

Disseny i desenvolupament d'un mòdul per a la integració del middleware SOS

León Font, Enric

Curs 2013-2014

Director: Davinia Hernández-Leo

GRAU EN ENGINYERIA EN INFORMÀTICA



Universitat
Pompeu Fabra
Barcelona

Escola
Superior Politècnica

Treball de Fi de Grau

Disseny i desenvolupament d'un mòdul per a la integració del middleware SOS amb Moodle

Enric León Font

Treball Final de Grau / 2014

TUTORS
Davinia Hernández-Leo DEPARTAMENT
GTI



Abstract

Currently, when activities take place in classroom physical spaces, the teacher is usually the figure in charge of managing the creation of groups of students and the tasks that the groups or students have to complete. This control implies an additional workload for the teacher, and it can increase the difficulty of the activity to be performed.

There are multiple alternatives for automating everything that is included in the process of creation and monitoring of groups and tasks during learning activities. One example is found in the Signal Orchestration System (SOS). It is a system that consists of multiple devices and software that enables management of these activities, and it has the aim of letting the student easily identify which group they belong to in each moment and, at the same time, allows the teacher to manage the groups of students in a more automated manner depending on different parameters.

These systems are focused on the management and creation of groups of students for activities in physical spaces, but there are occasions where they could be complemented by other activities in virtual spaces like, for example, Learning Management Systems (LMSs) (Moodle, Blackboard, etc). This way it would be possible to better organize the time students dedicate to the tasks depending on whether the activities are more individual-focused or more interactive-focused. This would allow the former to be completed outside of class hours leaving class hours free for the group activities that need a more dynamic organization even while keeping a dependency relationship and continuous follow up between them both in the physical and the virtual domains.

This project attempts to integrate physical spaces activities organized for orchestration systems like SOS with purely virtual spaces like Moodle.

Resum

Actualment, en la realització d'activitats en espais físics que tenen lloc a les aules, el professor és habitualment la figura que s'encarrega de gestionar la formació de grups d'alumnes i les tasques que els grups o alumnes han de portar a terme. Aquest control implica una càrrega de feina addicional per part del professor, i pot arribar a incrementar la dificultat de l'activitat que s'ha de realitzar.

Hi ha diverses alternatives per tal d'automatitzar tot el que inclou el procés de formació i monitorització de grups i tasques en les activitats d'aprenentatge. Un exemple el podem trobar en el Signal Orchestration System (SOS). Es tracta d'un sistema que consta de múltiples dispositius i un software que en permet la gestió, i que té com a objectiu que l'alumne identifiqui fàcilment quin és el grup al que pertany en tot moment. A la vegada també permet al professor gestionar els grups d'alumnes d'una forma més automàtica en funció de diferents paràmetres.

Aquests sistemes estan centrats en la gestió i formació de grups d'alumnes per a activitats en espais físics, però hi ha ocasions on es podrien complementar aquestes activitats amb d'altres en espais virtuals com per exemple Learning Management Systems (LMSs) (Moodle, Blackboard, etc). D'aquesta manera es podria organitzar millor el temps de dedicació dels alumnes a les tasques que han de fer segons si es tracta d'activitats de caràcter més individual o activitats de caire més interactiu. Això permetria realitzar les primeres fora de l'aula i deixar aquesta lliure per a les activitats grupals o que requereixin una organització més dinàmica, tot mantenint una relació de dependència i monitorització continua tan en el domini físic com en el virtual.

Aquest projecte pretén integrar aquest tipus d'activitats en espais físics organitzades per a sistemes de orquestració com el SOS amb espais purament virtuals com Moodle.

Índex

Índex de figures	vii
Índex de taules	ix
Acrònims	x
1 INTRODUCCIÓ	1
1.1 Context	1
1.2 Objectius	2
1.3 Metodologia i planificació	2
2 TECNOLOGIES INVOLUCRADES	7
2.1 Learning Management Systems	7
2.2 Moodle	9
2.2.1 Entorn	10
2.2.2 Versió	10
2.3 Tecnologies per a entorns d'aprenentatge físics	11
2.3.1 Audience response systems (Clickers)	12
2.3.2 Pissarra electrònica interactiva	12
2.4 Signal Orchestration System (SOS)	14
3 ANÀLISI I DISSENY DEL MÒDUL DE MOODLE	19
3.1 Requeriments	19
3.2 Casos d'ús	22
3.3 Disseny de la API	22
4 IMPLEMENTACIÓ I INTEGRACIÓ	31

4.1	Implementació	31
4.1.1	Capa de dades	32
4.1.2	Lògica de negoci	33
5	CONCLUSIONS I PROPOSTES DE MILLORA	39
5.1	Conclusions	39
5.2	Possibles millores i propostes	40
	Bibliografia	43

Índex de figures

1.1	Planificació per al desenvolupament del mòdul de Moodle	4
2.1	Llista dels 10 LMSs més populars a gener del 2014	9
2.2	A la dreta: Noves instal·lacions des del març del 2014. Esquerre: Instal·lacions fins el maig del 2014	11
2.3	Exemple de dispositius de resposta immediata	13
2.4	Nens utilitzant una pissarra electrònica interactiva	13
2.5	Conjunt de dispositius que formen el SOS	14
2.6	Arquitectura simplificada del sistema i l'extensió de Moodle (Sig- nal Orchestration System (SOS) Service)	15
3.1	Diagrama de casos d'ús.	21

Índex de taules

3.1	Consulta el llistat de cursos	22
3.2	Consulta d'un curs específic	23
3.3	Consulta el llistat de usuaris	23
3.4	Consulta el llistat d'usuaris d'un curs específic	24
3.5	Consulta d'un usuari específic	24
3.6	Consulta el llistat de grups d'un curs específic	25
3.7	Modifica l'organització de grups d'un curs específic	25
3.8	Assigna un recurs a un grup	26
3.9	Esborra totes les assignacions de recursos d'un curs específic	26

Acrònims

SOS Signal Orchestration System	vii
LMS Learning Management System	1
UPF Universitat Pompeu Fabra	1
GTI Grup de Tecnologies Interactives	1
ORM Object-relational mapping	32
SRP Single Responsibility Principle	33

Capítol 1

INTRODUCCIÓ

1.1 Context

Quan parlem de tecnologia a les aules parlem normalment de dispositius pensats per a poder donar suport a la organització de les tasques que es duen a terme dins d'aquestes. Això fa que es pugui treure molt més profit a nivell d'aprenentatge d'aquelles activitats que requereixen un elevat grau de control i monitorització, ja que aquests processos es poden fer de manera automàtica o semi-automàtica.

En general podríem distingir-ne de dos grups bàsics. Aquelles tecnologies que es fan servir dins les aules, en activitats reals, i aquelles que molts cops es fan servir per a docència de tipus no presencial, on no cal la figura del professor. En el primer grup situaríem els clickers (dispositius de resposta immediata), les pissarres digitals interactives i els sistemes com el SOS [1] entre d'altres. En el segon grup hi podríem situar eines de software com els Learning Management Systems (LMSs), o qualsevol tipus de software pensat per a donar suport a la docència.

Els dispositius que es fan servir a les aules, potencien les activitats que s'hi realitzen, però normalment la seva utilització no va més enllà del context on s'emmarca, en aquest cas el context físic de l'aula. Aquest projecte intenta contribuir a resoldre aquest problema en el marc del Grup de Tecnologies Interactives (GTI) de la Universitat Pompeu Fabra (UPF) amb el projecte de investigació EEE (TIN2011-

28308-C03-03).

1.2 Objectius

En concret l'objectiu general d'aquest projecte és el de proposar una solució per tal d'extendre un LMS (Moodle) de forma que es pugui aconseguir una integració efectiva entre les activitats que es porten a terme dins de l'espai d'aprenentatge de Moodle i aquelles activitats col·laboratives que es porten a terme en espais físics utilitzant el SOS. Això vol dir establir una continuïtat entre les activitats a l'aula i les activitats al Moodle.

Els objectius específics del projecte són:

- Analitzar Moodle, ja que és el LMS d'ús més extens de codi lliure.
- Participar en l'equip de desenvolupament del SOS per tal d'extendre la seva interoperabilitat amb altres sistemes.
- Dissenyar i desenvolupar una extensió de Moodle que permeti la integració amb el SOS.
- Portar a terme una prova pilot on es pugui comprovar el correcte funcionament del desenvolupament realitzat juntament amb els altres components del SOS.

1.3 Metodologia i planificació

El gruix principal de feina del projecte consisteix en la realització de l'extensió de Moodle. Per tal de portar a terme aquest desenvolupament és interessant seguir una metodologia que ens permeti certa flexibilitat, ja que no totes estan enfocades a projectes on la dedicació no és la mateixa cada dia, i a la vegada ens hauria de donar certa visió de progrés per tal de saber en quin punt del desenvolupament ens trobem en cada moment.

Com que aquest projecte és part i depèn a la vegada d'un altre projecte en desenvolupament (SOS) és necessari fer reunions setmanals, o si no és possible quinzenals per tal d'anar veient si els requeriments inicials han canviat, si la viabilitat d'algun plantejament que teníem ja no és possible, i en general per portar un seguiment del projecte en termes globals i poder anar avançant de forma que no quedi cap interrogant per resoldre de cara a l'experiment que hi havia planificat a un centre educatiu.

Com que la visibilitat del progrés d'aquest projecte és una peça clau pel seu èxit, s'ha d'escollir una metodologia que permeti obtenir aquesta visió del progrés en un moment donat, i una de les millors maneres d'aconseguir aquesta visió en quant als petits objectius que es van fixant és seguint una metodologia àgil, un concepte que està enfocat sobretot a projectes on hi ha entregues periòdiques constants, de una a quatre setmanes.

Aquest projecte, ha seguit un procés basat en les metodologies àgils, amb revisions setmanals, i el calendari de planificació es pot veure a la figura 1.1.

S'ha estructurat la planificació en quatre petites entregues per tal de poder anar fent una integració continua amb la plataforma SOS. D'aquesta manera si surten problemes durant la integració es poden anar resolent de forma més concreta i la integració final resulta molt més fàcil que no pas si es desenvolupa tot per separat i després s'integra. S'ha dividit en quatre fases ben marcades.

1. Fase 1: Preparació de l'entorn local per al desenvolupament i d'un entorn online per a la integració amb el middleware SOS. Bàsicament es tracta de tots els passos prèvis al desenvolupament que inclouen la instal·lació de PHP i mySql tan en la màquina de desenvolupament com en el servidor online del GTI per tal de fer les proves corresponents amb el SOS middleware.
2. Fase 2: Preparació del plugin i primera versió funcionant. Investigar com s'ha de fer per tal de construir un plugin de Moodle que permeti exposar una sèrie de mètodes com a web service. Quina estructura ha de seguir, com ha d'estar estructurat, quin tipus de add-on ens va bé pel que volem fer, etc.
3. Fase 3: Funcionalitat bàsica. (Que tot tingui una estructura correcta i llistar dades d'usuaris i cursos). Començar a implementar la funcionalitat bàsica

▲ Fase 1	19 dies	lun 23/12	jue 16/01
Instal·lació de l'entorn (PHP MySQL)	1 dia	lun 23/12	lun 23/12
Instal·lació de l'IDE i configuració debug en PHP	1 dia?	vie 27/12	vie 27/12
Obtenir i instal·lar versió actual de moodle	1 dia?	mié 08/01	mié 08/01
Configurar moodle amb dades dummy	1 dia	mié 15/01	mié 15/01
Fase 1 completa	0 dies	jue 16/01	jue 16/01
▲ Fase 2	10 dies	jue 23/01	mié 05/02
Configurar moodle per a serveis web i documentar-ho	3 dies	jue 23/01	lun 27/01
Estructura inicial del plugin	4 dies	jue 30/01	mar 04/02
Fase 2 completa	0 dies	mié 05/02	mié 05/02
▲ Fase 3	36 dies	mié 12/02	mié 02/04
Llista de cursos	3 dies	mié 12/02	vie 14/02
Llista d'usuaris de moodle	2 dies	mié 19/02	jue 20/02
Arquitecturar i refactoritzar codi existent	3 dies	mié 26/02	vie 28/02
Estructura de classes	2 dies	mié 05/03	jue 06/03
Unificació de nomenclatures	1 dia	mié 12/03	mié 12/03
Llista d'usuaris d'un curs	1 dia	mié 19/03	mié 19/03
Llista de grups d'un curs	1 dia	mié 26/03	mié 26/03
Fase 3 completa	0 dies	mié 02/04	mié 02/04
▲ Fase 4	27 dies	lun 07/04	mar 13/05
Empaquetació del plugin	3 dies	lun 07/04	mié 09/04
Automatitzar l'empaquetació del plugin	2 dies	lun 14/04	mar 15/04
Llistat de recursos	4 dies	lun 21/04	jue 24/04
Assignació de grups recursos a groupings	2 dies	vie 25/04	lun 28/04
Restricció de visibilitat de recursos segons groupings	5 dies	mié 07/05	mar 13/05
Fase 4 completa	0 dies	mar 13/05	mar 13/05

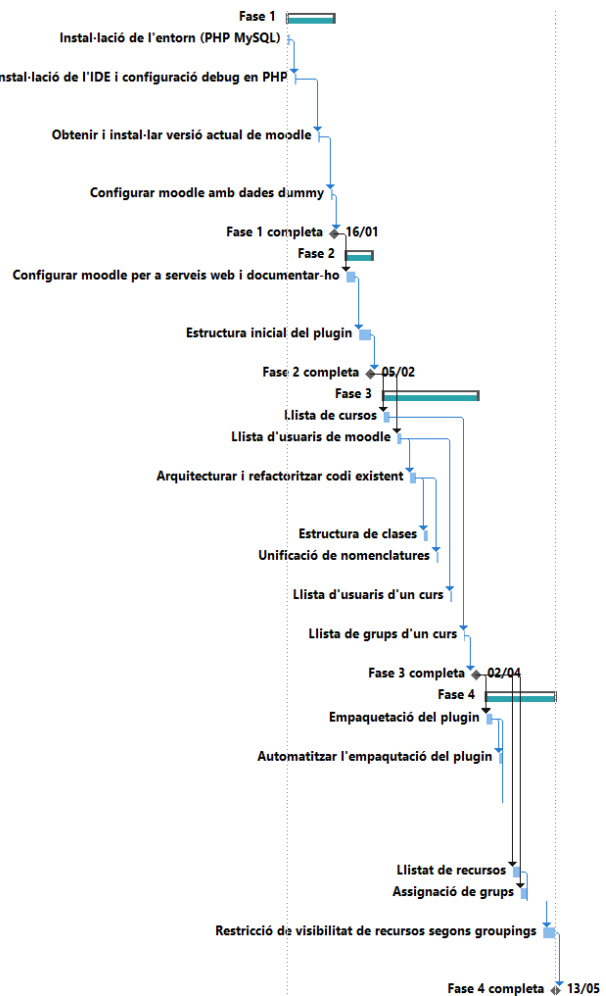


Figura 1.1: Planificació per al desenvolupament del mòdul de Moodle

de consulta de dades a la base de dades de Moodle. Com obtenim els cursos, els usuaris, etc.

4. Fase 4: Automatització de la instal·lació i funcionalitats avançades (Llistat de recursos i escriptura de dades de grups i de recursos) Aquí és on hi haurà no la major part de feina però sí la part més crítica, ja que és aquí on desenvoluparem la funcionalitat relacionada amb els grups i els recursos, que és precisament el que posarem a prova amb una prova pilot que es durà a terme al centre educatiu.

Un cop s'hagi acabat amb el desenvolupament i la integració amb el SOS s'haurà de provar en una sessió de simulació d'una activitat que segueixi un patró d'aprenentatge col·laboratiu com podria ser el jigsaw learning flow [2]. D'aquesta manera es podrà comprovar si tot encaixa bé, analitzar-ne els punts dèbils, i potenciar-ne els punts forts. Així podrem llistar una sèrie de requeriments que ens ajudaran a millorar el sistema de cara a la prova pilot que es durà a terme al centre educatiu.

Capítol 2

TECNOLOGIES INVOLUCRADES

Per tal d'entendre què és el que hi ha darrere d'aquest projecte a nivell conceptual i a nivell tècnic cal entendre també quines són les tecnologies que hi ha darrere i quin paper tenen dins del projecte. És per això que s'ha fet un anàlisi de les diferents tecnologies involucrades. Veurem què és un LMS, per què ens interessa per al nostre projecte i com es relaciona amb l'aplicació que volem fer.

2.1 Learning Management Systems

Els LMSs són eines per a gestionar les activitats d'aprenentatge en espais virtuals, és a dir no presencials. Normalment formen part de l'entorn corporatiu d'escoles, instituts o universitats. La idea que hi ha darrere d'un LMS és la de ser un framework que s'encarregui de totes les etapes del procés d'aprenentatge, i això inclou, entre d'altres, la gestió de les qualificacions dels alumnes, els recursos disponibles per a una assignatura concreta, el les entregues dels estudiants per a les diferents assignatures, etc,...

Els LMSs són força utilitzats per a complementar les activitats d'aprenentatge que tenen lloc a cursos presencials, però també se'n utilitzen per a entorns purament virtuals com MOOCs o plataformes d'e-Learning. Aquest segon concepte per

això, no entra dins l'abast del projecte, ja que ens interessa més la part d'integració d'activitats presencials amb activitats virtuals.

Les característiques que ofereixen els LMSs són prou diverses i engloben diferents àmbits. En general un bon LMS hauria de donar suport a:

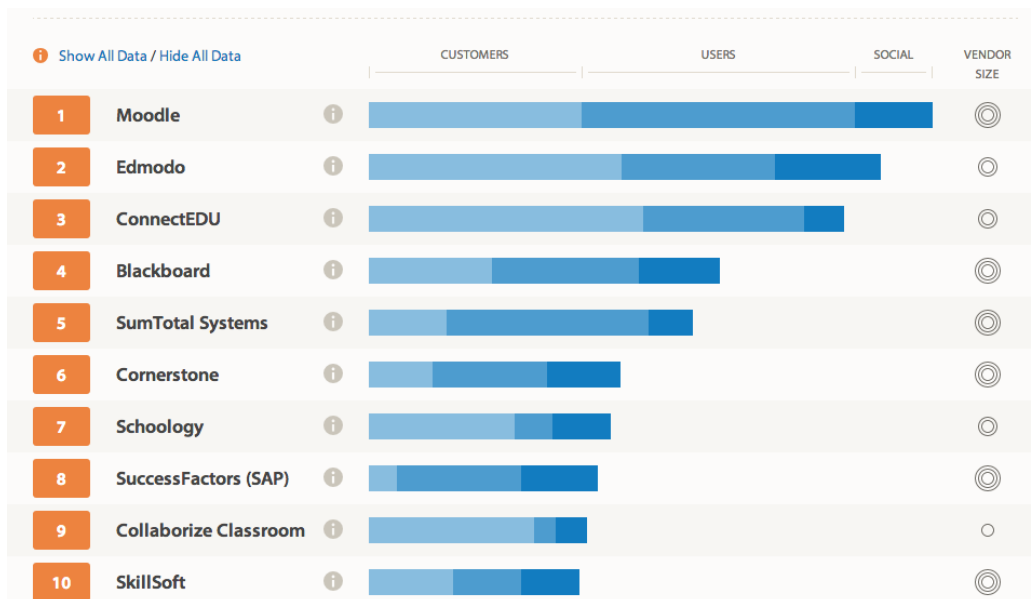
- Distribució de continguts: Els LMSs permeten distribuir continguts als seus usuaris (administradors, professors, alumnes,...) però normalment no estan enfocats a la creació d'aquests.
- Avaluació i seguiment dels alumnes: Han de ser capaços de portar un seguiment efectiu del rendiment acadèmic i de l'acompliment de les tasques assignades als alumnes així com la possibilitat per part dels professors de poder generar informes detallats d'aquestes dades.
- Administració i assignació de recursos: Han de ser capaços també d'administrar els continguts disponibles al sistema i donar la possibilitat de gestionar-ne la visibilitat per a determinats usuaris o grups.

Actualment, la llista de LMSs que hi ha al mercat és força àmplia, però els més importants [3] es poden veure a la figura 2.1

Podem veure, doncs que les opcions son força diverses, però hi ha un tret clau que és el que ens permetrà portar a terme el nostre projecte. És necessari recordar que aquest projecte pretén estendre la funcionalitat dels LMSs per tal de poder-los utilitzar a través d'una interfície que connecti amb sistemes d'orquestració d'activitats d'aprenentatge en espais físics com el SOS per tal de donar una continuïtat en els espais virtuals al seguiment realitzat de el SOS durant les activitats.

Per tal de portar a terme el projecte, doncs és necessari triar un LMS, i aquest hauria de permetre la utilització de plugins o extensions, ja que la nostra aplicació hauria de ser fàcilment empaquetable de forma que la instal·lació sigui fàcil i còmode. Els LMSs que ofereixen aquesta possibilitat són Moodle i Sakai [4].

Moodle esdevé la millor opció, ja que com el seu nom indica està dissenyat de forma modular, tenint en ment que els desenvolupadors que hi col·laboren puguin crear nous mòduls de forma fàcil. També cal destacar que és open source, com Sakai o Ilias i aquesta característica és clau ja que com a bon projecte open source té al darrere una comunitat de desenvolupadors que aporten molta informació que



Font: [3]

Figura 2.1: Llista dels 10 LMSs més populars a gener del 2014

podrem utilitzar si ens trobem amb qualsevol problema. A banda de les raons tècniques o de desenvolupament no s'ha d'oblidar que Moodle és el LMS més estès a nivell de centres educatius, la qual cosa farà que el nostre sistema sigui compatible d'entrada, i pugui arribar a interessar a moltes escoles i instituts que fan servir aquesta eina.

2.2 Moodle

Moodle és un sistema que pretén integrar tots els aspectes necessaris per al suport de processos d'ensenyament-aprenentatge. Es va dissenyar amb un principi bàsic en ment: la modularitat. Havia de ser molt flexible per tal de poder adaptar-se als canvis que es poguessin produir en les maneres d'ensenyar i així poder-los integrar. És una plataforma LMS que no dona més importància al professor que a l'alumne sinó que els té als dos en compte per igual. Està pensat tan com per classes o cursos que es portin a terme únicament a través d'internet, com per a ser

utilitzat de suport per cursos o activitats presencials.

Al darrere hi ha una filosofia basada en el constructivisme social que planteja l'aprenentatge com un procés que no és un simple traspàs de coneixements del llibre a l'individu sinó que és un procés que a més té en compte el context de la persona i els lligams socio-culturals que aquella persona pot tenir. Això es nota en que Moodle està pensat totalment per a que tan alumnes com professors col·laborin amb els seus companys, i d'aquesta manera es generi més coneixement. Aquesta col·laboració es pot fer mitjançant fòrums, activitats col·laboratives, comentaris, etc,...

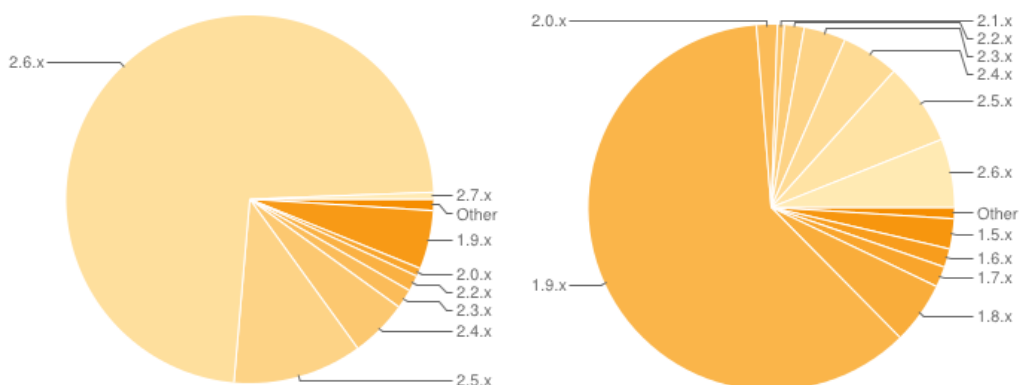
És això precisament el que ens interessa de Moodle, no només el fet que hi hagi molts tipus d'activitat per defecte, sinó que a més es poden crear add-ons personalitzats per fer el que necessitem segons la situació, i això li dóna una riquesa és difícil d'igualar. Per altra banda és open source, i això ens ajudarà molt també a trobar documentació i suport en una comunitat que cada dia és més gran.

2.2.1 Entorn

Moodle és força fàcil d'instal·lar, ja que està programat en PHP això el fa fàcilment portable. L'únic que necessitem serà un servidor amb suport per PHP i un sistema de base de dades suportat per Moodle, en el nostre cas MySql per la seva popularitat i perquè és la opció recomanada per Moodle durant la instal·lació.

2.2.2 Versió

La versió més nova de Moodle és la versió 2.7 (12 de Maig del 2014), tot i que encara no és la més instal·lada. Aquest és un factor important a tenir en compte ja que un dels requeriments de l'aplicació és que sigui fàcilment instal·lable. El tema de la compatibilitat és una mica dispers a Moodle, ja que hi ha força canvis de la API del core de Moodle d'una versió a una altra. En principi però, hi ha dos grups de versions força diferenciades. Les versions fins a la 1.9 i les versions a partir de la 2.X.



Font: [5]

Figura 2.2: A la dreta: Noves instal·lacions des del març del 2014. Esquerra: Instal·lacions fins el maig del 2014

Un altre factor a tenir en compte es la externalització de web services. La versió 2.0 introdueix la possibilitat de definir web services dins de la instància de Moodle utilitzant diferents protocols (XML-RPC, AMF, REST i SOAP), així que clarament si volem portar a terme aquest projecte hauríem de escollir una versió igual o superior a la 2.0, ja que ens serà més fàcil implementar una API amb total compatibilitat amb la gestió de Moodle que faci us dels mètodes de la API interna de Moodle.

Hem de tenir en compte també que a la llarga, les versions anteriors a la 1.9 tindran tendència a desaparèixer, segons les estadístiques de Moodle [5]. Ho podem veure a la figura 2.2. Finalment hem triat la versió 2.6 per a portar a terme el projecte, ja que era la més nova al moment de instal·lar Moodle als nostres servidors, i era raonable decantar-se pel segon grup de versions.

2.3 Tecnologies per a entorns d'aprenentatge físics

Quan parlem de tecnologia per a entorns d'aprenentatge físics ens estem referint bàsicament a tot allò que s'utilitza a les aules per a dinamitzar, potencia o automatitzar les activitats que s'hi porten a terme. Moltes d'aquestes activitats tenen una càrrega afegida d'orquestració, per exemple, activitats d'estil enquesta a tots els

alumnes i valoració de respostes. Aquestes tecnologies, doncs, tenen com a propòsit alleugerar aquesta càrrega i automatitzar la tasca purament d'orquestració per tal de focalitzar l'atenció al procés d'aprenentatge.

Ens referim doncs, a pissarres electròniques, a clickers (sistemes de resposta instantània), ordinadors, els propis dispositius dels estudiants i en general, a tecnologies que augmenten l'espai d'aprenentatge físic [6] per tal de poder realitzar activitats que altrament podrien arribar a portar més càrrega de feina per culpa de l'organització que requereixen que no pas avantatges a nivell d'aprenentatge.

2.3.1 Audience response systems (Clickers)

Els sistemes de resposta immediata, bàsicament són dispositius en mans dels estudiants que envien senyals per radiofreqüència a un altre dispositiu, en aquest cas l'ordinador del professor, per tal de dur a terme diferents activitats, com un preguntes que pot fer el professor. Es poden fer diferents tipus de preguntes com preguntes de múltiple resposta, preguntes individuals, preguntes col·lectives, a grups, etc.

Aquests sistemes ofereixen diversos avantatges, entre els quals, incentiven la participació a les aules, ja que les respostes es poden visualitzen de forma anònima en conjunt, i d'aquesta manera els estudiants no tenen perquè sentir-se jutjats davant dels seus companys. Per altra banda, les respostes sí que son avaluades individualment per a cada estudiant de forma que es pot recompensar a cada estudiant amb certa puntuació en cas de que la pregunta sigui resposta de forma correcte. [7]

2.3.2 Pissarra electrònica interactiva

Les pissarres electròniques són senzillament un dispositiu de visualització on es connecta l'ordinador, com un projector. La única diferència és que a la vegada es pot fer servir com a dispositiu d'entrada, com per exemple amb les pissarres electròniques interactives tàctils, on es pot controlar el dispositiu amb el dit.

Els avantatges respecte una pissarra convencional són diversos. Ofereixen la possibilitat de poder realitzar activitats una mica més complexes i dinàmiques que les



Font: [8]

Figura 2.3: Exemple de dispositius de resposta immediata



Font: Stan Vonog, 2007

Figura 2.4: Nens utilitzant una pissarra electrònica interactiva



Font: [10]

Figura 2.5: Conjunt de dispositius que formen el SOS

que es podrien fer amb l'altre tipus de pissarres. Obre la porta a activitats de caire més col·laboratiu i més creatives, com podem veure a la figura 2.4. També ofereixen la possibilitat als professors de crear continguts més interessants i sobretot ajuden a captar l'atenció dels alumnes mitjançant aquest tipus d'activitats.

Tot i així també hi ha alguns inconvenients, el més rellevant és el fet que el professor ha de dominar la tecnologia i ha d'estar format en aquest tipus de dispositius [9].

2.4 Signal Orchestration System (SOS)

El SOS és un sistema pensat per al desenvolupament de dinàmiques d'aprenentatge col·laboratives dins de l'aula [1], i l'objectiu principal és el d'ajudar al professor a desenvolupar aquestes dinàmiques gestionant tot allò que té a veure amb l'orquestració d'aquestes activitats, és a dir, la formació de grups, la distribució d'activitats, la organització d'àrees de col·laboració, la gestió dels canvis de di-

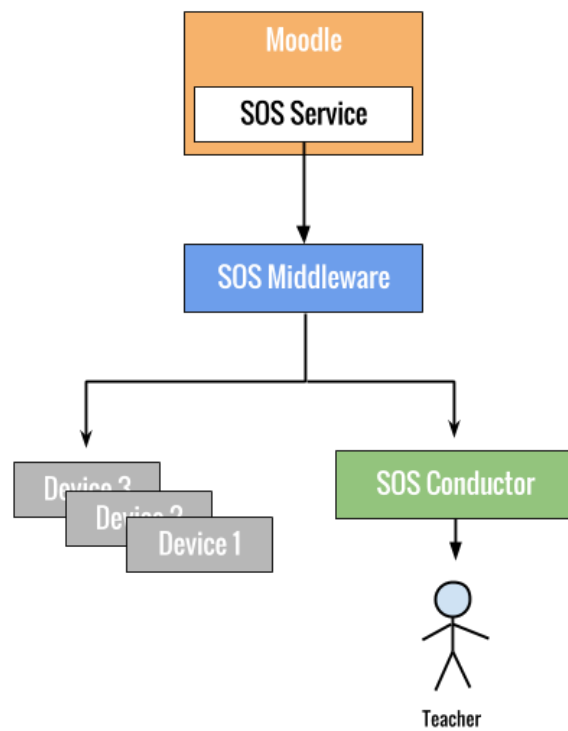


Figura 2.6: Arquitectura simplificada del sistema i l'extensió de Moodle (SOS Service)

nàmiques, etc. A més de gestionar la organització d'aquestes dinàmiques també s'encarrega de mantenir la informació de l'estat actual de l'activitat digitalitzada, i permetre el control d'aquesta informació per part del professor d'una forma còmode i controlada des d'una interfície web.

El sistema porta a terme totes aquestes tasques, augmentant l'espai físic on es porta a terme l'activitat mitjançant una sèrie de dispositius d'entrada i sortida que els alumnes portaran a sobre a mode de braçalet. Com que aquests dispositius són limitats, aquells alumnes que no puguin fer-ne ús es podran descarregar l'aplicació mòbil del SOS amb la que es poden realitzar les mateixes funcions.

Aquests dispositius disposen d'uns leds de senyalització de diferents colors, que conjuntament amb les làmpades d'àrea permeten a l'alumne identificar-se amb una de les làmpades, facilitant la formació de grups per cada color (veure figura 2.5). Els dispositius disposen també de dos botons que permetran la gestió del feedback dels alumnes per part del sistema, donant la oportunitat a aquests d'influir en el desenvolupament de les activitats enriquint l'entorn d'aprenentatge.

A nivell tècnic, la interfície del SOS està formada per dos elements clau que cal analitzar per a veure com es pot integrar amb aquest projecte. Bàsicament podem dir que està format per dos components clau:

- SOS conductor: És la interfície principal des d'on es gestionen les activitats que es poden dur a terme dins de l'aula. És des d'aquí des d'on es poden veure quins són els dispositius que hi ha connectats i quins són els grups que s'han de formar per a cada fase de les activitats.
- SOS middleware: Consisteix bàsicament en una API que s'encarrega d'exposar una sèrie de mètodes per a consultar i escriure tota aquella informació rellevant per el SOS. Tot el que passa al sistema passa pel SOS middleware. És el nexa entre el SOS conductor i les dades.

Tal com està muntada l'arquitectura del sistema, per tal de donar compatibilitat a serveis externs com Moodle, s'ha de modificar el sistema. Es podria crear una sèrie de mòduls dins del que seria el SOS middleware, i cada un d'ells donaria un conjunt de funcionalitats per tal de connectar-se, rebre i escriure dades dels serveis externs. Seguint aquest plantejament, la API del SOS middleware en faria

ús per tal de connectar amb el plugin de Moodle (SOS Service) i gestionar-ne les dades.

La estructura de tot plegat es mostra a la figura 2.6

Capítol 3

ANÀLISI I DISSENY DEL MÒDUL DE MOODLE

3.1 Requeriments

La primera fase en qualsevol projecte és la de presa de requeriments. És en aquesta fase quan s'assenten les bases del que ha de fer i del que no ha de fer l'aplicació/software. Els requeriments són descripcions concretes d'aquelles accions que es vol que el sistema pugui permetre. S'ha de tenir en compte que aquestes descripcions poden contenir conceptes ambigus, ja que les connotacions de les paraules no són iguals per tothom, així que és habitual construir un glossari de termes per tal de que tothom identifiqui les paraules amb els mateixos conceptes.

En el nostre cas, els requeriments de l'aplicació venien donats pel client, en aquest cas l'equip del GTI de tecnologies per l'aprenentatge, tot i que alguns d'ells anaven apareixent sobre la marxa en funció de les possibilitats que s'analitzaven setmana a setmana.

Així doncs, com que passava que els requeriments podien anar canviant o evolucionant a mesura que el desenvolupament, una opció raonable era dividir el desenvolupament en diferents fases, i a cada fase portar a terme el procés de presa de requeriments (bàsicament descobrir els nous i actualitzar els que no s'havien realitzat), anàlisi del sistema, disseny de les funcionalitats, implementació i proves

en un entorn online.

Una de les assumpcions que es van fer abans de la presa dels requeriments va ser el fet de assumir que el plugin s'havia de desenvolupar per a Moodle i que havia de ser un plugin. La primera idea que ens va venir al cap quan vam pensar en com integrar Moodle amb el SOS va ser la d'utilitzar directament l'API de Moodle des del SOS Middleware. Els avantatges que aportava aquest enfocament era que molts dels mètodes que necessitaríem ja estaven implementats. Per altra banda, canvia de interfície cada poques versions i és molt plana en el sentit que els mètodes més complexos de formació de grups i assignació de recursos no hi estaven disponibles. És per això que es va decidir implementar un plugin que inclogués totes aquestes funcionalitats, ja que d'aquesta manera quedava separat el "paquet" que feia que la instància de Moodle quedés plenament integrada amb el SOS del Moodle en si. Gràcies a això també podríem incloure molta lògica de negoci que només té relació amb Moodle dins del propi plugin, i no haver de tractar amb informació interna de Moodle en el SOS Middleware, el qual no té la responsabilitat d'aquestes dades. Tenint això clar, els requeriments que es van extreure al llarg del projecte són els següents:

1. L'aplicació ha de ser instal·lable per a qualsevol usuari amb drets de administrador.
2. L'aplicació ha de portar a terme automàticament durant la instal·lació aquelles configuracions que un usuari no experimentat (professor) no sabia portar a terme.
3. L'aplicació ha de permetre consultar la llista de cursos de Moodle.
4. L'aplicació ha de permetre consultar la llista d'usuaris de Moodle.
5. L'aplicació ha de permetre consultar la llista d'usuaris que estiguin matriculats a un curs concret de Moodle.
6. L'aplicació no tindrà cap mena d'interfície gràfica. La interfície a la qual s'accedirà serà una API.
7. S'ha de poder autenticar les crides a la API de l'Aplicació de manera que si no s'ha autoritzat l'accés prèviament no s'hi pugui accedir.

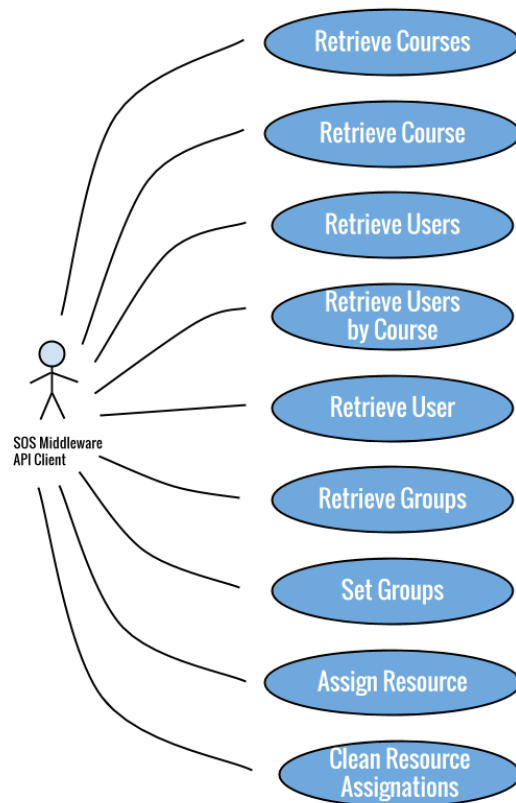


Figura 3.1: Diagrama de casos d'ús.

8. L'aplicació ha de permetre la creació / modificació de grups d'usuaris per a cursos específics.

9. L'aplicació ha de permetre consultar la llista de recursos disponibles a un curs específic de Moodle.

10. L'aplicació ha de permetre assignar els recursos disponibles d'un curs específic de Moodle a un grup o usuaris concrets de manera que la resta de grups o usuaris no puguin veure llistat el recurs en qüestió.

Descripció	Retorna el llistat de cursos que hi ha registrats a la instància del Moodle
Assumpcions	S'ha aconseguit un token vàlid a través de la interfície de Moodle
Passos	1. Cridar el mètode <code>sos_course_get</code>
Comentaris	La id de cada objecte curs es refereix a la id interna que aquell curs té dins de Moodle

Taula 3.1: Consulta el llistat de cursos

3.2 Casos d'ús

Aquests requeriments es poden veure plasmats en la llista dels següents casos d'ús. En el nostre cas, com que l'actor principal és un client de la API que volem fer, i com que aquesta API no té estat, no hi ha una sèrie de passos previs per tal de realitzar una acció dins del sistema, ja que cada acció és atòmica en el sentit que no depèn d'altres accions per tal de ser portada a terme. Seria diferent en el cas de plantejar l'escenari complet d'ús del sistema, on l'actor principal desenvolupa les accions que faran que el SOS Middleware anés a demanar la informació pertinent al SOS Service.

Podem veure el diagrama de casos d'ús a la figura 3.1 i les fitxes de cada cas d'ús a les taules 3.1 fins la 3.9.

3.3 Disseny de la API

Un cop tenim els casos d'ús, és força directe passar-ho al que seria el disseny de l'aplicació en si, ja que té sentit que cada un d'aquests casos d'ús es relacionin directament en un mètode de la API per a cada entitat del domini de Moodle. Així doncs, en el domini de Moodle trobem: **User**, **Course**, **Group** i **Resource**.

La principal restricció, però és de nomenclatures, ja que a l'hora de definir un web

Descripció	Retorna la estructura d'un curs en concret.
Assumpcions	<ul style="list-style-type: none"> • S'ha aconseguit un token vàlid a través de la interfície de Moodle • Es coneix la id del curs en qüestió.
Passos	1. Cridar el mètode sos_course_get_by_id .
Comentaris	La id de cada objecte curs es refereix a la id interna que aquell curs té dins de Moodle

Taula 3.2: Consulta d'un curs específic

Descripció	Retorna el llistat d'usuaris que hi ha registrats a la instància del Moodle
Assumpcions	S'ha aconseguit un token vàlid a través de la interfície de Moodle
Passos	1. Cridar el mètode sos_user_get per a saber quins usuaris hi ha.
Comentaris	La id de cada objecte usuari es refereix a la id interna que aquell usuari té dins de Moodle

Taula 3.3: Consulta el llistat de usuaris

Descripció	Retorna el llistat d'usuaris que estan matriculats a un curs específic
Assumpcions	<ul style="list-style-type: none"> • S'ha aconseguit un token vàlid a través de la interfície de Moodle • Es coneix la id del curs en qüestió.
Passos	1. Cridar el mètode sos_user_get_by_course per a saber quins usuaris hi ha registrats al curs en qüestió.
Comentaris	La id de cada objecte usuari es refereix a la id interna que aquell usuari té dins de Moodle

Taula 3.4: Consulta el llistat d'usuaris d'un curs específic

Descripció	Retorna l'objecte usuari donada la seva id
Assumpcions	<ul style="list-style-type: none"> • S'ha aconseguit un token vàlid a través de la interfície de Moodle • Es coneix la id de l'usuari en qüestió.
Passos	1. Cridar el mètode sos_user_get_by_id per a obtenir l'objecte usuari.
Comentaris	La id de cada objecte usuari es refereix a la id interna que aquell usuari té dins de Moodle

Taula 3.5: Consulta d'un usuari específic

Descripció	Retorna el llistat de grups que formen els usuaris d'un curs específic
Assumpcions	<ul style="list-style-type: none"> • S'ha aconseguit un token vàlid a través de la interfície de Moodle • Es coneix la id del curs en qüestió.
Passos	1. Cridar el mètode sos_group_get_by_course per a obtenir el llistat de grups amb les ids dels usuaris que els formen.
Comentaris	El llistat de grups no incorpora la id del grup, sinó el nom, que aquest és únic per a cada curs.

Taula 3.6: Consulta el llistat de grups d'un curs específic

Descripció	Modifica l'organització dels grups d'un curs específic.
Assumpcions	<ul style="list-style-type: none"> • S'ha aconseguit un token vàlid a través de la interfície de Moodle • Es coneix la id del curs en qüestió.
Passos	1. Cridar el mètode sos_group_set_by_course
Comentaris	Aquest mètode accepta com a paràmetres un llistat de grups. El que farà és actualitzar els grups que hi ha dins del curs de Moodle per que siguin idèntics al llistat de grups que es passa per paràmetre.

Taula 3.7: Modifica l'organització de grups d'un curs específic

Descripció	Assigna un recurs donada la id del recurs a un grup donat la id del curs i la id i tipus del grup en qüestió.
Assumpcions	<ul style="list-style-type: none"> • S'ha aconseguit un token vàlid a través de la interfície de Moodle • Es coneix la id del curs en qüestió. • Es coneix la id del recurs en qüestió. • Es coneix la id i tipus del grup en qüestió.
Passos	1. Cridar el mètode sos_resource_assign_group_by_course
Comentaris	N/A

Taula 3.8: Assigna un recurs a un grup

Descripció	Esborra totes les relacions que s'haurien fet prèviament entre els recursos d'un curs específic i els seus grups.
Assumpcions	<ul style="list-style-type: none"> • S'ha aconseguit un token vàlid a través de la interfície de Moodle • Es coneix la id del curs en qüestió.
Passos	1. Cridar el mètode sos_resource_clean_assign_by_course
Comentaris	N/A

Taula 3.9: Esborra totes les assignacions de recursos d'un curs específic

service per un plugin de Moodle s'ha de seguir una nomenclatura concreta [11]. Així doncs, les principals restriccions a l'hora de dissenyar l'arquitectura lògica són les següents.

- Les classes del projecte han d'anomenar-se **<plugin_prefix>_name**
- El desenvolupador no té control sobre el nom dels mètodes de la API. Tenint en compte que aquestes classes tindran uns mètodes anomenats **<function_name>**, els mètodes de la API s'anomenaran **classname_<function_name>**

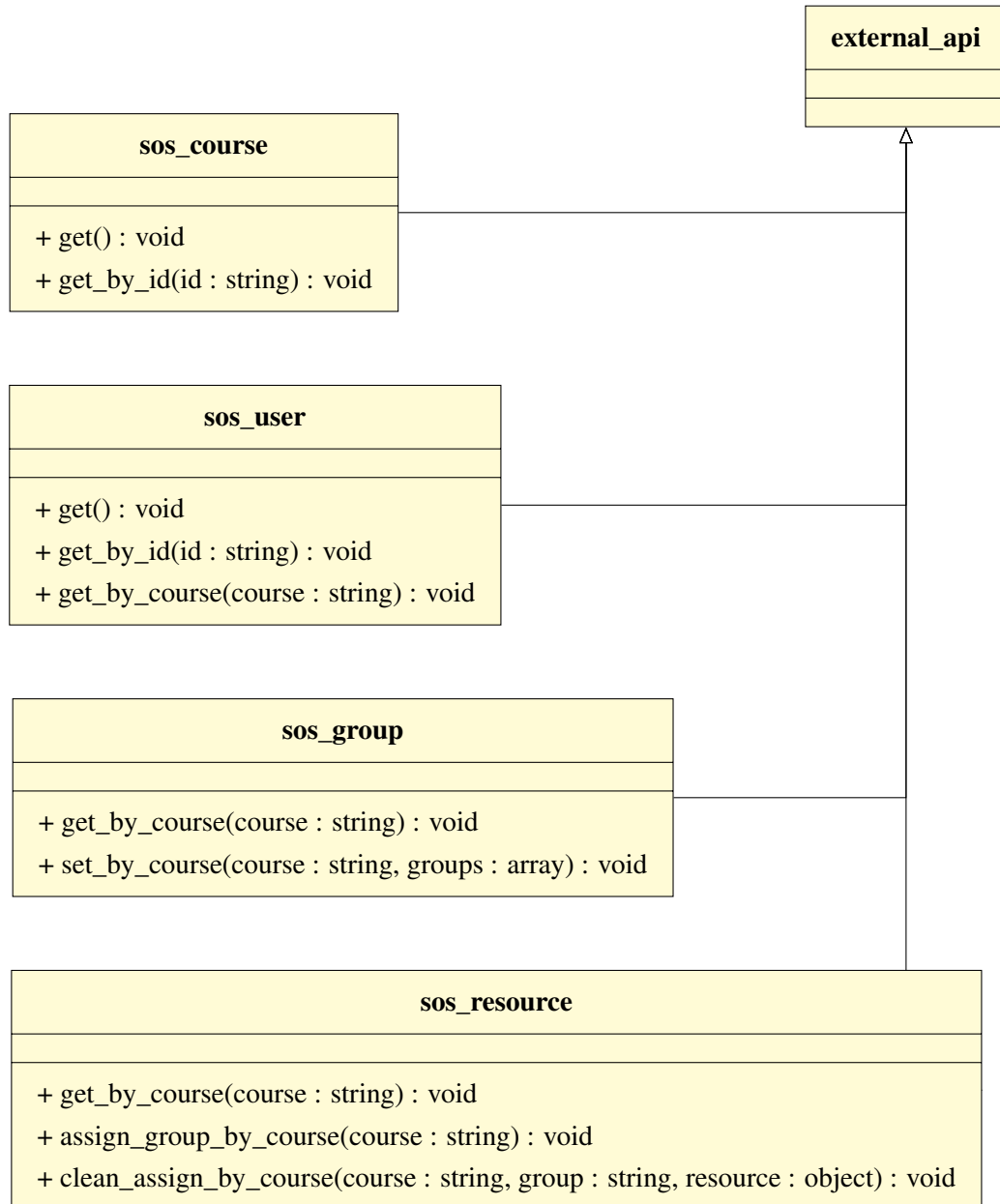
Així doncs, decidim que el **<plugin_prefix>** serà *sos*. Les classes seran les següents.

Finalment, per fer les crides a la api, si el que volem és consultar la llista d'usuaris haurem de fer una crida a **sos_user_get**. Pot semblar una mica estrany aquest tipus de nomenclatura, però el fet que un plugin web service de Moodle posi com a prefix de cada mètode de la API el nom de la classe on es troba aquell mètode força a que els mètodes de cada classe no portin cap prefix de l'entitat a la que fan referència.

Normalment, el que seria lògic seria seguir uns estàndards de nomenclatura per a APIs RESTful, que ens permetessin vincular cada recurs amb les accions Http de cada mètode [12], així, per al nostre cas ens quedaria una cosa de l'estil:

- GET courses HTTP/1.1
- GET courses/1 HTTP/1.1
- GET users HTTP/1.1
- GET users/1 HTTP/1.1
- GET users?course_id=1 HTTP/1.1
- etc.

Sembla però, que hi ha una confusió en el que seria la definició de protocols de web services dins de Moodle, ja que el nostre plugin està categoritzat dins del que seria web services REST dins de Moodle, però en canvi les restriccions que té Moodle envers les nomenclatures fan que la nomenclatura dels mètodes de la API



no s'assemblin en absolut al que seria una API RESTful. Tot i així ens hi podem adaptar.

Com que el que ens interessa és que els mètodes de la API siguin el més explicatius possibles, sembla raonable que els mètodes de cada classe no portin el nom de l'entitat (grup, curs, usuari i recurs) dins del nom del mètode. Això faria que els mètodes de la API fossin massa llargs i poc entenibles.

Capítol 4

IMPLEMENTACIÓ I INTEGRACIÓ

En aquest capítol es detallen els problemes que han anat apareixent durant la fase de desenvolupament, les possibles solucions i la implementació de la solució més adequada per aquests problemes. La dificultat principal que ha aparegut durant la fase de implementació ha estat la falta de claredat de la documentació de la API interna de Moodle, document de referència principal per a desenvolupar un plugin de Moodle.

4.1 Implementació

Per tal de implementar cada un dels mètodes de la API calia entendre molt bé com s'estructura internament el Moodle, quines són les llibreries que hem d'utilitzar i quines són les vies d'accés a la capa de dades de Moodle. També cal entendre quina estructura té la base de dades de Moodle i quines relacions hi ha.

4.1.1 Capa de dades

La base de dades de Moodle segueix una nomenclatura força bàsica. Podem veure que totes les taules de la base de dades de Moodle tenen com a prefix *mdl* seguit del nom de l'entitat que referencien i de les possibles entitats vinculades a cada una d'aquestes. Això facilita força la comprensió de l'estructura, de forma que si el que volem és consultar el llistat de *modules* d'un curs en concret, hauríem de consultar la taula *mdl_course_modules*.

Això és així per a totes aquelles entitats que tenen sentit únicament dins d'una altra, així, podem veure que els *modules* d'un curs té sentit que estiguin continguts dins de *mdl_course*, però no seria el cas per exemple de les matriculacions a cursos concrets, que és una entitat global continguda a *mdl_user_enrolments*, que contindria una referència a usuari i a *mdl_enrol*, que a la seva vegada fa referència a *mdl_course*.

Com més ens endinsem en l'estructura de dades de Moodle ens podem adonar de la complexitat amb la que està dissenyada i la complexitat també de les relacions que hi han contingudes així com de les dades que s'hi emmagatzemen. Caldrà doncs buscar algun tipus de llibreria dins de Moodle que ens abstregui aquestes intricades relacions. Típicament aquesta és la feina de un Object-relational mapping (ORM) [13]. Analitzant l'estructura de fitxers de Moodle veiem que a l'arrel d'una instància de Moodle hi ha carpetes per cada tipus de modul que aquest inclou. Per norma general, dins de cada una d'aquestes carpetes hi podem trobar un fitxer *lib.php*, on hi haurà els mètodes que ens permetran crear, esborrar, i fer operacions diverses per cada un dels mòduls. Aquests mètodes, al seu torn utilitzen la llibreria que fa de ORM. Aquesta llibreria conté tots aquells mètodes per tal de consultar dades a la base de dades de forma segura [14].

Així doncs, en general, si el que volem és fer consultes d'un tipus concret de mòdul el més simple és utilitzar els mètodes disponibles a aquesta llibreria.

```
<?php
    /* The list of courses in the moodle instance. */
    public static function get() {
        global $CFG, $DB;
        require_once("$CFG->dirroot/course/lib.php");
```

```
$courses = $DB->get_records_select('course',  
                                   "format_<>_'site'");
```

4.1.2 Lògica de negoci

En quant a la funcionalitat dels mètodes en sí s'ha intentat mantenir sempre un compromís entre la facilitat d'ús de la API pels clients que l'utilitzaran, en aquest cas el SOS middleware, però a la vegada intentant que la lògica interna de Moodle no afectés al codi del client, és a dir, mantenir el Single Responsibility Principle (SRP) [15]. Aquest principi sosté que una classe ha de tenir una i només una raó per canviar. Això bàsicament vol dir que no s'han de donar responsabilitats de lògica de negoci a la capa de presentació, o responsabilitats de accés a dades a la capa de lògica de negoci.

Aquest principi s'ha tingut en compte des del punt de vista que no s'han de donar responsabilitats sobre el tipus de dades i la lògica interna de Moodle al client de la API, ja que aquest hauria de ser totalment agnòstic al que passi dins de la API. En relació a aquest principi s'ha creat un esquema de dades al SOS middleware per tal de integrar el plugin de Moodle que no té perquè tenir cap relació directe amb les dades internes de Moodle, tot i que per facilitat hi ha molts camps que coincideixen.

Així doncs queda establert que el SOS middleware s'encarregarà de la responsabilitat de presentar aquestes dades a través de la API de la forma que li convingui, i el plugin de Moodle transformarà l'esquema de dades de Moodle al format més agnòstic possible per tal de no mantenir dependències fortes.

```
course: {  
    id,  
    fullname,  
    shortname,  
    summary,  
    timemodified  
}
```

```

user: {
  id,
  firstname,
  lastname,
  email,
  role
}

resource: {
  name,
  description,
  internal_id,
  internal_type
}

group: {
  id,
  courseid,
  name,
  description,
  users: [{
    userid,
    timeadded
  }]
}

```

Tenint aquesta estructura fixada, la major part de feina ha estat la d'adaptar l'estructura interna de Moodle al format de sortida, ja que en alguns casos hi havia moltes transformacions que no eren gaire directes. Ha estat sobretot el cas de **sos_group_set_by_course** i de **sos_resource_assign_group_by_course**.

Modificació dels grups d'un curs de Moodle

Un dels principals requeriments del sistema era el fet de que es poguessin fixar els grups d'un curs de Moodle a través del plugin de forma que (ampliant el context) el professor que fes servir el SOS Conductor, un cop decidits els grups que

realitzen la activitat a l'aula, pogués sincronitzar els grups de l'activitat amb els grups de Moodle. Això permetria poder representar l'estat de l'activitat al Moodle i d'aquesta forma poder restringir certs recursos a grups concrets.

Per tal de fer això, i tenint en ment la facilitat d'ús de la API per part del client (SOS Middleware) es decideix que el mètode `sos_group_set_by_course` accepta com a paràmetre una representació dels grups i usuaris que després el plugin hauria de replicar, a més de la id del curs en qüestió.

```
groups: [{
    name,
    users: [id]
}]
```

En aquest cas, seria força arriscat fer ús de `$DB` per tal de modificar les taules de grups i assignacions d'usuaris d'un curs, ja que l'estructura podria ser força complexe, així que el més interessant seria indagar en la llibreria de grups `/group/lib.php` per tal de descobrir quins mètodes podrien fer servei. Així doncs, s'utilitzen els mètodes `groups_get_members_by_role`, `groups_add_member`, `groups_remove_member` i `groups_create_group` per tal de modificar els grups d'un curs de moodle de forma que reflexi els grups passats per paràmetre. Un altre enfoc hagués estat esborrar tots els grups d'un curs i crear-los de nou, però si es fes així es perdrien les ids internes dels grups, i en el cas que s'estiguessin fent servir per altres motius podria haver-hi conflictes.

Un factor a tenir en compte ha estat també l'atomicitat d'aquestes accions en el sentit que no es poden deixar a mitges. És per això que s'ha fet ús del mètode `$transaction->start_delegated_transaction()` i `$transaction->allow_commit()` que assegura el fet de que si per qualsevol motiu alguna de les operacions amb la base de dades s'interromp es farà un `rollback` de tot allò que s'hagués començat a fer just després de la primera instrucció.

Assignació d'un recurs a un grup d'un curs determinat

Per tal de assignar un recurs concret a un grup d'usuaris, s'ha analitzat l'estructura de *resource*, *group* i *grouping*. Actualment, a la versió 2.6 de Moodle, l'assignació

de recursos a grups està suportada de forma experimental. Es pot configurar a l'apartat de configuració de la instància de Moodle el fet de permetre aquesta assignació.

En aquest punt és on apareixen els *groupings*. Aquesta entitat fa la funció d'agrupador de grups. És possible que es doni el cas on a un professor d'una assignatura li pugui interessar crear configuracions de grups. Aquestes configuracions podrien tenir sentit si es volen portar a terme diferents activitats i per cada una d'elles els grups hagin de ser diferents. Com que aquest és el cas de les activitats que es porten a terme mitjançant el SOS Middleware.

Una de les restriccions envers aquest tema és el fet que un recurs només pot ser assignat a una configuració de grups i no a un grup determinat. La estratègia que s'ha seguit en aquest cas és la de crear un *grouping* que fes la funció de "wrapper" per a un grup determinat. És a dir, si es vol assignar un recurs concret al *Grup A* el que s'ha de fer és crear un *grouping* *grouping_group_a* i assignar el recurs al *grouping* que s'acaba de crear.

```
<?php
    if (!$grouping) {
        // Insert new grouping for the group parameter
        $new_grouping = new stdClass();
        $new_grouping->courseid = $course->id;
        $new_grouping->name = 'grouping_group_' . $group->id;
        $new_grouping->description =
            'This_grouping_is_only_to_wrap_the_group_' . $group->name;
        $new_grouping->descriptionformat = '1';

        $lastinsertid = $DB->insert_record('groupings',
            $new_grouping);

        $grouping = $DB->get_record('groupings',
            array('id' => $lastinsertid));
    }
```

Un cop creat el *grouping*, utilitzant l'accés a base de dades que proporciona *\$DB* i les funcions corresponents de la llibreria */course/lib.php* es pot gestionar

l'actualització de les dades corresponents a l'assignació de groupings i recursos i la corresponent actualització de la cache del curs en qüestió per tal que es vegi reflectit aquest canvi immediatament a la web.

Modificació de configuració durant la instal·lació

Un cop desenvolupat el plugin i empaquetat per a la seva instal·lació, un usuari administrador o professor amb drets d'administració haurà d'instal·lar-lo a la instància de Moodle corresponent. Aquesta feina inclou alguns passos que no són gaire fàcils de trobar per a usuaris inexperts i la seva simplificació podria marcar la diferència a l'hora de prendre la decisió d'utilitzar aquesta eina. S'ha fet ús de l'estructura de plugins de Moodle que permet l'execució de scripts un cop acabada l'instal·lació del plugin, i és en aquest punt on ens interessa realitzar aquestes configuracions automàticament.

- Habilitar la opció avançada *web service* (Administration > Site administration > Advanced features)
- Habilitar la opció *REST protocol* dins dels possibles protocols que ofereix el plugin Web Services de Moodle (l'anomenen REST tot i que no segueix les indicacions d'una API RESTful)
- Habilitar la opció *Group Members Only* de Administration > Site administration > Experimental > Settings

```
<?php
    if (empty($CFG->enablewebservices)) {
        set_config('enablewebservices', '1');
    }

    if (empty($CFG->webserviceprotocols)) {
        $webserviceprotocols = array();
    } else {
        $webserviceprotocols = explode(',', $CFG->webserviceprotocols);
    }

    if (!in_array('rest', $webserviceprotocols)) {
```



```
$webserviceprotocols[] = 'rest';
$protocollist = implode(',', $webserviceprotocols);
set_config('webserviceprotocols', $protocollist);
}

if (empty($CFG->enablegroupmembersonly)) {
    set_config('enablegroupmembersonly', '1');
}
```

Un cop instal·lat el plugin aquestes opcions quedaran habilitades i l'única cosa que farà falta serà obtenir un token que permeti autenticar-se com a usuari del web service. Aquest pas, malauradament s'ha de fer manualment. (Administration > Site administration > Plugins > Web services > Manage tokens)

Tot i així cal comentar que per a la prova al centre educatiu aquesta funcionalitat no era necessària ja que la instància de Moodle que es va fer servir va ser la pròpia que el GTI tenia a la seva disposició per tal de fer les proves.

Capítol 5

CONCLUSIONS I PROPOSTES DE MILLORA

Finalment analitzarem els resultats obtinguts en quant als objectius proposats al començament del projecte i les possibles millores que se'n podrien derivar.

5.1 Conclusions

L'abast d'aquest projecte consistia en el desenvolupament d'un plugin de Moodle que fes possible la integració d'aquest amb el SOS, de manera que es poguessin portar a terme activitats d'aprenentatge col·laboratives en l'espai físic (aula) i que després, es pogués traspasar l'estat final d'aquestes activitats l'espai virtual (Moodle). Es va fer un anàlisi de requeriments per acotar més aquesta descripció i marcar els objectius del projecte. Es va analitzar el llistat de requeriments i en base a aquest es va fer un disseny lògic i físic que plasmés tot allò que estava recollit als requeriments.

Un cop acabat el desenvolupament del projecte, s'ha portat a terme una prova pilot on es van provar les noves funcionalitats del SOS, entre elles aquest mateix projecte. La prova consistia en veure l'aplicació real del sistema en el desenvolupament d'una activitat col·laborativa estructurada segons la tècnica d'aprenentatge

Jigsaw [2]. Un cop realitzada l'activitat, els alumnes tenien disponible al Moodle un document que corresponia al grup on havien participat durant l'activitat.

Per la part que li correspon a aquest projecte, aquesta prova va servir per demostrar l'estalvi de temps i feina que suposa pel professor haver de configurar manualment els grups de l'activitat i l'assignació de recursos a cada grup, alhora que va permetre detectar alguns defectes que potser d'altra manera no hagués estat possible veure.

Es pot dir doncs que s'han complert els objectius en bona mesura i que el resultat ha estat satisfactori tenint en compte el resultat de la prova pilot, tot i que també han sorgit moltes propostes de millora arrel de la primera prova. A nivell tècnic cal destacar que al final, el LMS escollit ha permès un desenvolupament còmode en el sentit que el marc de desenvolupament de plugins de Moodle està força marcat i acotat, de manera que el desenvolupador pot deixar de banda alguns detalls de nivell més estructural i centrar-se en el desenvolupament de la funcionalitat. Tot i així també ha imposat moltes restriccions i limitacions ja que aquesta estructura tan marcada no permet molta llibertat a l'hora de prendre algunes decisions de disseny.

A nivell d'aprenentatge, aquest projecte ha significat una oportunitat de desenvolupar-se i aprendre d'un camp poc conegut per l'autor, tan a nivell tècnic com a nivell personal.

5.2 Possibles millores i propostes

Ha quedat fora de l'abast d'aquest projecte la integració amb altres LMS, propostes també molt interessants i innovadores en el camp de la tecnologia per a la educació. Una de les propostes de millora seria la creació d'una solució més genèrica que unificués la integració amb diferents LMSs per tal de que aquesta sigui totalment transparent per el SOS middleware.

En segon terme queden pendents de millora alguns temes tècnics, com l'automatització d'algunes tasques de instal·lació del plugin o la de trobar un sistema de nomenclatures per a la API del plugin, més conforme al que seria una API

RESTful.

També queda pendent de millora alguns aspectes que poden dificultar el desenvolupament de les activitats. Una de les millores que pot ser necessària és el fet de poder amagar recursos des de la API, ja que d'aquesta manera si hi ha massa recursos a un curs i només es vol utilitzar per a l'activitat un grup reduït d'aquests, els alumnes no haurien de veure els recursos que no són de l'activitat quan visualitzen el curs del Moodle.

Bibliografía

- [1] Hernández-Leo D, Nieves R, Arroyo E, Rosales A, Melero J, Blat J. SOS: Orchestrating collaborative activities across digital and physical spaces using wearable signaling devices. *Journal of Universal Computer Science*. 2012;18(15):2165–2186. Available from: <http://www.scopus.com/inward/record.url?eid=2-s2.0-84871290680&partnerID=40&md5=3d2be030900e1f5f0f76c57638927049>.
- [2] Aronson E. The jigsaw classroom. *PsycINFO*. 1978; Available from: <http://psycnet.apa.org/psycinfo/1980-51351-000>.
- [3] Barrish J. Top Learning Management System Software Products; 2014. Available from: <http://www.capterra.com/learning-management-system-software/\#infographic> [cited 27/05/14].
- [4] Medved J. The Top 8 Free/Open Source LMSs; 2013. Available from: <http://blog.capterra.com/top-8-freeopen-source-lmss/> [cited 27/05/14].
- [5] Moodle. Moodle.org: Moodle Statistics; 2014. Available from: <https://moodle.org/stats/> [cited 29/05/14].
- [6] Santos P, Hernández-Leo D, Blat J. Routes of Geolocated Questions in Formal and Informal Learning Contexts. *Bulletin of the Technical Committee on Learning Technology*. 2012;14:21–24. Available from: <http://www.ieeetclt.org/issues/october2012/Santos.pdf>.

- [7] Kay R, Knaack L. Exploring the use of audience response systems in secondary school Science classrooms. *Journal of Science Education and Technology*. 2009;18:382–392.
- [8] Provini C. *Education World: Using Clickers in the Classroom*; 2012. Available from: http://www.educationworld.com/a_tech/are-clickers-effective-teaching-tools.shtml [cited 06/06/14].
- [9] SMART Technologies Inc . *Interactive Whiteboards and Learning*. 2006;.
- [10] Nieves R. *Diseño e implementación de una aplicación para la orquestación de actividades de aprendizaje colaborativo mediante el sistema de señalización SOS*. Barcelona; 2013.
- [11] Moodle. *Adding a web service to a plugin*; 2010. Available from: http://docs.moodle.org/dev/Adding_a_web_service_to_a_plugin.
- [12] Rodriguez A. *Restful web services: The basics*. Online article in IBM DeveloperWorks Technical Library. 2008;p. 1–11. Available from: <http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:RESTful+Web+services:+The+basics#0>.
- [13] Bisbé RL. *¿Qué es un ORM y por qué nos interesa?*; 2011. Available from: <http://rlbisbe.net/2011/12/13/que-es-un-orm-y-por-que-nos-interesa/#comments> [cited 10/06/2014].
- [14] Moodle. *Data manipulation API - MoodleDocs*; 2014. Available from: http://docs.moodle.org/dev/Data_manipulation_API [cited 10/06/14].
- [15] Atwood J. *Curly's Law: Do One Thing*; 2007. Available from: <http://blog.codinghorror.com/curlys-law-do-one-thing/>.