en aquest connector i les crides CRUD al servidor de *kPAX* també estan desenvolupades allà (fitxer *kpaxSrv.php*).

5.4. Pla de proves revisat

Les proves han estat fetes manualment i només s'ha pogut fer proves funcionals directament a la interfície web d'ELGG.

D'ençà que va començar el desenvolupament del projecte *kPAX* no s'han preparat ni desenvolupat proves automatitzades (ni unitàries, ni funcionals).

Cal dir, però que l'estructura i disseny que imposa el framework *ELGG* dificulta o inclòs impossibilita en molts casos el desenvolupament de proves automatitzades, sobretot les unitàries.

És per aquest motiu que personalment es recomana canviar de framework si en algun moment es planteja seriosament fer una cobertura de tests automatitzats al projecte.

Cal dir que amb les proves funcionals del connector s'ha perdut bastant temps innecessàriament ja que l'entorn i el fet d'haver d'executar-les manualment fa que la feina de testejar sigui feixuc i lent.

Finalment, cal remarcar que s'han testejat manualment les funcionalitats canviades i desenvolupades del connector «*manage_games*» de: crear un joc propi, visualitzar el llistat i la fitxa de jocs propis, i editar un joc propi.

5.5 . Principals problemes i solucions aportats

Els problemes a destacar que m'he trobat són els descrits a continuació:

- Documentació d'*ELGG* escassa i poc clara
- Documentació escassa de la plataforma
- Codi difícil de llegir i entendre
- Cal estar alerta i respectar l'ordenació i activació dels connectors degut a dependències entre ells
- Tot i que hi ha «*shell script*» per instal·lar els entorns de desenvolupament no estan adaptats per ordinadors on el host no és un Linux, com és el cas de *Windows* o *MacOS*
- S'han trobat alguns *bugs* inicials a la plataforma i als connectors que produïen incompatibilitats i inconsistències a la plataforma

Les solucions aportades són:

- S'ha afegit una configuració dels entorns de desenvolupament compatible amb altres sistemes operatius mitjançant la instal·lació de màquines virtuals per mitjà de l'eina *Vagrant*
- S'han solucionat algunes incompatibilitats entre el connector *kPAX_core* i *kPAX2_manage_games*, i incompatibilitats entre el connector *kPAX2_manage_games* i *likeKpax*
- S'han solucionat alguns *bugs* que s'han trobat inicialment al connector *kPAX2_manage_games*
- S'han desenvolupat més funcionalitats del connector *kPAX2_manage_games* i s'ha millorat l'estabilitat

6. Implantació

La implantació del connector «*kPAX2_manage_games*» actualment s'ha de dur a terme a un entorn local desenvolupament. D'una banda, perquè actualment no existeix un entorn de producció on integrar-lo i, de l'altra perquè primerament s'ha d'integrar al repositori principal del projecte a l'espera de l'acceptació del connector desenvolupat per part del client.

Cal dir que per poder implantar el connector en un entorn de desenvolupament cal dur a terme les següents tasques:

- Baixar-se el codi del connector del repositori «https://github.com/jaisato/kPAX2_elgg_frontend_core/tree/devel» o del repositori principal «https://github.com/drierat/kPAX2_elgg_frontend_core.git» un cop acceptades les modificacions (*pull request*)
- Sincronitzar o copiar el codi local descarregat del connector (directori *kPAX2_manage_games*) amb la carpeta de mòduls del framework *ELGG* (*/var/www/html/elgg-2.2.2/mod/*) instal·lat a l'entorn de desenvolupament
- Repetir la tasca anterior pel connector principal «*kPAX2_core*»
- Activar el connector «*manage_games*» si està desactivat des del panell d'administració d'*ELGG*
- Comprovar que el connector i la plataforma funcionen correctament
- Un cop revisats els canvis i el codi desenvolupat, integrar-los al repositori principal del *front-end* de *kPAX*, acceptant el *pull request* sol·licitat al client (en el cas que estigués pendent)

7. Manteniment

El manteniment del projecte i del codi desenvolupat, una vegada s'hagi lliurat al client FUOC-EIMT, correspon a ell acceptar-lo o rebutjar-lo. En cas d'acceptar-lo, s'ha d'integrar el codi nou desenvolupat al repositori principal del projecte per mitjà de l'acceptació del «*pull request*» sol·licitat al client.

Un cop integrat el desenvolupament, es pot continuar donant suport al client durant un temps raonable per tal de garantir l'estabilitat de la plataforma i assegurar que el client pot continuar amb el seu desenvolupament i manteniment de manera autònoma.

8. Conclusions

El present capítol té com objectiu principal fer una valoració i crítica global i constructiva del projecte desenvolupat. En primer lloc es fa una descripció de les lliçons apreses durant el desenvolupament del treball. Després, es fa una reflexió crítica constructiva dels objectius plantejats assolits i dels que no s'han pogut assolir. També s'analitza la metodologia inicial prevista i si finalment s'ha pogut seguir la planificació acordada inicialment. Finalment, s'exposen les línies futures de desenvolupament del projecte que queden pendents i que val la pena explorar.

Primerament, crec personalment que s'ha après molt sobre la plataforma *kPAX* i sobre les tecnologies que utilitza i amb les eines i marcs de desenvolupament en les que està basada.

També crec que la càrrega de treball duta a terme i la documentació feta del present treball és l'adequada per la càrrega lectiva que comporta i requereix l'assignatura del màster.

Respecte els objectius marcats inicialment al projecte, cal dir que els primers objectius d'aprenentatge i estudi de les tecnologies que utilitza el projecte i de la lògica de la plataforma *kPAX*, han estat assolits però malauradament han sigut massa costosos en temps i esforç. Conseqüentment, han produït que s'hagi pogut dedicar menys temps a altres objectius del projecte.

Segurament, aquesta imprecisió a subestimació dels objectius d'aprenentatge i estudi de la plataforma ha estat degut a que no s'esperava que la documentació fos tan escassa i que la informació i coneixement requerits per poder desenvolupar el projecte fossin tan grans.

També cal dir que, degut a imprevisions i incompatibilitats d'horaris de caire laboral, no s'ha pogut seguir un desenvolupament continu del projecte i ha també ha dificultat la comunicació i la realització de reunions continuades amb el client.

És per aquest motiu que també s'ha hagut de modificar la metodologia planificada inicialment al projecte. La metodologia que inicialment es va fer servir és una «*metodologia àgil*».

Tot i que aquesta metodologia en un principi és la més adequada, per la naturalesa del projecte i per la duració del treball, malauradament no es va poder seguir amb rigorositat durant el desenvolupament del projecte. Aquesta metodologia és iterativa i continua, utilitza terminis curts i fixats de temps per desenvolupar i lliurar les tasques al client establint una revisió al final de cada iteració. Com ja s'ha exposat, aquesta manera de

treballar no es va poder seguir degut a les incompatibilitats horàries i imprevistos laborals sorgits.

Per això, més endavant es va canviar a una metodologia de desenvolupament més clàssica, anomenada «*desenvolupament en cascada*» (*model Waterfall*).

Cal dir, que tot i que no es va poder seguir un desenvolupament iteratiu i continu es van poder realitzar algunes reunions amb el consultor i el client, així com una comunicació suficient per correu electrònic.

Per tant, cal dir que tot i que els objectius d'aprenentatge i de documentació s'han assolit, la planificació establerta inicialment i els objectius de desenvolupament de funcionalitats plantejats inicialment no s'ha pogut assolir.

Concretament, no s'ha pogut dur a terme la funcionalitat de donar de baixar un joc propi i, el cas d'ús de modificació d'un joc requereix d'algunes modificacions perquè sigui del tot estable. També crec que es convenient fer algunes proves funcionals més.

Per concloure, crec ha estat una experiència enriquidora, on s'han après diverses tecnologies, i s'ha tingut l'oportunitat de contribuir a la plataforma *kPAX*. Finalment, m'hagués agradat disposar de més temps per finalitzar el desenvolupament del connector i corregir alguns *bugs* de la plataforma. Tasques que no ha estat possible dur a terme pels motius exposats més amunt.

9. Glossari

1. **UOC:** *Universitat Oberta de Catalunya*
2. **FUOC-EIMT:** *Facultat d'Estudis d'Informàtica, Multimedia i Telecomunicacions de la Universitat Oberta de Catalunya*
3. **kPAX:** *xarxa social de software lliure basada en ELGG pel desenvolupament i gestió de jocs d'aprenentatge seriosos*
4. **ELGG:** *framework de software lliure pel desenvolupament de serveis i xarxes socials en llenguatge PHP*
5. **PHP (PHP Hypertext Preprocessor):** *llenguatge de programació d'scripting i de propòsit general del costat del servidor dissenyat especialment pel desenvolupament web de contingut dinàmic*
6. **Javascript:** *llenguatge de programació interpretat, orientat a objectes, prototipat, imperatiu, tipat dèbilment i dinàmic. S'utilitza al costat del client i del servidor amb el motor Node.js*
7. **NodeJS:** *entorn multiplataforma per a la capa del servidor basat en el llenguatge de programació Javascript (especificació ECMAScript). Arquitectura asíncrona i orientada a events. Basat en el motor V8 de Google*
8. **ECMAScript:** *eppecificació del llenguatge de programació publicat per ECMA International. Basat en el popular llenguatge de programació JavaScript proposat com a estàndar per Netscape Communications Corporation*
9. **MariaDB:** *sistema gestor de base de dades relacional derivat del gestor MySQL amb llicència GPL (software lliure)*
10. **MongoDB:** *és un sistema de base de dade NoSQL, orientat a documents, desenvolupat sota el concepte de codi obert (software lliure)*
11. **Base de dades NoSQL (o no relacional):** *bases de dades que no segueixen el model algebraic relacional típic dels gestors de base dades clàssics*
12. **GPL (GNU General Public License):** *llicència de programari lliure ampliament utilitzat al desenvolupament de software lliure. Licència creada originalment per Richard Stallman, fundador de la Free Software Foundation (FSF)*
13. **MIT (llicència):** *llicència de software lliure que es va originar a l'Institut Tecnològic de Massachusetts (MIT, Massachusetts Institute of Technology) als anys '80*
14. **Connector (plug-in):** *mòdul o component de software informàtic que es relaciona amb altre per afegir funcionalitats noves i generalment molt específiques*
15. **Shell script:** *programa informàtic d'arxiu de comandes, o arxiu de processament per lots. Generalment emmagatzemat a un arxiu de text pla i interpretat per una consola o intèrpret de comandes*
16. **Java:** *és un llenguatge de programació de propòsit general, concurrent, orientat a objectes i multiplataforma molt popular. Va ser dissenyat per Sun Microsystems l'any 1982 però actualment és propietat de la companyia Oracle*
17. **Metodologia àgil (de desenvolupament):** *enfoc de desenvolupament de software que requereix d'un esforç de col·laboració continu i creuat entre*

equips de desenvolupament i clients. Avoca per una planificació adaptativa, iterativa, i evolutiva del desenvolupament, i per una publicació ràpida i continuada del programari. Segueixen els valors i principis del manifest «Manifesto for Agile Software Development»

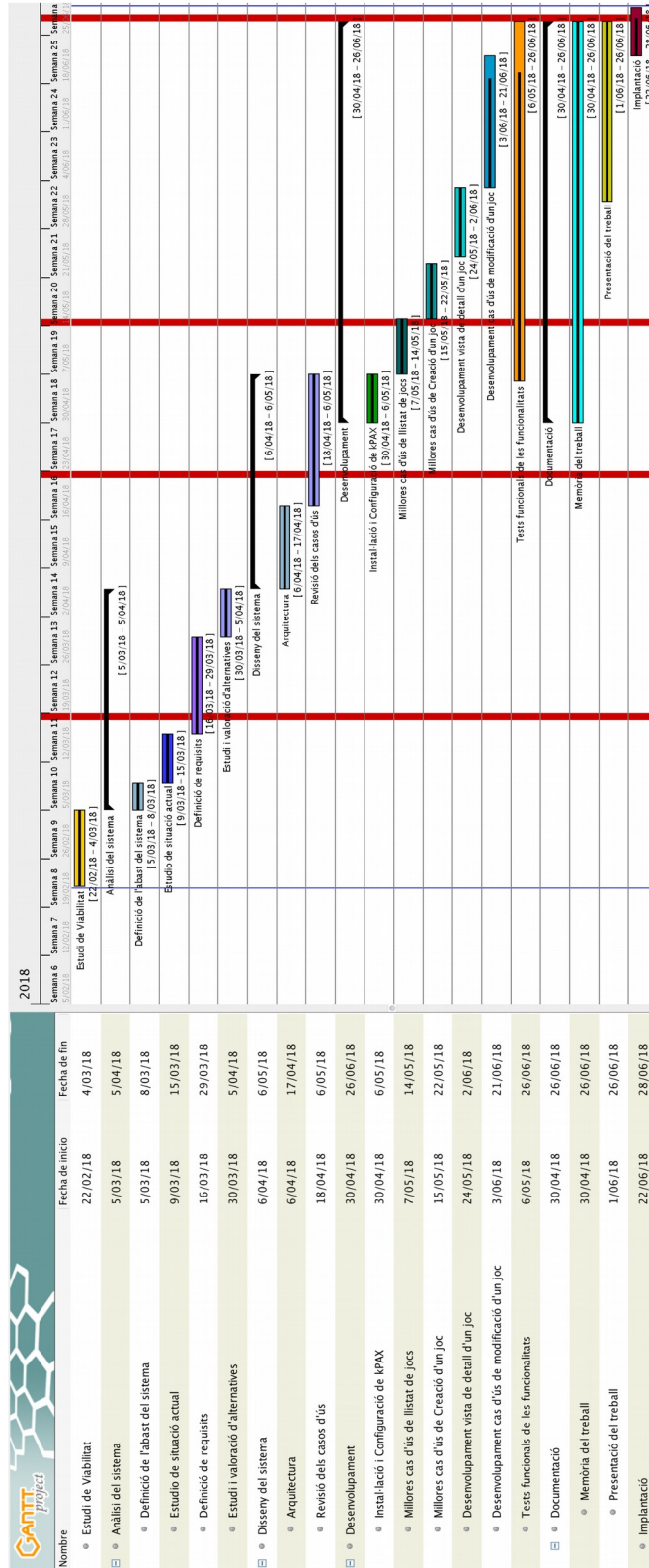
18. **Waterfall (model):** *és un model clàssic de desenvolupament de software amb un enfoc de disseny seqüencial i linial. Tendeix a ser un enfoc menys iteratiu i flexible que les metodologies de desenvolupament àgils*

10. Referència bibliogràfica

- *José Ramón Rodríguez (coordinador) i Pere Mariné Jové*. Ref. PID_00153520, “Gestió de projectes”, UOC
- *David Megías Jiménez (coordinador), Jordi Mas (coordinador) i, Alberto Otero García (autor)*. Ref. 90.803_a, “Projecte Web”, UOC
- *Roser Beneito Montagut*. Ref. P08/19018/00446, “Presentació de documents i elaboració de presentacions”. UOC
- *Nita Sáenz Higuera, Rut Vidal Oltra*. Ref. P08/19018/00445, “Redacció de textos científicotècnics”, UOC
- *Daniel Riera, J. Arnedo-Moreno*. kPAX: Experience on designing a gamified platform for serious gaming. EIMT, Open University of Catalonia, Barcelona, Spain, 2016
- *Francesc Pons Aróztegui*, kPAX2 Plug-in manage_games comprimit, Treball Final de Màster, Màster Oficial de Programari Lliure, UOC, <http://openaccess.uoc.edu/webapps/o2/handle/10609/72868>
- <https://elgg.org/>, ELGG framework, 2018
- <http://learn.elgg.org/en/stable/tutorials/index.html>, ELGG learn, 2018
- <http://learn.elgg.org/en/2.0/guides/>, ELGG programming guides, 2018
- <https://github.com/>, GitHub, Git repository server, 2018
- <http://php.net/manual/en/>, PHP documentation, 2018
- <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference>, Javascript documentation, 2018
- <https://nodejs.org/en/>, NodeJS, 2018
- <https://nodejs.org/docs/latest-v6.x/api/>, Nodejs 6 documentation, 2018
- <https://www.mongodb.com/>, MongoDB, 2018
- <https://docs.mongodb.com/manual/>, MongoDB manual, 2018
- <https://httpd.apache.org/docs/current/>, Apache Web Server documentation, 2018
- <https://mariadb.com/>, MariaDB, 2018
- <https://mariadb.com/kb/en/library/documentation/>, MariaDB documentation, 2018
- <https://www.vagrantup.com/>, Vagrant, 2018
- <https://www.ganttproject.biz/>, GanttProject, 2018
- <https://en.wikipedia.org/>, Wikipedia, 2018
- [https://en.wikipedia.org/wiki/Elgg_\(software\)](https://en.wikipedia.org/wiki/Elgg_(software)), ELGG, 2018
- <https://en.wikipedia.org/wiki/PHP>, PHP, 2018
- [https://en.wikipedia.org/wiki/Java_\(programming_language\)](https://en.wikipedia.org/wiki/Java_(programming_language)), Java, 2018
- <https://en.wikipedia.org/wiki/JavaScript>, Javascript, 2018
- <https://en.wikipedia.org/wiki/ECMAScript>, ECMAScript, 2018
- <https://en.wikipedia.org/wiki/Node.js>, Node.js, 2018
- <https://en.wikipedia.org/wiki/MariaDB>, MariaDB, 2018
- <https://en.wikipedia.org/wiki/MongoDB>, MongoDB, 2018
- https://en.wikipedia.org/wiki/Agile_software_development, Agile Software Development, 2018
- https://en.wikipedia.org/wiki/Waterfall_model, Waterfall model, 2018
- <https://www.fsf.org/licensing/>, FSF Licensing, 2018
- <https://www.libreoffice.org/>, LibreOffice, 2018
- <https://help.libreoffice.org/>, LibreOffice Help, 2018

11. Annexos

Annex I. Planificació temporal inicial



Annex II. Planificació temporal desenvolupament final



Annex III. Repositoris Git/GitHub del projecte

Els repositoris *Git* principals del projecte són d'en Dani Riera i Terrén (FUOC-EIMT):

- *Front-End*: https://github.com/drierat/kPAX2_elgg_frontend_core
- *Servidor*: https://github.com/drierat/kPAX2_server

Altres repositoris consultats:

- *Front-end kPAX2_manage_games* (fork d'en Francesc Pons Aróztegui): https://github.com/fponsar/kPAX2_elgg_frontend_core

Repositoris del desenvolupament d'aquest projecte:

- *Front-end kPAX2_manage_games* (fork): https://github.com/jaisato/kPAX2_elgg_frontend_core/
- *kPAX2 server* (fork): https://github.com/jaisato/kPAX2_server

Annex IV. Repositoris O2 de kPAX

Repositoris de *kPAX* allotjats al *Repositori Institucional (O2)* de la *UOC*:

- <http://openaccess.uoc.edu/webapps/o2/>

Títol: «kPAX2 Plug-in manage_games comprimit»

Autor: Pons Aróztegui, Francesc

Data de publicació: 25/01/2018

URI: <http://hdl.handle.net/10609/72868>

Títol: «Perfil d'usuari per la plataforma KPAX2»

Autor: Martínez Terés, Roger

Data de publicació: 07/2017

URI: <http://hdl.handle.net/10609/65525>

Títol: «Mejora en el listado de juegos de kPax»

Autor: Ramirez Sanchez, Javier

Data de publicació: 01/07/2017

URI: <http://hdl.handle.net/10609/65566>

Títol: «KPAX2: Instal·lació a Ubuntu 16.04LTS, revisió i adaptació de connectors, revisió CSS»

Autor: Olariaga Rodríguez, Ernesto

Data de publicació: 25/01/2017

URI: <http://hdl.handle.net/10609/60165>

Títol: «Kpax: Migración a Elgg 2.1.1»

Autor: Vinuesa Sánchez, Rubén

Data de publicació: 19/06/2016

URI: <http://hdl.handle.net/10609/52702>

Títol: «kPax plataforma d'aprenentatge en xarxa: Migració de kPaxServer a NodeJS amb MongoDB»

Autor: Muntaner Morey, Miquel A.

Data de publicació: 19/06/2016

URI: <http://hdl.handle.net/10609/52741>

Títol: «Mercat d'aplicacions per a la xarxa social kPAX»

Autor: Barceló Martí, Marc

Data de publicació: 24/06/2015

URI: <http://hdl.handle.net/10609/43063>

Títol: «Incorporació de funcionalitats a la xarxa social kPAX: Autenticació amb la UOC»

Autor: Pimentel Segura, Oscar

Data de publicació: jun-2015

URI: <http://hdl.handle.net/10609/42772>

Títol: «Incorporación de funcionalidades a la red social kPAX»
Autor: Brocca Freydoz, Juan Carlos
Data de publicació: 15/06/2015
URI: <http://hdl.handle.net/10609/42726>

Títol: «Integración de módulos a la red kPAX»
Autor: Camera López, Cecilia Edith
Data de publicació: 15/06/2015
URI: <http://hdl.handle.net/10609/42744>

Títol: «Incorporació de funcionalitats al projecte kPAX»
Autor: Corral Blanch, Víctor
Data de publicació: 03/07/2014
URI: <http://hdl.handle.net/10609/35341>

Títol: «La gestió dels jocs a kPAX»
Autor: Frau Aguiló, Sion Xavier
Data de publicació: 25/06/2014
URI: <http://hdl.handle.net/10609/32981>

Títol: «Ampliación de funcionalidades para la red KPAX»
Autor: Carrera Formoso, Eugenia
Data de publicació: 19/01/2014
URI: <http://hdl.handle.net/10609/29821>

Títol: «Perfil complet d'usuari per a la xarxa KPAX»
Autor: Farrerons i Herrero, Joan
Data de publicació: 12/01/2014
URI: <http://hdl.handle.net/10609/29902>

Títol: «Mòdul de cerca de jocs per a la plataforma kPAX»
Autor: Giró Oriol, Albert
Data de publicació: 06/2013
URI: <http://hdl.handle.net/10609/22443>

Títol: «Incorporació de funcionalitats a la xarxa social kPAX: autenticació amb Facebook»
Autor: Catalá Jiménez, Javier
Data de publicació: 06/2013
URI: <http://hdl.handle.net/10609/22524>