



MASTER DE PROGRAMARI LLIURE

TREBALL FINAL DE MASTER

Perfil d'usuari per la plataforma KPAX2

Alumne: Roger Martínez Terés

Tutor de la pràctica: Daniel Riera i Terrén

Consultor de l'assignatura: Francisco Javier Noguera Otero

Data de presentació: 27/06/2017

Llicència de publicació del document

Aquesta obra es troba subjecta a les condicions de la llicència lliure: **CC BY-SA 2.0**

Les clàusules de la CC BY-SA 2.0 en el text original en anglès es poden consultar a la següent pàgina web:

<https://creativecommons.org/licenses/by-sa/2.0/>

Resum del projecte

El projecte "Perfil d'usuari per la plataforma KPAX2[1]" es tracta de la implementació d'un perfil d'usuari per a la plataforma KPAX2. Consta principalment de dos grans apartats:

- Creació de l'API REST
- Modificació de la part client a la plataforma ELGG[2]

El primer apartat consta de la implementació dels serveis API RESTFUL[3] a la part de servidor. Aquesta implementació ha estat realitzada amb les tecnologies punteres de Node.js[4] i MongoDB[5]. Com tota API RESTFUL, consta de quatre serveis principals: GET, per obtenir el o els perfils d'usuaris, POST, per crear un nou perfil d'usuari, PUT per modificar-los i DELETE per esborrar-los.

El segon apartat consta de la implementació d'un connector dins la plataforma ELGG per poder utilitzar els serveis anteriorment descrits i mostrar-los a l'usuari en forma de pàgina web.

Per fer aquest perfil orientat a la plataforma KPAX2 i la seva ludificació, s'ha implementat seguin un model semblant a un joc de rol, aquest cas del joc de Rol Advanced Dungeons and Dragons, separant les habilitats dels jugadors en cinc característiques principals: força, agilitat, intel·ligència, carisma i voluntat i els seus coneixements en les matèries com experteses en matemàtiques, informàtica, etc.

TAULA DE CONTIGUT

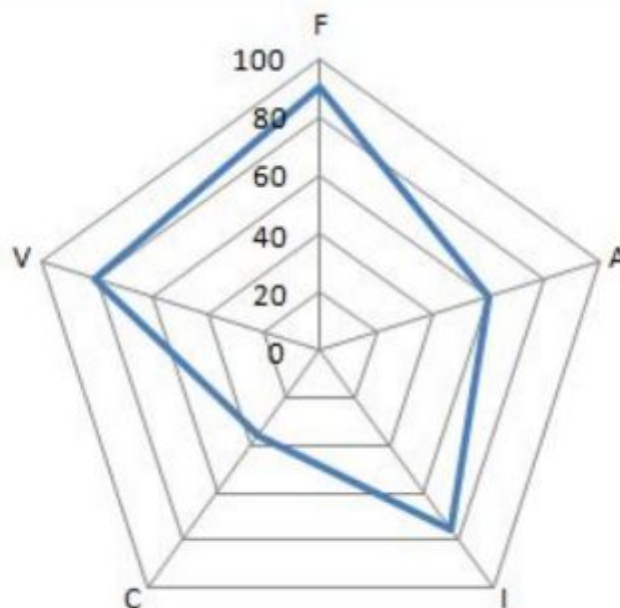
PERFIL D'USUARI PER LA PLATAFORMA KPAX2

1. Introducció	6
2. Estudi de viabilitat	10
2.1. Establiment de l'abast del sistema	10
2.2. Estudi de la situació actual	11
2.3. Establiment de requisits	11
3. Anàlisi	12
3.1. Descripció general del procés	12
3.2. Els actors	12
3.3. Els requisits del sistema	12
3.4 Model de casos d'us	12
4. Disseny	15
4.1 Introducció	15
4.2 L'arquitectura del sistema	16
Des del punt de vista logic (identificació de patrons de disseny):	17
Des del punt de vista físic:	18
Convencions de disseny	18
4.3 Detall de les modificacions a realitzar	19
5. Desenvolupament	19
5.1. Desenvolupament del connector anomenat KPAX2_profile	19
5.2. Desenvolupament dels serveis REST	20
5.3. Desenvolupament en KPAX2_server	20
5.4. Creació de la colecció de la base de dades	21
6. Implantació	21
6.1. Operacions necessàries per passar a l'entorn de producció	21
6.2. Punts importants a tenir en compte	22
7. Manteniment	22
8. Conclusions	23
9. Referències bibliogràfiques i de recursos web	24
ANNEXOS	26

1. Introducció

El projecte consistirà en el desenvolupament d'un connector per al sistema d'aprenentatge digital KPAX2. El connector que es desenvoluparà és el perfil d'usuari, aquest perfil estarà basat en la metàfora de personatge de joc de rol, amb 5 característiques principals, força, agilitat, intel·ligència, carisma i voluntat que es calcularan a partir de la participació de l'usuari en els jocs en els quals hagi participat dins la plataforma KPAX2.

S'hauran de definir les funcions específiques que defineix cadascuna d'aquestes característiques i representar-les tant en format percentual com en format gràfic, per exemple:



A més de les característiques principals, cada usuari tindrà expertesa en les matèries que aporta KPAX2 (matemàtiques, física, química, naturals, història, informàtica...). Per cada matèria es podran assolir cinc nivells d'expertesa, els quals aniran canviant en el temps, millorant o degradant dependent de les fites assolides per l'usuari. Per cada matèria hi hauran 4 valors associats:

- Nivell d'expertesa actual
- Temps al nivell d'expertesa actual
- Nivell màxim d'expertesa assolit
- Temps en el nivell màxim

Els nivells proposats serien:

- C (bronze)
- B (plata)
- A (or)
- A+ (rubí)
- S (diamant)

L'expertesa es calcularà en temps real, pels punts assolits en els jocs, per assolir els nivells C i B. Els nivells A, A+ i S seran més difícil d'assolir i es calcularan només un cop per setmana, per tant, si en una setmana no es juga, aquesta expertesa es degradarà.

El perfil tindrà tres zones organitzades verticalment:

- Informació del jugador
- Característiques
- Experteses

Seguidament es mostra una aproximació:

imatge

PERSONATGE : _____

FIGURA : _____

ANTIGUITAT : _____

- CARACTERÍSTIQUES -

x FORÇA: ☐
 x AGILITAT: ☐
 x INTEL·LIG.: ☐
 x CARISMA: ☐
 x VOLUNTAT: ☐

- EXPERTESIES -

	ACTUAL	TEMPS	MÀX.	TEMPS	XP	XPsetmanes
◦ MATEMÀTIQUES	<input type="radio"/>	xx	<input type="radio"/>	xx	xxx	x
◦ INFORMÀTICA	<input type="radio"/>	xx	<input type="radio"/>	xx	xxx	x
⋮						

Per aconseguir aquesta interfície gràfica, s'implementaran serveis REST per a obtenir les dades d'un servidor remot implementat en Node.js amb una base de dades MongoDB.

Els serveis a implementar seran 4:

- GET: /user/{id} que obtindrà les dades d'un usuari concret
- POST: /user que crearà un perfil d'usuari nou
- PUT: /user/{id} per modificar les dades d'un usuari
- DELETE: /user/{id} per esborrar a un usuari

El connector d'ELGG es connectarà amb aquests serveis per obtenir les dades.

Objectius del projecte

Els objectius del projecte els podem classificar en tres tipus: objectius docents, objectius generals i objectius específics.

A continuació els detallarem:

Objectius docents:

- Estudi i comprensió de la plataforma KPAX2
- Programació de connectors d'ELGG per KPAX2
- Integració dels connectors a KPAX2
- Tests funcionals dels connectors
- Implementació dels serveis necessaris al nucli de KPAX2, en Node.js i MongoDB
- Integració del connector amb els serveis

Objectius generals de la plataforma KPAX2:

- Facilitar el suport al treball i l'avaluació de competències mitjançant l'aprenentatge
- Fer us de les possibilitats que ofereixen les tecnologies mòbils
- Facilitar l'ús de les xarxes socials per afegir relacions virtuals multidispositiu
- Aconseguir que els resultats dels projectes d'innovació siguin transversals, aplicables a diversos àmbits, assignatures i estudis
- Construir una xarxa social que accepti usuaris externs a la UOC, donant visibilitat a la universitat i es converteixi en una eina tan intrauniversitària com extrauniversitària.

Objectius específics:

- Afegir una nova funcionalitat a la xarxa KPAX2: el perfil d'usuari
- Dissenyar un perfil que proporcioni informació sobre els atributs i les experteses assolides per l'usuari
- Implementar els serveis necessaris al nucli per tal d'obtenir aquesta informació de l'usuari
- Poder modificar les dades personals de l'usuari

Planificació del projecte

Dins del projecte es consideraran diferents tasques.

Les primeres seran de planificació, on es definiran els objectius del projecte, es farà la definició dels requeriments i de les tasques i una planificació temporal.

En un segon moment, es faran les tasques d'execució, tan prèvies al desenvolupament, com la instal·lació i configuració del programari com el disseny conceptual i les tasques pròpies de desenvolupament, on estan la implementació dels serveis del nucli de KPAX2 com la implementació del connector d'ELGG. A més, es faran les proves de funcionament.

Per acabar amb les tasques de documentació, on es farà la memòria final.

2. Estudi de viabilitat

2.1. Establiment de l'abast del sistema

La plataforma KPAX2 es troba encara en fase de desenvolupament, implementant alguns connectors per afegir més funcionalitats a la plataforma.

L'arquitectura triada per aquesta segona versió és la següent: un conjunt de serveis implementats en un servidor amb tecnologia Node.js, que accedeixen a la nova base de dades noSQL, MongoDB i finalment, un mecanisme de crides a aquests serveis en llenguatge PHP[6], a través d'una biblioteca centralitzada ubicada en el connector d'ELGG anomenat KPAX2.

Una de les necessitats actuals del client és la d'afegir una nova funcionalitat a la xarxa KPAX2: el desenvolupament d'un perfil d'usuari. Es tracta de dissenyar un perfil que proporcioni informació sobre les experteses assolides pels usuaris jugant a jocs i les seves característiques (força, destresa...). En definitiva, hauria de poder visualitzar-se en pantalla la següent informació:

- Dades personals de l'usuari: avatar, nickname, nom,...
- Característiques i els seus valors, numerics i en format gràfic
- Experteses aconseguides

Des de el punt de vista legal, cal desenvolupar una aplicació totalment compatible i integrable amb la plataforma KPAX2 i que, per consegüent haurà d'estar subjecta a la mateixa llicència lliure GPL[7].

2.2. Estudi de la situació actual

La plataforma KPAX2 té una arquitectura basada en connectors d'ELGG. Aquests connectors són els que permeten desenvolupar jocs i aplicacions externes que interactuin amb el nucli. Per la seva banda, el "nucli" s'encarrega de mantenir les dades en la base de dades noSQL.

En el nucli s'implementen els serveis REST per tal que els connectors puguin interactuar. Aquests serveis són implementant en Node.js.

Aquests serveis fan referencia a usuaris, jocs, desenvolupadors, etc.

2.3. Establiment de requisits

Per una banda, es tracta crear les operacions API al servidor per tal d'obtindre, modificar, eliminar i inserir el perfil d'usuari a la base de dades MongoDB. Per altra banda, es tracta de crear un connector que permeti mostrar a l'usuari els conceptes que s'han esmentat anteriorment en l'apartat d'objectius específics.

3. Anàlisi

3.1. Descripció general del procés

L'objectiu d'aquesta nova funcionalitat per a la plataforma és que l'usuari, un cop hagi de accedir a la pantalla del seu perfil personal pugui:

- Editar la seva imatge
- Editar les seves dades personals
- Veure les característiques obtingudes per la participació en els diferents jocs
- Veure les experteses obtingudes per la participació en els diferents jocs

3.2. Els actors

Existeixen dos actors en el sistema. El primer és l'administrador qui disposa d'accés a totes les dades del sistema i és la persona (o persones) encarregada de les tasques de control i supervisió i té la responsabilitat de controlar l'estat del sistema. El segon és l'usuari normal, que no té privilegis especials i pot accedir als jocs i al seu perfil.

3.3. Els requisits del sistema

S'han definit dos requisits en el sistema:

Primer requisit (R1): sistema d'identificació d'usuari. La plataforma KPAX2 ja ofereix aquest servei. Qualsevol connector pot obtenir el nom d'usuari registrat mitjançant el nucli d'ELGG

Segon requisit (R2): sistema de permisos per distingir usuaris. La plataforma KPAX2 ofereix aquest servei a través del nucli d'ELGG, que distingeix tres tipus d'usuari:

- Logged in
- Not logged in
- Administrator

3.4 Model de casos d'us

Es preveuen els diferents casos d'us:

- Presentació de l'informació del perfil
 - Dades personals
 - Atributs
 - Experteces
- Edició de les dades personals
- Edició de l'imatge de perfil
- Presentació gràfica dels atributs

A continuació s'expliquen amb més detall aquests casos d'us:

Nom	Presentació de l'informació del perfil
Número	1
Precondicions	L'usuari es troba a la pantalla de perfil.
Postcondicions	Es mostra el perfil d'usuari
Escenari principal	1. L'usuari fa login 2. L'usuari clica al perfil
Freqüència	Alta
Importancia	Alta
Urgencia	Alta
Requisits	R1 i R2

Nom	Edició de les dades personals
Número	2
Precondicions	L'usuari es troba a la pantalla de edició de dades personals.
Postcondicions	Les dades personals han estat canviades
Escenari principal	1. L'usuari fa login 2. L'usuari clica al perfil 3. L'usuari clica a modificar les dades 4. L'usuari modifica les dades 5. L'usuari clica al botó de enviar
Freqüència	Baixa

Importancia	Mitja
Urgencia	Mitja
Requisits	R1 i R2

Nom	Edició de l'imatge de perfil
Número	3
Precondicions	L'usuari es troba a la pantalla de edició de dades personals.
Postcondicions	L'imatge ha estat canviada
Escenari principal	<ol style="list-style-type: none"> 1. L'usuari fa login 2. L'usuari clica al perfil 3. L'usuari clica a modificar la imatge 4. L'usuari modifica la imatge 5. L'usuari clica al botó de enviar
Freqüència	Baixa
Importancia	Mitja
Urgencia	Baixa
Requisits	R1 i R2

Nom	Presentació gràfica dels atributs
Número	4
Precondicions	L'usuari es troba a la pantalla del perfil d'usuari
Postcondicions	Es mostren els atributs en forma gràfica
Escenari principal	<ol style="list-style-type: none"> 1. L'usuari fa login 2. L'usuari clica al perfil
Freqüència	Baixa
Importancia	Mitja
Urgencia	Baixa
Requisits	R1 i R2

4. Disseny

4.1 Introducció

Un cop definits els requisits del sistema, els casos d'ús, i el resultat esperat, l'objectiu de la fase de disseny és obtenir els models i les especificacions necessaris per al desenvolupament i la integració i també per definir l'entorn de proves i Implantació del sistema.

4.2 L'arquitectura del sistema

A continuació es detallen els blocs què componen el sistema:

Pel que fa a la capa visual de l'aplicació, l'arquitectura ja ve donada pel nucli d'ELGG, igualment que la seva base de dades.

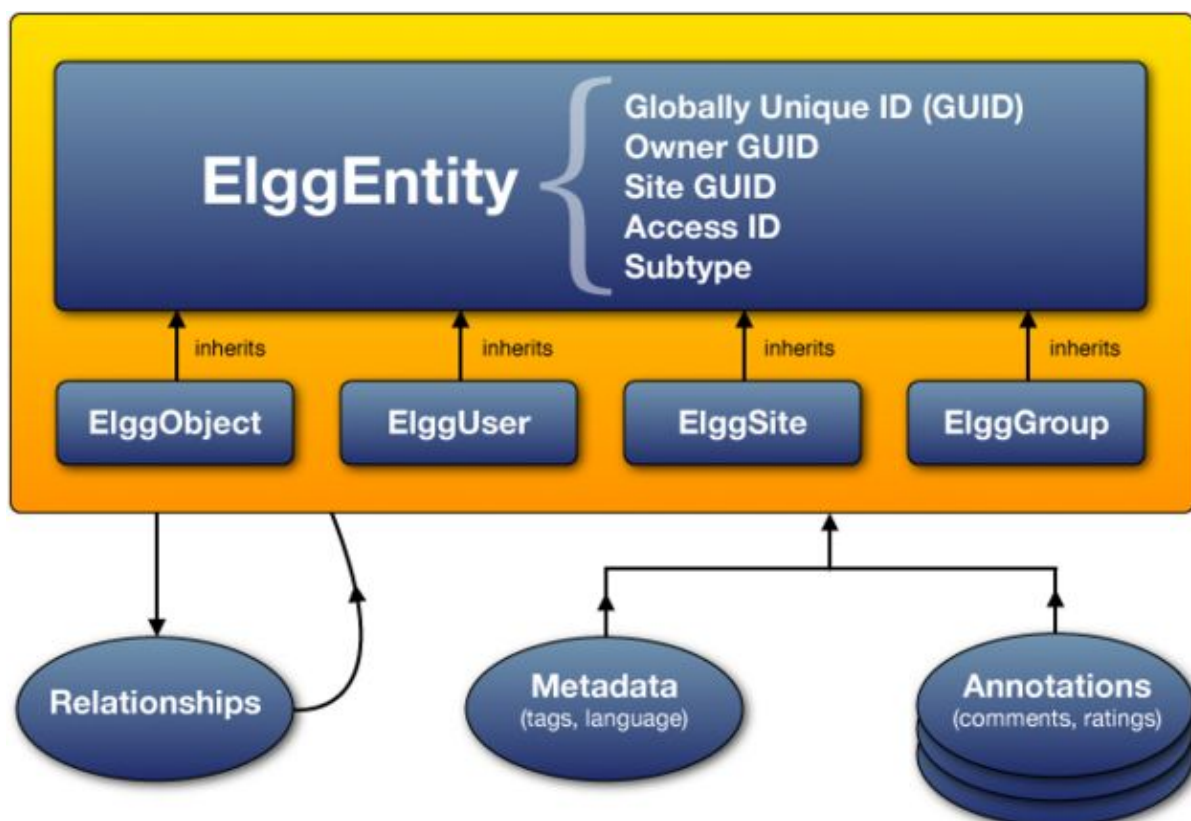


Figura: Esquema de la base de dades d'[ELGG](#)

Pel que fa a la capa de serveis o lògica de negoci, està formada per un servidor que executa Node.js per implementar les crides API REST que seran executades pels clients (interfície gràfica), aquest client és el connector que s'implementa a ELGG.

Per últim, en la capa de persistència de dades s'utilitzarà una base de dades mongoDB per guardar les diferents "col·leccions", en el nostre cas, la col·lecció "userprofile", que es representa així:

```
{
  "_id" : ObjectId("58f61aad98821063122b2fa5"),
  "name" : "Roger",
  "nickname" : "RuNcH",
  "created_at" : 20170604,
  "uodated_at" : 20170604,
  "photo_url" : "/foto.jpg",
  "attributes" : {
    "strength" : 80,
    "agility" : 80,
    "intelligence" : 80,
    "charisma" : 80,
    "will" : 80
  },
  "abilities" : [
    {
      "name" : "mathematics",
      "level" : "A+",
      "time_level" : 36000,
      "max_level" : "S",
      "time_max_level" : 3450,
      "experience" : 5000
    },
    {
      "name" : "computing",
      "level" : "S",
      "timelevel" : 36000,
      "maxlevel" : "S",
      "timemaxlevel" : 6000,
      "experience" : 8000
    },
    {
      "name" : "philosophy",
      "level" : "C",
      "timelevel" : 3600,
      "maxlevel" : "B",
      "timemaxlevel" : 340,
      "experience" : 400
    }
  ]
}
```

Figura: Representació de dades a MongoDB

Des del punt de vista lògic (identificació de patrons de disseny):

Pel que fa al conjunt del sistema, el patró de disseny es basa en l'arquitectura de 3 capes: presentació, lògica de negoci i persistència.

La capa de presentació

- Està formada per codi d'ELGG en PHP, tant el nucli com els connectors propis de KPAX2
- Aquesta capa s'encarrega de proporcionar l'interfície gràfica de l'usuari.

La capa de lògica de negoci

- Aquesta capa construeix el nucli que controla el funcionament de l'aplicació: recull les dades facilitades pels usuaris i comprova la seva validesa. També és l'encarregada d'autoritzar als usuaris a fer les peticions corresponents i efectua la lògica necessària per a arribar al resultat a enviar tant a la capa de persistència com a la capa de presentació
- En cas d'error, és l'encarregada d'informar a la capa de presentació de l'error donat perquè l'usuari en sigui conscient (per exemple un nom d'usuari que ja existeix...)

La capa de persistència

- Aquesta capa s'encarrega d'emmagatzemar les dades en col·leccions de KPAX2 en MongoDB
- Realitzar consultes i actualitzacions de dades a la base de dades MongoDB

Des del punt de vista físic:

Capa de presentació:

- Codi d'ELGG en llenguatge PHP
- Codi dels connectors d'ELGG per crear la xarxa KPAX2, creant la interfície gràfica que permet interactuar amb la plataforma

Capa de lògica de negoci:

- Implementació dels serveis REST en Node.js necessaris per a la comunicació amb els connectors d'ELGG
- Implementació de les connexions amb la capa de persistència per tal d'inserir o recuperar dades

Capa de persistència:

- Implementació de les crides a la base de dades NoSQL MongoDB per tal d'obtenir, crear, modificar o esborrar les dades, operacions CRUD

Convencions de disseny

En primer lloc cal remarcar que s'ha optat per aplicar les mateixes estratègies que s'han aplicat en els connectors anteriors implementats per programadors anteriors i que es poden observar en el codi preexistent.

Aquest disseny és basat en objectes JSON, fàcilment tractables tant en la capa de persistència en mongoDB com a la capa de lògica, treballant nativament amb javascript a través de Node.js.

4.3 Detall de les modificacions a realitzar

Pel que fa a la base de dades de KPAX2:

- Caldrà crear una colecció que representi les dades del perfil d'usuari, al ser una base de dades NoSQL, no caldrà implementar cap relació entre taules (mirar figura anterior).

Pel que fa al connector d'ELGG:

- Caldrà crear un connector que permeti l'accés a la pagina del perfil, tot clicant a l'avatar a la part superior de la pàgina. Aquest connector tindrà que permetre la modificació tan de les dades personals com de l'avatar de l'usuari.

Pel que fa als nous serveis web que s'implementaran:

- Caldrà crear un servei que retorni el perfil d'usuari complet
- Caldrà crear un servei que retorni perfils d'usuari mitjançant un filtre (query)
- Caldrà crear un servei per tal de modificar les dades personals i l'avatar
- Caldrà crear un servei per poder borrar un perfil

5. Desenvolupament

5.1. Desenvolupament del connector anomenat KPAX2_profile

S'ha implementat un connector que sobreescriu l'original d'ELGG per tal de tindre el control complet del perfil d'usuari. Aquest connector mostra tant l'avatar de l'usuari com la informació del perfil:

- Dades personals
- Atributs
- Experteses

Aquest connector també permetrà la modificació de les dades personals i de l'avatar.

5.2. Desenvolupament dels serveis REST

Per tal d'implementar les crides als serveis REST, codificades en PHP, s'ha d'implementar aquestes noves crides:

- Crida per obtenir el perfil d'usuari complet
- Crida per modificar les dades personals d'usuari
- Crida per modificar l'avatar

En la part del nucli, s'han d'implementar en codi Javascript en Node.js:

- Crida per obtenir el perfil d'usuari
- Crida per buscar perfils d'usuari
- Crida per modificar un perfil d'usuari
- Crida per esborrar un perfil d'usuari

5.3. Desenvolupament en KPAX2_server

Fitxers que s'han agut de modificar:

- El fitxer server/routes/index.js per tal d'afegir una nova ruta per les operacions API REST, la ruta és /userprofile

Fitxers que s'han agut de crear:

- El fitxer server/routes/userprofile.js on s'implementen les operacions API REST que han de veure amb el perfil d'usuari

5.4. Creació de la col·lecció de la base de dades

- S'ha creat una col·lecció nova a mongoDB pel perfil d'usuari: "userprofile" (figura anterior)
- Aquesta col·lecció té la informació següent:
 - `_id`
 - `Name`
 - `Nickname`
 - `Created_at`
 - `Updated_at`
 - `Photo_url`
 - `Attributes`
 - `Abilities`

(1) ObjectId("58f61aad98821063122b2fa5")	{ 8 fields }	Object
_id	ObjectId("58f61aad98821063122b2fa5")	ObjectId
name	Roger	String
nickname	RuNcH	String
created_at	20170604	Int32
uodated_at	20170604	Int32
photo_url	/foto.jpg	String
attributes	{ 5 fields }	Object
strength	80	Int32
agility	80	Int32
intelligence	80	Int32
charisma	80	Int32
will	80	Int32
abilities	[3 elements]	Array
[0]	{ 6 fields }	Object
name	mathematics	String
level	A+	String
time_level	36000	Int32
max_level	5	String
time_max_level	3450	Int32
experience	5000	Int32
[1]	{ 6 fields }	Object
[2]	{ 6 fields }	Object

Figura: Representació de la colecció userprofile

6. Implantació

6.1. Operacions necessàries per passar a entorn de producció

- Exportar el codi del connector des de l'entorn de desenvolupament a producció:

```
/elgg/mods/kpax2_profile
```

- Exportar el codi Node.js des de l'entorn de desenvolupament a entorn de producció
- Reiniciar el servidor Node.js
- Activar el connector kpax2_profile

6.2. Punts importants a tenir en compte

- El connector kpax2_profile te que esta carregat l'últim o després de profile i kpax2_core

7. Manteniment

La fase de manteniment consistirà no només a corregir els errors trobats, sinó a avaluar les millores que aquesta nova funcionalitat ha aportat a la plataforma KPAX2, principalment pel que fa a la usabilitat i disseny.

Hauríem de considerar els següents criteris a l'hora d'obtenir unes valoracions més altes en aquest aspecte en les noves funcionalitats a implementar:

- L'aplicació ha de ser fàcilment mantenible, cal que les següents funcionalitats no introdueixin més complexitat per tal de mantenir la facilitat per fer diagnòstic i correcció d'errors
- L'aplicació ha de tenir facilitat per ser refinada o modificada per nous requeriments dels usuaris
- El manteniment ha de ser preventiu, els canvis s'han d'orientar en l'augment de la fiabilitat i a evitar possibles errors futurs
- L'aplicació ha de ser adaptable per facilitar els canvis en l'ambient de programari. La plataforma KPAX2 és adaptable, ja que està força separada la lògica de negoci de la presentació, per tant, es podria canviar aquesta presentació fàcilment per una aplicació en Android, per exemple, o canviar la lògica de negoci a altres llenguatges sense canviar el connector (sempre mantenint els punts d'accés de l'API)

8. Conclusions

Les meves conclusions les podria valorar en dos àmbits, la capa visual amb ELGG i PHP i a servidor amb node.js i MongoDB.

Per la part de la capa visual, haver optat per un sistema de xarxes socials ja implementat com ELGG i orientat a connectors dona molta facilitat a l'hora d'estendre les funcionalitats de la xarxa KPAX2 però a l'hora, la corba d'aprenentatge resulta massa elevada.

Per altra banda, decidir que els serveis i la lògica de negoci estigi en mongoDB i node.js m'ha semblat un molt bon encert, molt fàcil d'implementar i sobretot d'executar, en comparació amb Java, que necessita un servidor d'aplicacions molt més robust.

9. Referències bibliogràfiques i de recursos web

[1] KPAX2: Experiences on designing a gamified platform for seriousgaming

https://www.researchgate.net/publication/304108277_Chapter_13_KPAX

[2] ELGG. Open Source Social Network Engine.

<http://elgg.org/>

[3] Restful RFC

<https://tools.ietf.org/html/rfc6690>

[4] Node.js

<https://nodejs.org/es/>

[5] MongoDB

<https://www.mongodb.com/es>

[6] PHP

<http://php.net/>

[7] GPL

<https://www.gnu.org/licenses/licenses.es.html>