



## TIMETABLE BLOCK A timetable block for Moodle

**TITULACIÓ:** Enginyeria Tècnica en Informàtica de Gestió

**AUTOR:** Joan Maria Talam Espelta

**Data:** Abril 2009

### Resum

Moodle és un servidor de continguts encarat als llocs o centres educatius, i està ampliament implementat en multitud de centres.

La majoria d'aquests centres, però, no solament ofereixen continguts on-line sinó que Moodle els serveix com a eina de suport a les seves classes.

Timetable Block, està destinat a ser utilitzat en aquests centres com a eina informativa i de gestió dels grups en que cada curs està dividit i el lloc físic on es fan les classes.

En primer lloc, s'ofereix informació immediata del lloc on estan programades les classes i qualsevol canvi en la programació de les classes es veu reflectit a tots els usuaris que hi tenen accés.

I en segon lloc, serveix per oferir informació addicional als professors sobre els accessos al lloc Moodle si son efectuats amb màquines dins de l'àmbit permès i comprovar l'assistència dels alumnes.

Per tant, en aquest bloc seran configurables tant les hores lectives com les aules de les que disposa el centre o institució per l'elaboració de les classes.

Resumint, aquest bloc ens ofereix les següent funcionalitats:

Una eina simple de gestió dels recursos físics com serien les aules on es donarien les classes. Aquesta gestió implicaria el manegament per cada aula de la seva capacitat com també si es tracta d'una aula amb ordinadors connectats en xarxa del rang d'adreces IP que tindrien les màquines ubicades en aquella aula.

Una eina de gestió en la que els professors poden confeccionar els horaris dels cursos que ells imparteixen. Aquesta gestió implica la gestió dels diferents grups en que es pot dividir un curs, com podrien ser els grups de classes pràctiques i els grups de classes teòriques, així com també l'assignació de l'aula, d'entre les disponibles en el moment de la classe.

Una eina informativa per als professors per poder consultar en tot moment quins són els alumnes de cada grup i desde quines màquines s'han realitzat els accessos durant les hores de classe i així poder portar un control de l'assistència virtual.

I finalment una eina informativa per l'alumnat per poder consultar els horaris i la disponibilitat dels grups en els quals es divideixen els cursos als quals està matriculat.

## Abstract

Moodle is a content management server used for schools and educational centers, and is widely implemented in many centers.

Most of these centers, however, not only offer content online, but Moodle is used as a tool to give support to their classes.

Timetable Block, is intended to be used in these centers as an information and management tool of the groups in which each course can be divided and the physical place where the classes are taken.

First, it provides immediate information on where the classes are scheduled, and any change in the classes scheduling is reflected for all users who have access.

And secondly, can be used to provide additional information to the teachers about Moodle site accesses and if this accesses are made with machines allowed in the class environment allowed and check the students attendance.

Therefore, in this block can be configured hours and classrooms available in the center or institution for the development of classes.

In short, this block offers the following features:

A simple tool for managing the physical resources as the classroom where he would give lessons. This management would manage each of their classroom as well as in the case of a classroom with networked computers, the range of the IP addresses of the machines would be located in the classroom.

A management tool for the teachers to do the timetables of the courses they teach. This management involves the management of different groups that can divide a course, such as groups of practical classes and groups of theoretical classes, as well as the allocation of the classroom, from those available at the time of the class .

An informative tool for teachers to consult at any time who are the students in each group and from which machines have made access during school hours, so we can keep track of virtual attendance.

And finally an informational tool for students to check timetables and availability of the groups that are divided the courses to which it is registered.

## Resumen

Moodle es un servidor de contenidos encarado a los lugares o centros educativos, y está ampliamente implementado en multitud de centros.

La mayoría de estos centros, sin embargo, no sólo ofrecen contenidos on-line sino que Moodle les sirve como herramienta de apoyo a sus clases.

Timetable Block, está destinado a ser utilizado en estos centros como herramienta informativa y de gestión de los grupos en que cada curso está dividido y el lugar físico donde se realizan las clases.

En primer lugar, se ofrece información inmediata del lugar donde están programadas las clases y cualquier cambio en la programación de las clases se ve reflejado para todos los usuarios que tengan acceso.

Y en segundo lugar, sirve para ofrecer información adicional a los profesores sobre los accesos al sitio Moodle y si son efectuados con máquinas dentro del ámbito permitido y poder comprobar la asistencia de los alumnos.

Por tanto, en este bloque serán configurables tanto las horas lectivas como las aulas de las que dispone el centro o institución para la elaboración de las clases.

Resumiendo, este bloque nos ofrece las siguientes funcionalidades:

Una herramienta simple de gestión de los recursos físicos como serían las aulas donde se darían las clases. Esta gestión implicaría el manejo para cada aula de su capacidad así como también, si se trata de una aula con ordenadores conectados en red, del rango de direcciones IP que tendrían las máquinas ubicadas en esa aula.

Una herramienta de gestión en la que los profesores pueden confeccionar los horarios de los cursos que ellos imparten. Esta gestión implica la gestión de los diferentes grupos en que se puede dividir un curso, como podrían ser los grupos de clases prácticas y los grupos de clases teóricas, así como también la asignación del aula, de entre las disponibles en el momento de la clase.

Una herramienta informativa para los profesores para poder consultar en todo momento cuáles son los alumnos de cada grupo y desde qué máquinas se han realizado los accesos durante las horas de clase, y así poder llevar un control de la asistencia virtual.

Y finalmente una herramienta informativa para el alumnado para poder consultar los horarios y la disponibilidad de los grupos en los que se dividen los cursos a los que está matriculado.

Por tanto, en este bloque serán configurables tanto las horas lectivas como las aulas de las que dispone el centro o institución para la elaboración de las clases.

# Índex

<b>1</b>	<b>Introducció</b>	<b>3</b>
1.1	Què és un CMS	3
1.2	Moodle	3
1.2.1	Què és	3
1.2.2	Filosofia	4
1.2.3	Característiques	4
1.2.4	Com funciona	4
1.2.4.1	Requisits	4
1.2.4.2	Usuaris	5
1.2.4.3	Estructura de directors de Moodle	6
1.2.4.4	Creació d'una base de dades buida	6
1.2.4.5	Creació del directori de dades (moodledata)	6
1.2.4.6	Creació d'un curs	6
1.2.4.7	Fent un nou plug-in	7
1.2.4.8	Moduls d'activitat	7
1.2.4.9	Recursos	9
1.2.4.10	Blocs	10
1.3	Timetable Block	10
<b>2</b>	<b>Especificació funcional – Primera revisió</b>	<b>12</b>
2.1	Casos d'ús	12
2.1.1	Administrador	12
2.1.2	Professor	13
2.1.3	Alumne	14
2.2	Entitat-Relació	14
2.2.1	Entitats	14
2.2.2	Relacions	15
2.3	Esquema bàsic de la base de dades	15
2.4	Fitxer xmldb per la creació de taules	16
2.5	Diagrama d'estats	19
2.6	Interfície d'usuari	20
2.6.1	Administrador	20
2.6.2	Professor	21
2.6.3	Alumne	22
<b>3</b>	<b>Especificació funcional – Revisió final</b>	<b>24</b>
3.1	Casos d'ús	24
3.1.1	Administrador	25
3.1.2	Professor	26
3.1.3	Alumne	27
3.2	Entitat-Relació	27
3.2.1	Entitats	27
3.2.2	Relacions	28
3.3	Esquema bàsic de la base de dades	29
3.4	Fitxer xmldb per la creació de taules	30
3.5	Diagrama d'estats	34
3.6	Interfície d'usuari	34

<b>4</b>	<b>Evolució del projecte i decisions de disseny</b>	<b>35</b>
4.1	Previsió de l'evolució del projecte	35
4.2	Evolució del projecte	35
4.2.1	Anàlisi	36
4.2.1.1	Base de dades	36
4.2.1.2	Rols d'usuari	38
4.2.1.3	Interfície d'usuari	39
4.2.1.4	Disseny. Model implementat	39
4.2.1.5	Control de l'assistència	40
4.2.1.6	Flexibilitat amb la configuració del calendari de Moodle	41
4.2.1.7	Internacionalització i ajuda	42
4.2.1.8	Versió de Moodle	42
4.2.2	Programació	43
4.2.2.1	Objectes d'accés a dades	43
4.2.2.2	Interacció amb l'usuari i Lògica de l'aplicació	43
4.2.3	Fase de proves. Problemes i resolució.	44
4.2.3.1	Taules amb camp Id	44
4.2.3.2	Funció time() i zones horàries	44
4.2.3.3	Visualització del registre d'assistència	45
<b>5</b>	<b>Guia d'instal·lació</b>	<b>46</b>
<b>6</b>	<b>Manual d'usuari</b>	<b>50</b>
6.1	Introducció	50
6.2	Editant les aules	52
6.3	Editant hores de classe	56
6.4	Editant Grups	59
6.5	Veient els grups	66
<b>7</b>	<b>Avaluacions</b>	<b>74</b>
7.1	Servidor	74
7.2	Client	75
<b>8</b>	<b>Conclusions</b>	<b>76</b>
<b>9</b>	<b>Annex. Extractes de la documentació utilitzada.</b>	<b>77</b>
9.1	Definint el fitxer xmldb	77
9.2	Variables globals	78
9.3	Definició de rols. Fitxer access.db.	79
9.4	Funcions d'accés a dades.	80
9.5	Debat sobre l'obligatorietat del camp ID.	82
9.6	Correcció de problemes amb la funció usergetdate().	83

## Capítol 1

# Introducció

### 1.1 Què és un CMS

CMS són les sigles de Content Management System, en català, Sistema de Gestió de Continguts. Els gestors de continguts són sistemes per mantenir i actualitzar webs de manera fàcil. Normalment funcionen a través del mateix web, i a vegades a través d'aplicacions específiques. Els sistemes de gestió de continguts que s'han extès més són els que tenen origen al programari lliure. Gràcies a llicències com la GNU GPL, el codi font és públic i qualsevol és lliure de modificar-lo, copiar-lo o distribuir-lo. En conseqüència, qualsevol programador té accés al codi i pot col·laborar lliurement en tots aquests projectes, fent possible el seu desenvolupament i extensió. Els CMS més coneguts són programats amb llenguatges interpretats tals com PHP, Perl, Ruby, pel que fa a programari lliure, i amb ASP o JSP, pel que fa al privatiu. Els objectius dels CMS són molt variats: intranets, blogs, portals de notícies, webs d'empreses...

### 1.2 Moodle

#### 1.2.1 Què és

Moodle és un paquet de programari per a la creació de cursos i llocs web basats en Internet. És un projecte en desenvolupament dissenyat per a donar suport a un marc d'educació constructivista social. Moodle es distribueix de manera gratuïta com a programari lliure (Open Source) (sota la Llicència Pública GNU). Bàsicament això vol dir que Moodle té drets d'autor (copyright), però que teniu algunes llibertats. Podeu copiar, utilitzar i modificar Moodle sempre que accepteu: proporcionar el codi font als altres, no modificar o eliminar la llicència original i els drets d'autor, i aplicar aquesta mateixa llicència a qualsevol treball que en derivi. Llegiu la llicència per a més detalls i contacteu amb l'amo dels drets d'autor directament si teniu alguna pregunta. Moodle pot funcionar en qualsevol ordinador on pugui córrer PHP i suporta diversos tipus de base de dades (en especial MySQL). La paraula Moodle era al principi un acrònim de Modular Object-Oriented Dynamic Learning Environment (Entorn d'Aprenentatge Dinàmic Orientat a Objectes i Modular), la qual cosa resulta fonamentalment útil per a programadors i teòrics de l'educació. També és un verb que descriu el procés de deambular peresosament a través d'alguna cosa i fer les coses quan se t'acut fer-les, un plaent nyap que sovint et porta a a visió i a la creativitat. Totes dues accepcions s'apliquen a la manera en què es va desenvolupar Moodle i a la manera en què un estudiant o podria aproximar-se a l'estudi o ensenyament d'un curs en línia. Tot aquell qui utilitza Moodle és un Moodler. El projecte fou iniciat, i és mantingut, per l'australià Martin Dougiamas. Però ha aconseguit reunir una gran comunitat internacional d'usuaris, programadors, desenvolupadors i traductors de tal forma que actualment té 75.000 usuaris registrats i està traduït a 70 llengües, inclòs el català. La clau del seu èxit rau en la facilitat d'instal·lació (només necessita un servidor php i una base de dades com MySQL), de configuració, creació de cursos i manteniment del lloc. Està construït sota una òptica de construcció del coneixement basat en el diàleg entre els participants, el constructivisme social. Per això és tan fàcil crear-hi fòrums. Tot i això accepta molts altres tipus d'activitats didàctiques: qüestionaris, lliçons, tasques, wikis, xats, etc.

#### 1.2.2 Filosofia

El diseny i desenvolupament de Moodle es basen en la idea de la pedagogia construccionista social. Aquesta filosofia radica en la cooperació en grup i en la idea de que el coneixement es pot crear o construir. Com a curiositat, cal fer menció que Moodle és la primera comunitat Open Source no formada íntegrament per informàtics, sinó que també participen pedagogs i altra gent relacionada amb l'ensenyament.

#### 1.2.3 Característiques

Promou la pedagogia constructivista social.  
Es poden fer les classes 100% on-line.  
Té una interfície senzilla, lleugera i molt intuïtiva.  
Instal·lable en qualsevol plataforma que suporti PHP (només requereix l'existència d'una base de dades).  
Ofereix un llistat dels cursos amb descripcions i la posibilitat d'entrar-hi com a usuari convidat.  
Seguretat ben sòlida, els formularis es revisen, les cookies s'encripten.  
El tema de la interfície és altament personalitzable.  
Els paquets d'idioma permeten localitzar a qualsevol idioma per complet (70 idiomes).  
Inclou diversos mòduls per a tallers, fòrums, consultes, xats, etc.  
Alta Disponibilitat.  
Escalabilitat.  
Fàcil ús.  
Interoperable.  
Estable.  
Segur.

#### 1.2.4 Com funciona

##### 1.2.4.1 Requisits

Moodle està preparat per al Linux fent servir Apache, MySQL PHP (sovint es reconeix amb el nom LAMP el conjunt d'aquests quatre components), però també s'ha provat als sistemes operatius Windows XP/2000/2003 (WAMP), Solaris 10 (Sparc i x64), Mac OS X i Netware 6. També es pot treballar amb els servidors de dades PostgreSQL, Oracle i Microsoft SQL.

Els requisits per al Moodle són els següents:

- Maquinari (si utilitzeu un servidor propi):
  - Espai de disc: 160Mb lliures (com a mínim). Us caldrà més espai lliure per desar-hi els materials d'ensenyament.
  - Memòria: 256Mb (mínim), 1Gb (recomanat). Com a norma general, Moodle suportarà 50 usuaris "concurrents" per cada gigabit de memòria RAM, però això pot variar en funció del maquinari específic i la combinació de programes.
- Programari:
  - Servidor web. La majoria de gent utilitza Apache, però el Moodle hauria de treballar correctament en qualsevol servidor web que suporti PHP, com ara IIS a les plataformes Windows. PHP imposa unes necessitats segons la versió del servidor web, que són complexes, i normalment el consell general que es pot donar és que es faci servir la versió més recent possible del servidor web escollit.
  - El llenguatge d'escripts PHP. (Penseu que hi ha hagut millores havent instal·lat el Moodle amb el programari PHP-Accelerator). De moment hi ha dues versions de PHP accessibles: PHP4 i PHP5.
    - \* Per al Moodle 1.4 o posterior, es suporten aquestes versions: PHP4 (versió 4.1.0 o posterior) o PHP5 (versió 5.1.0 o posterior).
    - \* Per al Moodle 1.6 o posterior, es suporten aquestes versions: PHP4 (versió 4.3.0 o posterior) o PHP5 (versió 5.1.0 o posterior).

- \* Les versions futures del Moodle 2.0 o posteriors ja no acceptaran PHP4 i necessitaran PHP5 o posteriors).
- Un servidor de base de dades: MySQL i PostgreSQL estan completament suportats i recomanats. El suport per a Microsoft SQL Server i Oracle s'ha afegit al Moodle 1.7. MySQL és el que tria molta gent perquè és molt popular, però hi ha alguns arguments a favor de PostgreSQL, especialment si heu planejat un desenvolupament molt extens.
  - \* Per al Moodle 1.5 o posteriors, MySQL (versió 3.23 o posterior) o PostgreSQL (7.4 o posterior).
  - \* Per al Moodle 1.6 o posteriors, MySQL (versió 4.1.12 o posterior) o PostgreSQL (7.4 o posterior).
  - \* **Notes:** Per al Moodle 1.6 o posterior. Si utilitzeu codis de llenguatge només llatins, podeu utilitzar MySQL 4.1.12. Si utilitzeu codis no-llatins us caldrà MySQL 4.1.16 o posterior. De moment el paràmetre "strict mode" de MySQL ha de ser OFF (establert a "" o "MYSQL40") al fitxer de configuració de MySQL. La versió mínima de PostgreSQL és 7.4 i s'utilitzen molt la 8.0 i la 8.1. La versió mínima de Microsoft SQL Server és la 2005 (versió 9). També hi ha hagut intents amb SQL Server Express 2005.
- Requeriments addicionals de PHP
  - Paràmetres de PHP
    - \* El Safe mode ha d'estar en OFF (comproveu-ho al fitxer php.ini o al fitxer de configuració d'Apache).
    - \* El paràmetre memory\_limit ha de ser com a mínim 16M (es recomana que sigui 32M per al Moodle 1.7 o posterior).
    - \* El paràmetre de PHP session.save\_handler ha d'establir-se a files.
  - Extensions de PHP i llibreries
    - \* L'extensió mbstring es necessita per controlar cadenes multi-byte.
    - \* L'extensió iconv també es necessita per al Moodle 1.6 o posterior.
    - \* Les llibreries i extensions GD library i FreeType 2 també es necessiten per poder mostrar els gràfics dinàmics que es fan a les pàgines de registres.
    - \* L'extensió mysql cal si utilitzeu la base de dades MySQL. Fixeu-vos que en algunes distribucions de Linux (especialment Red Hat) es tracta d'una instal·lació opcional.
    - \* L'extensió pgsql cal si utilitzeu la base de dades PostgreSQL.
    - \* L'extensió zlib cal per afegir la funcionalitat zip de compressió i descompressió.
    - \* Pot ser que calguin altres extensions PHP per poder suportar altres funcionalitats del Moodle, especialment les d'autenticació i/o enrolament (p.ex. l'extensió LDAP per a l'autenticació LDAP i l'extensió de sockets per al servidor de Xat)

### 1.2.4.2 Usuaris

A més del maquinari i del programari, també heu de pensar en la capacitat de la instal·lació Moodle en funció del nombre d'usuaris que ha de poder suportar. Cal tenir en compte dos números:

- **Usuaris de navegació:** el nombre màxim d'usuaris que podran navegar pel lloc Moodle. És el nombre d'ordinadors del centre o del curs (el més alt).
- **Usuaris concurrents de la base de dades:** el nombre màxim d'usuaris concurrents de la base de dades (en activitats Moodle com ara qüestionaris). És el nombre d'usuaris que utilitzaran el Moodle simultàniament. En una institució educativa, calculeu amb els horaris i llistes en mà.

Quan sapigueu aquestes quantitats, podeu deduir el model de la instal·lació. El nombre exacte d'usuaris depèn de la combinació maquinari/programari/xarxa. Normalment la quantitat de memòria (RAM) instal·lada és el factor decisiu, però un processador més ràpid també contribuirà a reduir el temps de càrrega de les pàgines. La regla general per a un servidor sol és que el nombre màxim aproximat d'usuaris = RAM (en Gb) \* 50 i que el nombre màxim aproximat d'usuaris de navegació = Màxim aproximat d'usuaris concurrents \* 5. Per exemple, una universitat amb un total de 500 ordinadors al campus i 100 usuaris concurrents, necessitarà aproximadament 2Gb de RAM en un servidor per donar-li suport.

### 1.2.4.3 Estructura de directoris de Moodle

Un resum dels continguts del directori Moodle, per orientar-vos:

- config.php - conté paràmetres bàsics. El fitxer no es descarrega amb el moodle, es crea.
- install.php - l'escript que s'engega per crear el fitxer config.php
- version.php - defineix la versió en curs del codi Moodle
- index.php - la pàgina principal del lloc
- admin/ - codi per administrar tot el servei
- auth/ - mòduls dels connectors per a autenticació d'usuaris
- blocks/ - mòduls dels connectors per als blocs laterals de la majoria de pàgines
- calendar/ - tot el codi per al maneig i la visualització de calendaris
- course/ - codi per a mostrar i gestionar cursos
- doc/ - documentació d'ajuda per al Moodle (per exemple, aquesta pàgina)
- files/ - codi per a mostrar i gestionar fitxers penjats al lloc
- lang/ - textos en llengües diverses, un directori per a cada llengua
- lib/ - col·leccions de tot el codi Moodle
- login/ - codi per a controlar l'entrada al lloc i creació de comptes d'usuari
- mod/ - tot el codi dels mòduls de gestió de cursos
- pix/ - els gràfics genèrics del lloc
- theme/ - paquets de temes/pells per a canviar l'aspecte del lloc.
- user/ - codi per al maneig, dades i llistes d'usuaris

### 1.2.4.4 Creació d'una base de dades buida

Caldrà que creeu una base de dades buida (per exemple, "moodle") al sistema de base de dades, i un usuari especial (per exemple, "usuari\_moodle") que tingui accés a aquesta base de dades (aquesta i prou). Podeu fer servir l'usuari "root" si treballeu en un servidor de prova, però no és gens recomanable en un sistema de producció: si un hacker arriba a descobrir el mot de pas tot el sistema de base de dades estarà en perill, no només una base de dades.

### 1.2.4.5 Creació del directori de dades (moodledata)

Moodle també necessitarà espai al disc dur del servidor per guardar-hi els fitxers penjats, com ara documents de curs i fotografies d'usuaris. L'instal·lador del Moodle intentarà crear-vos el directori però si no pot l'haureu de crear manualment.

### 1.2.4.6 Creació d'un curs

Quan el Moodle treballi correctament, podeu crear un curs per començar. Trieu "Afegir un curs" des de la pàgina d'Administració (o des dels vincles d'administració de cursos de la pàgina principal). Empleneu el formulari, amb una atenció especial quant al format de curs. No cal que us preocupeu excessivament pels detalls en aquest moment, perquè es pot canviar tot més endavant. Fixeu-vos que les icones grogues que hi ha pertot proporcionen ajuda contextual sobre qualsevol aspecte. Feu clic a "Desa els canvis", i arribareu a un altre formulari on podreu assignar professorat al curs. Només podreu afegir-hi comptes d'usuari existents - si voleu crear comptes de professorat, demaneu a cada professor que es creï el seu compte personal (vegeu la pàgina d'entrada), o creeu-ne un a través dels vincles "Usuaris", "Afegeix usuaris" de la pàgina d'Administració. En acabar, el curs ja està a punt de solfa, i és accessible al vincle "Cursos" de la pàgina principal.

### 1.2.4.7 Fent un nou plug-in

La M de Moodle ve de modular, i la forma més fàcil i més mantenible d'afegir noves funcionalitats a Moodle és usant un dels molts plug-ins de les API. Hi ha molts tipus de plug-ins que pots escriure:

- Activity modules
- Admin reports
- Assignment types
- Authentication plugins
- Blocks
- Course formats
- Course reports
- Database fields
- Database presets
- Enrolment plugins
- Filters
- Grade export/import
- Grade report
- Question engine
- Question import/export formats
- Quiz reports
- Resource types
- SSO plugins

Els tres tipus de plug-ins més utilitzats i en els que la comunitat Moodle col·labora amb més aportacions són els que a continuació es defineixen breument: Mòduls d'activitat, Recursos i Blocs.

### 1.2.4.8 Mòduls d'activitat

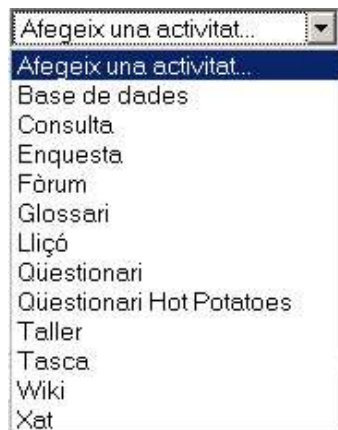


Figura 1.1: Llista de mòduls d'activitat per afegir.

Moodle disposa d'activitats d'aprenentatge diverses que podeu afegir als vostres cursos (Figura 1.1). La comunicació i la col·laboració entre els estudiants es pot produir utilitzant els xats i els fòrums. Els missatges permeten converses bidireccionals entre dos participants d'un curs. També poden treballar de manera col·laborativa editant de manera conjunta un wiki, una manera excel·lent per tal que els estudiants treballin plegats fent un projecte conjunt.

Les consultes ens permeten recollir l'opinió dels participants sobre un tema concret. També podem crear glossaris de paraules o termes clau i fins i tot permetre que els estudiants creïn les entrades (i avaluar-les). Les bases de dades, disponibles a partir de la versió 1.6, també es poden utilitzar amb aquest objectiu.

Els estudiants ens poden trametre els seus treballs mitjançant les tasques, que poden escriure directament en línia o bé mitjançant un fitxer que prèviament han elaborat. Els tallers permeten també la tramesa de treballs que són, a més, revisats entre els estudiants. Els qüestionaris disposen de diferents tipus de preguntes (opció múltiple, aparellament,...) amb correcció automàtica. També podem incloure activitats que haguem elaborat amb el programa HotPotatoes i integrar les qualificacions que l'estudiant obté a la seva taula de qualificacions del curs.

Les enquestes ja vénen predefinides, i poden ser útils per a convidar a la reflexió i conèixer la concepció de l'aprenentatge dels estudiants.

Però, a més, hi ha tot un seguit d'activitats no estàndard que no es troben al programa oficial de Moodle però que es poden instal·lar fàcilment. Entre elles podem destacar les activitats JClic o el mòdul de Quaderns Virtuals.

Aquests són amb molt els mòduls més importants, i es troben al directori "mod". Per defecte hi ha set mòduls: Tasca, Consulta, Fòrum, Glossari, Qüestionari, Recurs i Enquesta. Cada mòdul està en un subdirectorí separat i consisteix amb els següents elements obligatoris (més els scripts extra que són únics per a cada mòdul):

- mod.html: un formulari per establir o actualitzar una instància d'aquest mòdul
- version.php: defineix alguna meta-informació i proporciona codi d'actualització
- icon.gif: una icona de 16x16 per al mòdul
- db/: bolcats SQL de totes les taules i dades requerides de una base de dades (per a cada tipus de base de dades)
- index.php: una pàgina per representar la llista de totes les instàncies en un curs
- view.php: una pàgina per veure una instància en particular
- lib.php: qualsevol/totes les funcions definides per al mòdul haurien d'ésser aquí.

Si el mòdul es diu "andromina", llavors les funcions requerides inclouen:

- andromina\_add\_instance() - codi per afegir una nova instància d'andromina
- andromina\_update\_instance() - codi per actualitzar una instància existent
- andromina\_delete\_instance() - codi per esborrar una instància
- andromina\_user\_outline() - donada una instància, retorna un resum d'una contribució d'un usuari
- andromina\_user\_complete() - donada una instància, imprimeix detalls sobre la contribució d'un usuari

Altres funcions interessants però no obligatòries són:

- andromina\_delete\_course() - per esborrar tot el que sigui necessari després d'esborrar totes les instàncies d'un curs
- andromina\_process\_options() - per pre-processar la informació dels ajustaments de la instància

Per evitar possibles conflictes, qualsevol de les funcions d'un mòdul haurà de ser nombrada començant amb andromina\_ (el nom del mòdul més un guió baix) i qualsevol constant que defineixi vostè haurà de començar amb ANDROMINA\_

- config.html - (opcional) un formulari per ajustar les preferències globals del mòdul

Finalment, cada mòdul tindrà alguns fitxers d'idioma que contenen cadenes per aquest mòdul. Llegeixi més avall.

IMPORTANT: Al crear un nou mòdul, el nom del mateix haurà de ser una paraula senzilla, amb anglès, que no inclogui números o altres caràcters especials. També haurà d'assegurar-se de que el seu mòdul proporciona suport per algunes característiques com grups, metacursos i còpies de seguretat i restauració.

#### 1.2.4.9 Recursos

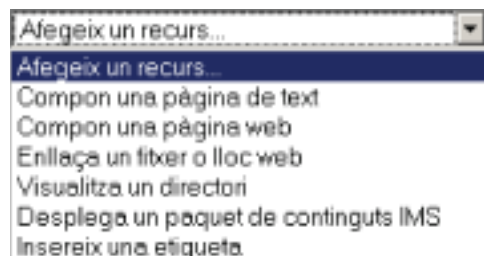


Figura 1.2: Llista de recursos disponibles.

Moodle disposa de diversos recursos que ens permeten posar qualsevol tipus de contingut en format digital a l'abast dels estudiants. Per tal d'afegir-los, amb l'edició activada, hem de seleccionar l'opció corresponent del menú (Figura 1.2) "Afegeix un recurs". Una Pàgina de text és un recurs senzill que consisteix en una pàgina escrita en text sense format (en principi, tot i que accepta etiquetes html). Les pàgines de text són molt ràpides de fer i, tot i que no permeten donar format al text d'una manera senzilla, poden ser una bona opció per a escriure informacions curtes o instruccions breus. Si la informació que voleu donar als estudiants requereix un cert format del text amb títols, taules, imatges,... és millor que utilitzeu el recurs Composa una pàgina web, que posarà al vostre abast un editor del tipus WYSIWYG. Els recursos poden tenir formats molt diversos. En aquest cas hem d'utilitzar l'opció Enllaçar un fitxer o lloc web o bé posar a l'abast dels estudiants els fitxers que hem pujat a un directori del curs mitjançant el recurs Visualitza un directori. Si disposeu d'un paquet de contingut IMS podeu utilitzar el recurs Desplega un paquet de contingut IMS. Les etiquetes poden ser molt útils per a mostrar informació o instruccions. També ens permeten afegir imatges i altres fitxers multimèdia a les diferents seccions del curs de manera que siguin visibles directament a la pàgina principal del curs.

#### 1.2.4.10 Blocs

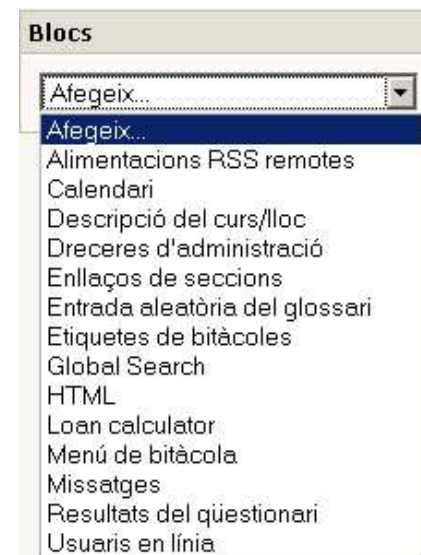


Figura 1.3: Llista de blocs disponibles per ser afegits.

Les pàgines principals d'un curs contenen, generalment, blocs a les bandes esquerra i dreta, amb una columna central que mostra els continguts del curs. Amb l'edició activada es poden afegir, esborrar o amagar blocs, moure'ls amunt i avall i d'esquerra a dreta o vice-versa. La imatge mostra els blocs que podem afegir al curs: "Usuaris en línia", "Missatges", "Calendari",... (Figura 1.3) Una llista de més de 16 tipus diferents de blocs pot proveir informació o funcionalitats addicionals per als professors i estudiants. A la dreta es mostren els blocs estàndard que venen amb Moodle. Hi ha molts més blocs no estàndard desenvolupats per Moodlers que un administrador pot instal·lar i afegir a aquesta llista. Un professor amb drets d'editor tindrà també un bloc d'administració. Aquest bloc té submenús per a la gestió del curs: còpia de seguretat i restauració, inscripcions, format, informes, qualificacions, registres d'activitat, fitxers i altres eines útils.

### 1.3 Timetable Block

Els blocs poden ser aplicats en quant a disponibilitat per tot un curs o per tot el lloc Moodle en sí. Timetable Block, està destinat a ser utilitzat en cursos del centre que a part de tenir la part on-line també realitzen classes de periodicitat semanal en llocs físics.

Per tant, en aquest bloc seràn configurables tant les hores lectives com les aules de les que disposa el centre o institució per l'elaboració de les classes.

Un administrador del lloc, que coneixi les hores en que el centre fa classes haurà de crear aquestes hores en la configuració del bloc, així com també les aules de que disposa el centre.

Cada curs, en molts centres considerats com assignatures, es podrà dividir en diversos grups.

D'aquests grups en hi han, a nivell lògic, de dos tipus: els grups comuns i els grups "normals".

Els grups normals són exclusius entre sí, és a dir, si un alumne està apuntat a un dels grups no pot ser-hi en un altre. En canvi els grups comuns poden conuiu amb els grups normals en un curs però apuntar-s'hi no exclourà a l'alumne de cap altre grup.

Per exemplificar-ho amb un cas concret, un curs podria tenir un grup comú on el professor podria

donar els conceptes teòrics en un aula amb prou capacitat per tots els alumnes que hi estessin apuntats i dos grups on els alumnes es repartirien per poder posar en pràctica aquests conceptes però degut a les capacitats de les aules en un grup no podrien accedir-hi tots els alumnes que sí poden anar a les classes del grup comú.

Els professors que imparteixin les classes coneixedors de quina manera es repartiran les classes i els grups hauràn de crear per cada curs els grups i assignar-los-hi les hores de classe i les aules creades per l'Administrador.

Els alumnes podran consultar aquests horaris i apuntar-s'hi als grups si s'escau.

Per últim es podrà comprovar l'assistència dels estudiants a les classes ja que s'emmagatzema un registre de les classes realitzades fins el moment de la visualització i es mostra mitjançant un creuament amb el registres que té el sistema Moodle amb els accessos dels usuaris i desde quina IP han estat realitzats.

## Capítol 2

# Especificació funcional – Primera revisió

Aquest capítol recull la informació necessària a partir de la qual, es podria començar a desenvolupar el mòdul. En un primer moment no és va tenir en compte la possibilitat que el professorat pogués controlar l'assistència dels alumnes i tampoc contempla les decisions que podrien ser derivades de possibles problemes durant la programació del bloc.

### 2.1 Casos d'ús

S'adjunta imatge (Figures 2.1, 2.2 i 2.3). Hi hauran tres rols per aquesta aplicació:

#### 2.1.1 Administrador

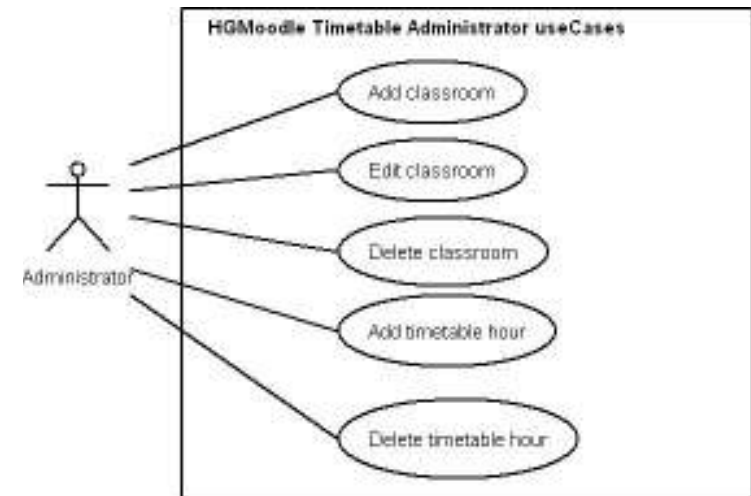


Figura 2.1: Casos d'ús per l'administrador.

Encarregat de gestionar (crear,editar,esborrar) les aules.

El fet de gestionar les aules implica tenir-hi assignat per cadascuna d'elles com a mínim un identificador. També seria possible tenir com a atribut la capacitat de l'aula.



L'altra parametrització que farà l'administrador serà introduir les hores lectives al llarg del dia i els dies lectius de la setmana.

2.1.2 Professor

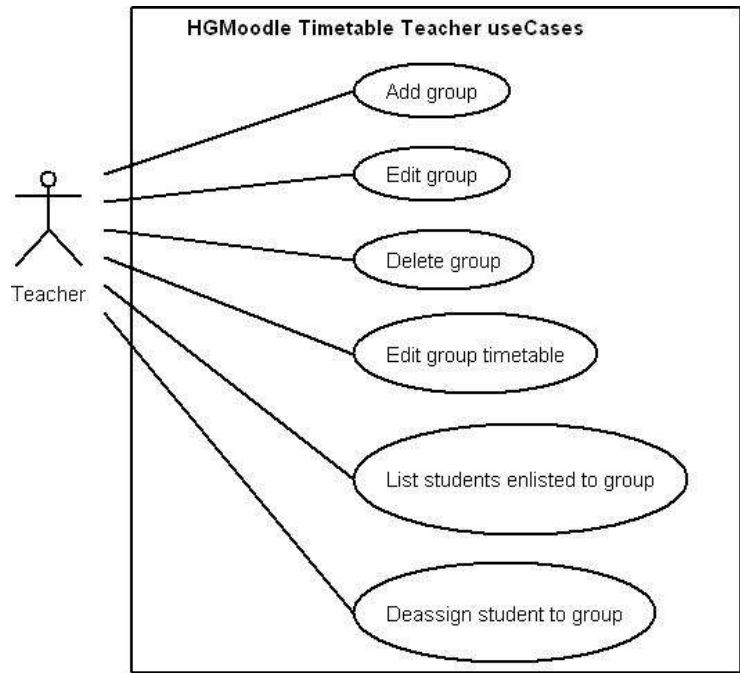


Figura 2.2: Casos d'ús per al professor.

Encarregat de gestionar els grups. Els grups tindran com a mínim un identificador, així com també el numero d'alumnes assignats per cada grup. Establir un horari serà el cas en que el professor li haurà d'assignar un horari i una aula disponibles per la seva classe. Tindrà també l'opció de permetre o no als alumnes canviar-se de grup. Per ultim podra veure el llistat d'alumnes apuntat a cada grup i podrà desassignar-lo si cal.

2.1.3 Alumne

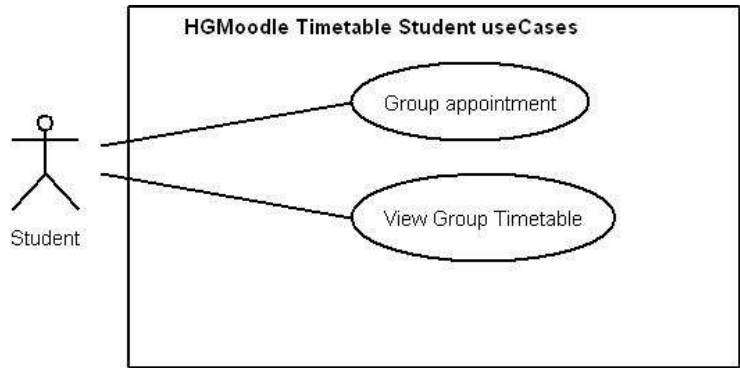


Figura 2.3: Casos d'ús per a l'estudiant.

Únicament tenen el cas de apuntar-s'hi a un grup dels creats. Si ja està apuntat tindrà la possibilitat de visualitzar l'horari del grup al qual s'ha assignat i canviar de grup si el professor permet aquesta acció.

2.2 Entitat-Relació

En els següents subapartats es mostra una enumeració de les entitats que intervenen dins el sistema (secció 2.2.1, pàgina 14) i la relació que hi ha entre elles (secció 2.2.2, pàgina 15).

2.2.1 Entitats

<b>Grup</b> Identificador Número d'alumnes	<b>Timetable</b> Identificador Identificador Curs
<b>Hores vàlides</b> Hora d'inici Hora de fi	<b>Horari</b> Hora inici Hora fi Dia setmanal Identificador de grup Identificador de aula Identificador Curs
<b>Professor _ Grup</b> Identificador Professor Identificador Grup	<b>Aula</b> Identificador Capacitat
<b>Alumne _ Grup</b> Identificador Alumne Identificador Grup	<b>Dies lectius</b> Nom del dia setmanal

### 2.2.2 Relacions

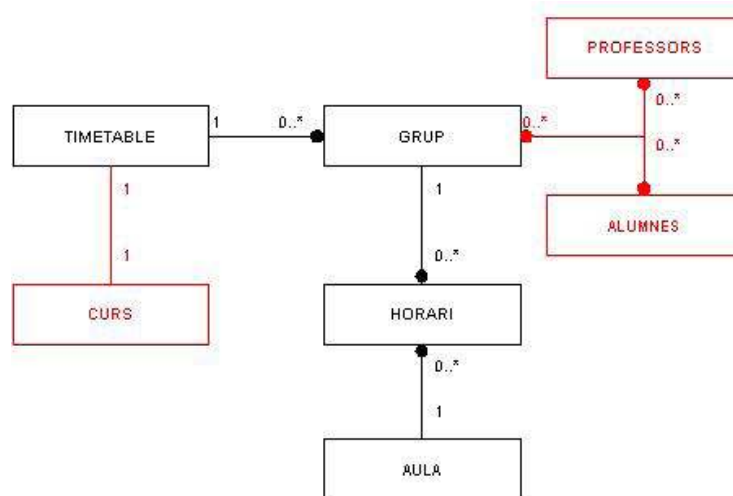


Figura 2.4: Diagrama de relacions entre les entitats. En vermell entitats ja existents a Moodle.

Un grup en un horari, està situat en una aula. A una aula, en un horari determinat només hi haurà com a màxim un grup (o cap). Un grup repetirà a una aula  $n$  cops.

Cada assignació tindrà una durada, és a dir, tindrà una data d'inici i una data de fi.

Es considera que cada grup a una hora determinada i un dia setmanal determinat estarà situat en una aula (que sempre serà la mateixa). Això vol dir, que durant una setmana en diferents dies de la setmana i en una hora determinada d'aquest dia de la setmana un grup estarà a l'aula que te assignada. És a dir, un professor assignarà una aula, i independentment d'aquesta assignació el grup tindrà un horari setmanal.

Professor i Alumne i Curs són entitats que són gestionades pel Moodle, s'haurà de veure de quina manera s'obtenen aquestes dades, però no les gestionarà aquesta aplicació.

Cada grup es gestionat per algun o alguns professors (o cap). Cada alumne pertany a un grup (o cap).

### 2.3 Esquema bàsic de la base de dades

Diagrama extret de la base de dades, per representar explícitament les taules i els camps necessaris i les relacions entre aquestes taules.

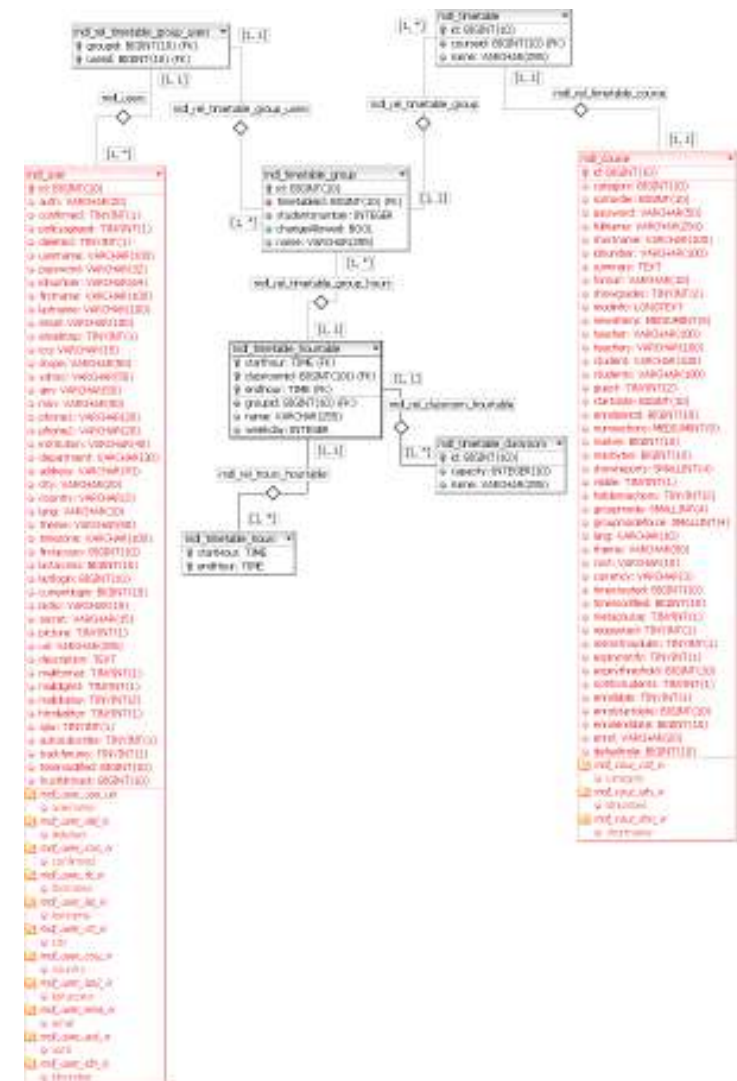


Figura 2.5: Esquema de la base de dades. En vermell taules de Moodle.

## 2.4 Fitxer xmldb per la creació de taules

Tal com s'explica en la secció 4.2.1.1 (pàgina 36) aquest és el fitxer que necessita Moodle per crear les taules necessàries sense haver de crear scripts especialitzats per cadascun dels sistemes gestors de

base de dades en els que es vulgui fer compatible el mòdul o bloc.

Es tracta, doncs, d'un fitxer XML que serveix com capa d'abstracció entre el desenvolupament i la base de dades.

Listing 2.1: Fitxer xmldb install.xml per la creació de les taules

```

1 <?xml version="1.0" encoding="UTF-8" ?>
2 <XMLDB PATH="blocks/timetable/db" VERSION="20070922" COMMENT="XMLDB file for
  Moodle blocks/timetable"
3 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4 xsi:noNamespaceSchemaLocation="../../../lib/xmldb/xmldb.xsd">
5 <TABLES>
6 <TABLE NAME="timetable_hours" COMMENT="timetable_hours table retrofitted
  from MySQL" NEXT="timetable_classroom">
7 <FIELDS>
8 <FIELD NAME="starthour" TYPE="time" LENGTH="small" NOTNULL="true"
  SEQUENCE="false" ENUM="false" NEXT="endhour"/>
9 <FIELD NAME="endhour" TYPE="time" LENGTH="small" NOTNULL="true"
  SEQUENCE="false" ENUM="false" PREVIOUS="starthour"/>
10 </FIELDS>
11 <KEYS>
12 <KEY NAME="primary" TYPE="primary" FIELDS="starthour, endhour"
  COMMENT="Primary key for timetable_hours"/>
13 </KEYS>
14 </TABLE>
15 <TABLE NAME="timetable_classroom" COMMENT="timetable_classroom table
  retrofitted from MySQL" PREVIOUS="timetable_hours" NEXT="timetable">
16 <FIELDS>
17 <FIELD NAME="id" TYPE="int" LENGTH="10" NOTNULL="true" UNSIGNED="true"
  SEQUENCE="true" ENUM="false" NEXT="capacity"/>
18 <FIELD NAME="capacity" TYPE="int" LENGTH="10" NOTNULL="true" UNSIGNED=
  "true" SEQUENCE="false" ENUM="false" PREVIOUS="id" NEXT="name"/>
19 <FIELD NAME="name" TYPE="char" LENGTH="255" NOTNULL="true" SEQUENCE="
  false" ENUM="false" PREVIOUS="capacity"/>
20 </FIELDS>
21 <KEYS>
22 <KEY NAME="primary" TYPE="primary" FIELDS="id" COMMENT="Primary key
  for timetable_classroom"/>
23 </KEYS>
24 </TABLE>
25 <TABLE NAME="timetable" COMMENT="timetable table retrofitted from MySQL"
  PREVIOUS="timetable_classroom" NEXT="timetable_group">
26 <FIELDS>
27 <FIELD NAME="id" TYPE="int" LENGTH="10" NOTNULL="true" UNSIGNED="true"
  SEQUENCE="true" ENUM="false" NEXT="courseid"/>
28 <FIELD NAME="courseid" TYPE="int" LENGTH="10" NOTNULL="true" UNSIGNED=
  "false" SEQUENCE="false" ENUM="false" PREVIOUS="id" NEXT="name"/>
29 <FIELD NAME="name" TYPE="char" LENGTH="255" NOTNULL="false" SEQUENCE=
  "false" ENUM="false" PREVIOUS="courseid"/>
30 </FIELDS>
31 <KEYS>
32 <KEY NAME="primary" TYPE="primary" FIELDS="id" COMMENT="Primary key
  for timetable" NEXT="timetablecourseid" />
33 <KEY NAME="timetablecourseid" TYPE="foreign" FIELDS="courseid"
  REFTABLE="course" REFFIELDS="id" PREVIOUS="primary" />
34 </KEYS>
35 <INDEXES>
36 <INDEX NAME="courseid" UNIQUE="false" FIELDS="courseid"/>
37 </INDEXES>
38 </TABLE>
39 <TABLE NAME="timetable_group" COMMENT="timetable_group table retrofitted
  from MySQL" PREVIOUS="timetable" NEXT="timetable_users">
40 <FIELDS>

```

```

41 <FIELD NAME="id" TYPE="int" LENGTH="10" NOTNULL="true" UNSIGNED="true"
  SEQUENCE="true" ENUM="false" NEXT="timetableid"/>
42 <FIELD NAME="timetableid" TYPE="int" LENGTH="10" NOTNULL="true"
  UNSIGNED="false" SEQUENCE="false" ENUM="false" PREVIOUS="id" NEXT=
  "studentsnumber"/>
43 <FIELD NAME="studentsnumber" TYPE="int" LENGTH="10" NOTNULL="false"
  UNSIGNED="true" SEQUENCE="false" ENUM="false" PREVIOUS="
  timetableid" NEXT="changeallowed"/>
44 <FIELD NAME="changeallowed" TYPE="int" LENGTH="1" NOTNULL="false"
  UNSIGNED="false" SEQUENCE="false" ENUM="false" PREVIOUS="
  studentsnumber" NEXT="name"/>
45 <FIELD NAME="name" TYPE="char" LENGTH="255" NOTNULL="false" SEQUENCE=
  "false" ENUM="false" PREVIOUS="changeallowed"/>
46 </FIELDS>
47 <KEYS>
48 <KEY NAME="primary" TYPE="primary" FIELDS="id" COMMENT="Primary key
  for timetable_group" NEXT="grouptimetableid"/>
49 <KEY NAME="grouptimetableid" TYPE="foreign" FIELDS="
  timetableid" REFTABLE="timetable" REFFIELDS="id" PREVIOUS=
  "primary" />
50 </KEYS>
51 <INDEXES>
52 <INDEX NAME="timetableid" UNIQUE="false" FIELDS="timetableid"/>
53 </INDEXES>
54 </TABLE>
55 <TABLE NAME="timetable_users" COMMENT="timetable_users table retrofitted
  from MySQL" PREVIOUS="timetable_group" NEXT="timetable_hourtable">
56 <FIELDS>
57 <FIELD NAME="groupid" TYPE="int" LENGTH="10" NOTNULL="true" UNSIGNED=
  "false" SEQUENCE="false" ENUM="false" NEXT="userid"/>
58 <FIELD NAME="userid" TYPE="int" LENGTH="10" NOTNULL="true" UNSIGNED="
  false" SEQUENCE="false" ENUM="false" PREVIOUS="groupid"/>
59 </FIELDS>
60 <KEYS>
61 <KEY NAME="primary" TYPE="primary" FIELDS="groupid, userid" COMMENT="
  Primary key for timetable_users" NEXT="usergroupid"/>
62 <KEY NAME="usergroupid" TYPE="foreign" FIELDS="groupid"
  REFTABLE="timetable_group" REFFIELDS="id" PREVIOUS="
  primary" NEXT="useruserid" />
63 <KEY NAME="useruserid" TYPE="foreign" FIELDS="userid"
  REFTABLE="users" REFFIELDS="id" PREVIOUS="usergroupid" />
64 </KEYS>
65 <INDEXES>
66 <INDEX NAME="userid" UNIQUE="false" FIELDS="userid"/>
67 </INDEXES>
68 </TABLE>
69 <TABLE NAME="timetable_hourtable" COMMENT="timetable_hourtable table
  retrofitted from MySQL" PREVIOUS="timetable_users">
70 <FIELDS>
71 <FIELD NAME="starthour" TYPE="text" LENGTH="small" NOTNULL="true"
  SEQUENCE="false" ENUM="false" NEXT="weekday"/>
72 <FIELD NAME="weekday" TYPE="char" LENGTH="1" NOTNULL="true" SEQUENCE=
  "false" ENUM="false" PREVIOUS="starthour" NEXT="classroomid"/>
73 <FIELD NAME="classroomid" TYPE="int" LENGTH="10" NOTNULL="true"
  UNSIGNED="false" SEQUENCE="false" ENUM="false" PREVIOUS="weekday"
  NEXT="endhour"/>
74 <FIELD NAME="endhour" TYPE="text" LENGTH="small" NOTNULL="true"
  SEQUENCE="false" ENUM="false" PREVIOUS="classroomid" NEXT="groupid"
  "/>
75 <FIELD NAME="groupid" TYPE="int" LENGTH="10" NOTNULL="true" UNSIGNED=
  "false" SEQUENCE="false" ENUM="false" PREVIOUS="endhour" NEXT="
  name"/>
76 <FIELD NAME="name" TYPE="char" LENGTH="255" NOTNULL="false" SEQUENCE=

```

## 2.6 Interfície d'usuari

Les pantalles amb les que els usuaris interaccionaran amb el sistema seran aproximadament així, ja que es tracta d'un primera aproximació al que podria ser la interfície d'usuari

### 2.6.1 Administrador

Administrador, interfície (Figura 2.7) per gestionar aules, creació, modificació i eliminació.



Figura 2.7: Interfície per gestionar aules.

Interfície (Figura 2.8) per manegar les hores lectives i dies lectius del centre i diversos parametres de configuració.

```

77      "false" ENUM="false" PREVIOUS="groupid"/>
78      </FIELDS>
79      <KEYS>
80      <KEY NAME="primary" TYPE="primary" FIELDS="starthour, weekday,
        groupid, classroomid, endhour" COMMENT="Primary key for
        timetable_hourtable" NEXT="starthourendhour" />
81      <KEY NAME="starthourendhour" TYPE="foreign" FIELDS="starthour
        , endhour" REFTABLE="timetable_hours" REFFIELDS="starthour,
        endhour" PREVIOUS="primary" NEXT="hourtableclassroomid"/>
82      <KEY NAME="hourtableclassroomid" TYPE="foreign" FIELDS="
        classroomid" REFTABLE="timetable_classroom" REFFIELDS="id"
        PREVIOUS="starthourendhour" NEXT="hourtablegroupid" />
83      <KEY NAME="hourtablegroupid" TYPE="foreign" FIELDS="groupid"
        REFTABLE="timetable_group" REFFIELDS="id" PREVIOUS="
        hourtableclassroomid"/>
84      </KEYS>
85      <INDEXES>
86      <INDEX NAME="classroomid" UNIQUE="false" FIELDS="classroomid" NEXT="
        starthour" />
87      <INDEX NAME="starthour" UNIQUE="false" FIELDS="starthour, endhour"
        PREVIOUS="classroomid" NEXT="groupid" />
88      <INDEX NAME="groupid" UNIQUE="false" FIELDS="groupid" PREVIOUS="
        starthour" />
89      </INDEXES>
90      </TABLE>
91      </TABLES>
92      </XMLDB>

```

## 2.5 Diagrama d'estats

La manera d'interactuar de l'usuari dependrà del seu rol i segons aquest les operacions que podrà realitzar cada usuari seran diferents. Per tant podem interpretar el següent diagrama (Figura 2.6) com a visió global del funcionament del sistema.

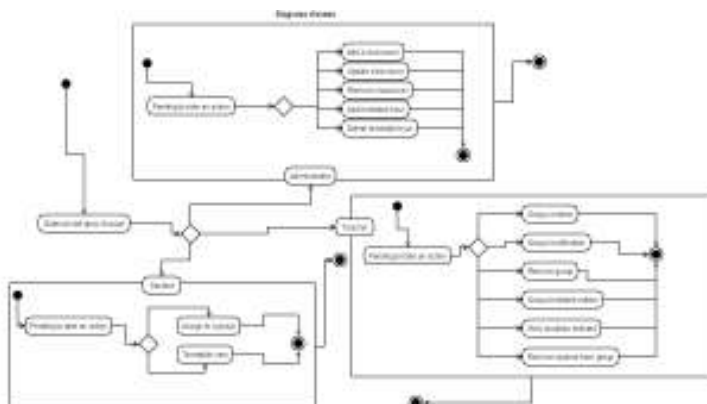


Figura 2.6: Diagrama de estats del sistema.

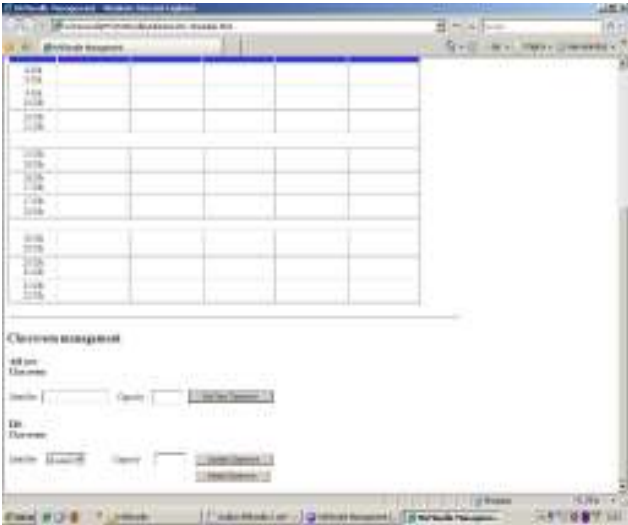


Figura 2.8: Interfície per gestionar aules.

2.6.2 Professor

Professor, interfície (Figura 2.9) per assignar horaris i aules a grups i manegar grups.



Figura 2.9: Interfície per gestionar horaris de grups.

Professor, visualització (Figura 2.10) dels alumnes apuntats al grup i opció de desassignació.

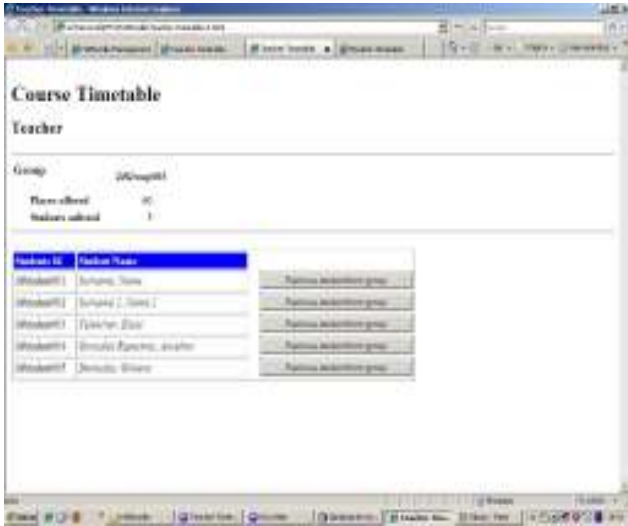


Figura 2.10: Visualització llistat d'alumnes.

2.6.3 Alumne

Alumne, interfície (Figura 2.11) per apuntar-se als grups.

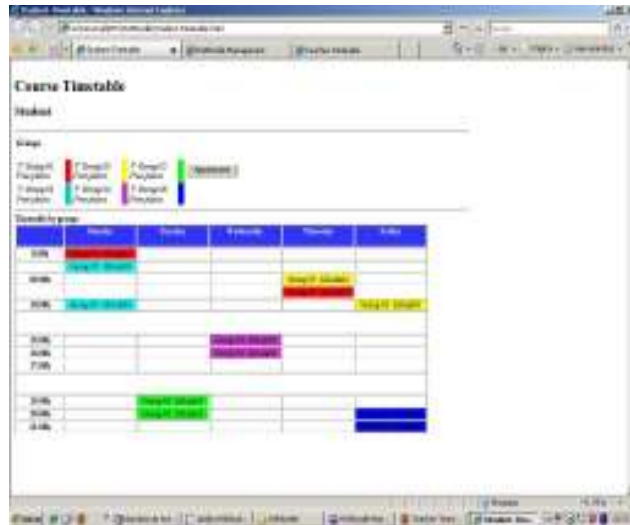


Figura 2.11: Interfície per apuntar-se a grups.

## Capítol 3

# Especificació funcional – Revisió final

Aquest capítol recull la informació necessària a partir de la qual, es podria començar a desenvolupar el mòdul. En aquest cas ja estan contemplades totes les decisions posteriors al plantejament inicial com ara el control de l'assistència dels alumnes per part dels professors o els canvis en les decisions de disseny que haguessin pogut ser produïdes degut a problemes durant la fase de desenvolupament. És a dir, en aquest capítol trobem la informació necessària per poder començar el desenvolupament del block en cas que s'hagués de tornar a implementar de nou, tenint en compte els problemes que s'han trobat a l'hora de desenvolupar-lo.

### 3.1 Casos d'ús

S'adjunta imatge (Figures 3.1, 3.2 i 3.3). Hi hauran tres rols per aquesta aplicació:

### 3.1.1 Administrador

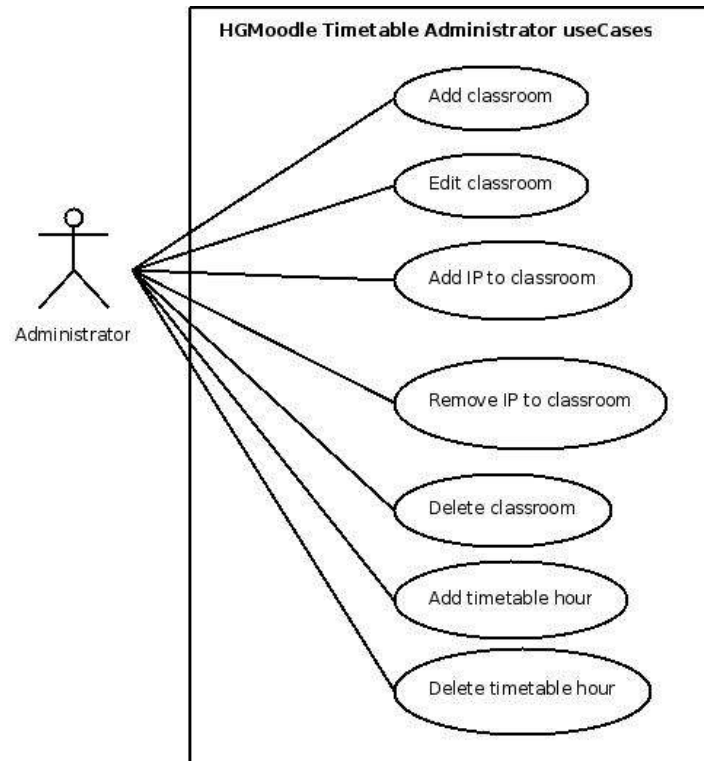


Figura 3.1: Casos d'ús per a l'administrador.

Encarregat de gestionar (crear, editar, esborrar) les aules.

El fet de gestionar les aules implica tenir-hi assignat per cadascuna d'elles com a mínim un identificador. També seria possible tenir com a atribut la capacitat de l'aula.

L'altra parametrització que farà l'administrador serà introduir les hores lectives al llarg del dia i els dies lectius de la setmana.

Per últim l'administrador es qui gestiona les adreces IP vàlides i s'encarrega d'associar (o des associar) aquestes adreces a cadascuna de les aules físiques.

### 3.1.2 Professor

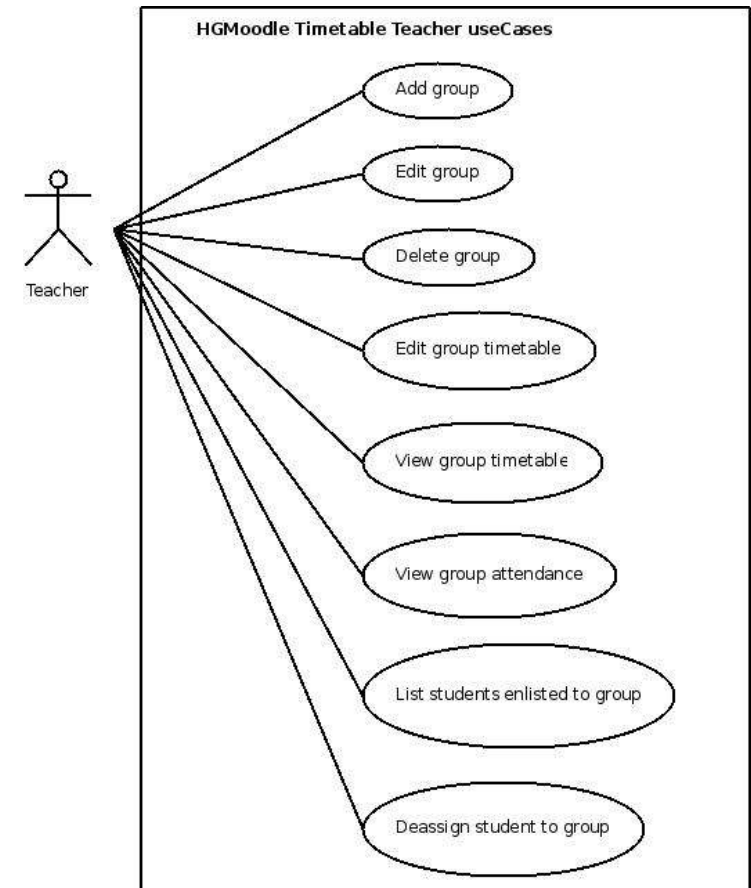


Figura 3.2: Casos d'ús per al professor.

Encarregat de gestionar els grups.

Els grups tindran com a mínim un identificador, així com també el numero d'alumnes assignats per cada grup. Establir un horari serà el cas en que el professor li haurà d'assignar un horari i una aula disponibles per la seva classe. Tindrà també l'opció de permetre o no als alumnes canviar-se de grup. També podrà veure el llistat d'alumnes apuntat a cada grup i podrà des assignar-lo si cal.

Per últim els professors tenen la possibilitat d'observar els accessos al sistema Moodle en cada moment que hi ha hagut cadascuna de les classes i comprovar si ha estat un accés des d'una adreça vàlida, de les associades a l'aula per l'administrador. Això li permetrà confirmar l'assistència dels alumnes a les classes.

3.1.3 Alumne

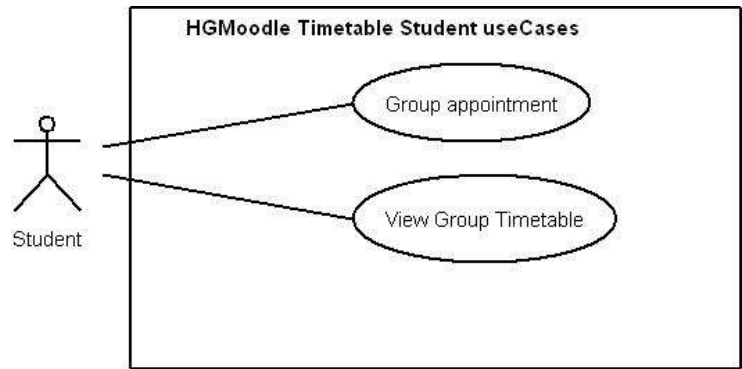


Figura 3.3: Casos d'ús per a l'estudiant.

Únicament tenen el cas de apuntar-s'hi a un grup dels creats. Si ja està à apuntat tindrà la possibilitat de visualitzar l'horari del grup al qual s'ha assignat i canviar de grup si el professor permet aquesta acció.

3.2 Entitat-Relació

En els següents subapartats es mostra una enumeració de les entitats que intervenen dins el sistema (secció 3.2.1, pàgina 27) i la relació que hi ha entre elles (secció 3.2.2, pàgina 28).

3.2.1 Entitats

A continuació hi ha una representació de la idea de les entitats que s'hauràn de gestionar amb el sistema.

<b>Grup</b> Identificador Número d'alumnes
--

<b>Timetable</b> Identificador Identificador Curs
---

<b>Hores vàlides</b> Hora d'inici Hora de fi
--

<b>Horari</b> Hora inici Hora fi Dia setmanal Identificador de grup Identificador de aula Identificador Curs
--

<b>Professor_Grup</b> Identificador Professor Identificador Grup
--

<b>Aula</b> Identificador Capacitat
---

<b>Alumne_Grup</b> Identificador Alumne Identificador Grup
--

<b>Dies lectius</b> Nom del dia setmanal
---

<b>Adreça_IP</b> Adreça IP Identificador Aula
---

<b>Històric de Horari</b> Data Hora inici Hora fi Dia setmanal Identificador de grup Identificador de aula Identificador Curs Llista Adreces IP vàlides per data i aula de la classe
---

3.2.2 Relacions

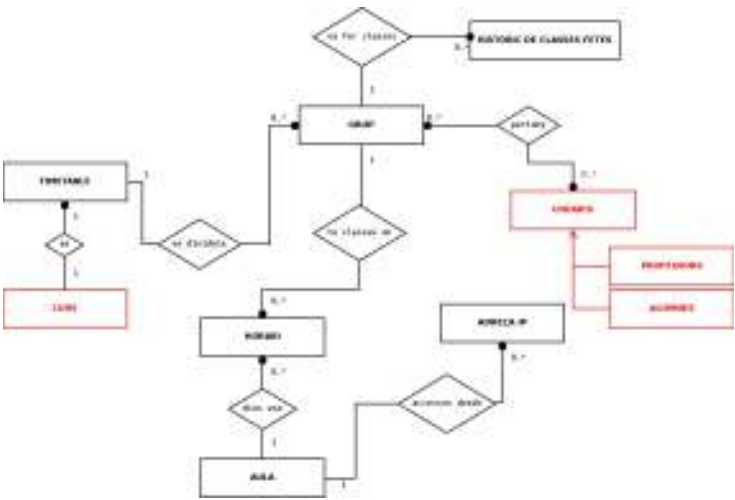


Figura 3.4: Diagrama de relacions entre les entitats. En vermell entitats ja existents a Moodle.

Un grup en un horari, està situat en una aula. A una aula, en un horari determinat només hi haurà com a màxim un grup (o cap). Un grup repetirà a una aula n cops. Cada assignació tindrà una durada, és a dir, tindrà una data d'inici i una data de fi. Es considera que cada grup a una hora determinada i un dia setmanal determinat estarà situat en una aula (que sempre serà la mateixa). Això vol dir, que durant una setmana en diferents dies de la setmana i en una hora determinada d'aquest dia de la setmana un grup estarà a l'aula que te assignada. És a dir, un professor assignarà una aula, i independentment d'aquesta assignació el grup tindrà un horari setmanal. Totes les classes realitzades per cada grup s'aniran enregistrant a un històric de classes per grup per poder creuar aquestes dades amb les dades dels accessos pròpies ja del sistema Moodle. Cada aula té unes adreces IP considerades vàlides, cada aula pot tenir assignades varies adreces IP (o cap), una adreça IP pot ser assignada a moltes aules (o cap). Professor i Alumne i Curs són entitats que són gestionades pel Moodle, s'haurà de veure de quina



manera s'obtenen aquestes dades, però no les gestionarà aquesta aplicació.  
Cada grup es gestiona per algun o alguns professors (o cap). Cada alumne pertany a un grup (o cap).

### 3.3 Esquema bàsic de la base de dades

Diagrama extret de la base de dades, per representar explícitament les taules i els camps necessaris i les relacions entre aquestes taules.

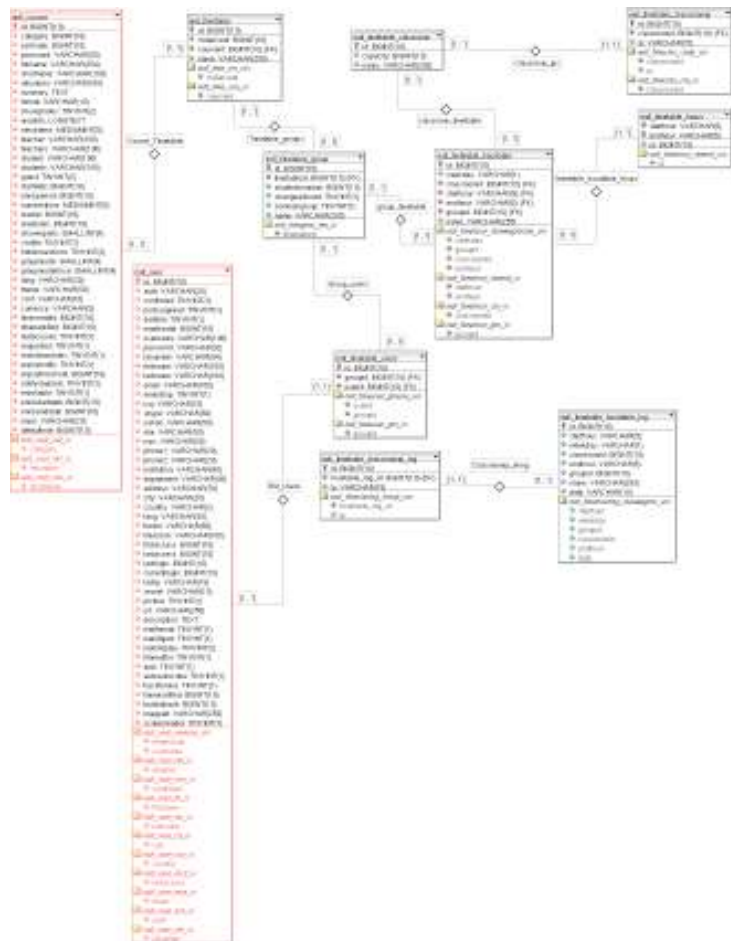


Figura 3.5: Esquema de la base de dades. En vermell taules de Moodle.

### 3.4 Fitxer xmldb per la creació de taules

Tal com s'explica en la secció 4.2.1.1 (pàgina 36) aquest és el fitxer que necessita Moodle per crear les taules necessàries sense haver de creat scripts especialitzats per cadascun dels sistemes gestors de base de dades en els que es vulgui fer compatible el mòdul o bloc.

Es tracta, doncs, d'un fitxer XML que serveix com capa d'abstracció entre el desenvolupament i la base de dades.

Listing 3.1: Fitxer xmldb install.xml per la creació de les taules

```

1 <?xml version="1.0" encoding="UTF-8" ?>
2 <XMLDB PATH="blocks/timetable/db" VERSION="20080423" COMMENT="XMLDB file for
   Moodle blocks/timetable"
3   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4   xsi:noNamespaceSchemaLocation="../../lib/xmldb/xmldb.xsd"
5 >
6 <TABLES>
7   <TABLE NAME="timetable_hours" COMMENT="timetable_hours table retrofitted
   from MySQL" NEXT="timetable_classroom">
8     <FIELDS>
9       <FIELD NAME="id" TYPE="int" LENGTH="10" NOTNULL="true" UNSIGNED="true
   " SEQUENCE="true" ENUM="false" NEXT="starthour"/>
10      <FIELD NAME="starthour" TYPE="char" LENGTH="5" NOTNULL="true"
   SEQUENCE="false" ENUM="false" NEXT="endhour" PREVIOUS="id"
   " />
11      <FIELD NAME="endhour" TYPE="char" LENGTH="5" NOTNULL="true" SEQUENCE=
   "false" ENUM="false" PREVIOUS="starthour"/>
12    </FIELDS>
13    <KEYS>
14      <KEY NAME="primary" TYPE="primary" FIELDS="id" COMMENT="Key
   for id for timetable_hours" NEXT="starthourendhour"/>
15      <KEY NAME="starthourendhour" TYPE="unique" FIELDS="starthour, endhour"
   COMMENT="Primary key for timetable_hours" PREVIOUS="primary" />
16    </KEYS>
17  </TABLE>
18  <TABLE NAME="timetable_classroom" COMMENT="timetable_classroom table
   retrofitted from MySQL" PREVIOUS="timetable_hours" NEXT="timetable">
19    <FIELDS>
20      <FIELD NAME="id" TYPE="int" LENGTH="10" NOTNULL="true" UNSIGNED="true
   " SEQUENCE="true" ENUM="false" NEXT="capacity"/>
21      <FIELD NAME="capacity" TYPE="int" LENGTH="10" NOTNULL="true" UNSIGNED
   ="true" SEQUENCE="false" ENUM="false" PREVIOUS="id" NEXT="name"/>
22      <FIELD NAME="name" TYPE="char" LENGTH="255" NOTNULL="true" SEQUENCE="
   false" ENUM="false" PREVIOUS="capacity"/>
23    </FIELDS>
24    <KEYS>
25      <KEY NAME="primary" TYPE="primary" FIELDS="id" COMMENT="Primary key
   for timetable_classroom" />
26    </KEYS>
27  </TABLE>
28  <TABLE NAME="timetable" COMMENT="timetable table retrofitted from MySQL"
   PREVIOUS="timetable_classroom" NEXT="timetable_group">
29    <FIELDS>
30      <FIELD NAME="id" TYPE="int" LENGTH="10" NOTNULL="true" UNSIGNED="true
   " ENUM="false" SEQUENCE="true" NEXT="instanceid"/>
31      <FIELD NAME="instanceid" TYPE="int" LENGTH="10" NOTNULL="true"
   UNSIGNED="true" ENUM="false" SEQUENCE="false" PREVIOUS="id"
   id" NEXT="courseid"/>
32      <FIELD NAME="courseid" TYPE="int" LENGTH="10" NOTNULL="true" UNSIGNED
   ="false" SEQUENCE="false" ENUM="false" PREVIOUS="instanceid" NEXT=
   "name"/>
33      <FIELD NAME="name" TYPE="char" LENGTH="255" NOTNULL="false" SEQUENCE=
   "false" ENUM="false" PREVIOUS="courseid"/>

```

```

34 </FIELDS>
35 <KEYS>
36   <KEY NAME="primary" TYPE="primary" FIELDS="id" COMMENT="Primary
37     key for timetable" NEXT="timetablecourseid" />
38   <KEY NAME="timetablecourseid" TYPE="foreign" FIELDS="courseid
39     " REFTABLE="course" REFFIELDS="id" PREVIOUS="primary" NEXT
40     ="instanceid" />
41   <KEY NAME="instanceid" TYPE="unique" FIELDS="instanceid"
42     COMMENT="Instance key for timetable" PREVIOUS="
43     timetablecourseid"/>
44 </KEYS>
45 <!-- INDEXES -->
46 <INDEX NAME="courseid" UNIQUE="false" FIELDS="courseid"/>
47 </INDEXES -->
48 </TABLE>
49 <TABLE NAME="timetable_group" COMMENT="timetable_group table retrofitted
50   from MySQL" PREVIOUS="timetable" NEXT="timetable_users">
51   <FIELDS>
52     <FIELD NAME="id" TYPE="int" LENGTH="10" NOTNULL="true" UNSIGNED="true
53       " SEQUENCE="true" ENUM="false" NEXT="timetableid"/>
54     <FIELD NAME="timetableid" TYPE="int" LENGTH="10" NOTNULL="true"
55       UNSIGNED="false" SEQUENCE="false" ENUM="false" PREVIOUS="id" NEXT="
56       studentsnumber"/>
57     <FIELD NAME="studentsnumber" TYPE="int" LENGTH="10" NOTNULL="false"
58       UNSIGNED="true" SEQUENCE="false" ENUM="false" PREVIOUS="
59       timetableid" NEXT="changeallowed"/>
60     <FIELD NAME="changeallowed" TYPE="int" LENGTH="1" NOTNULL="false"
61       UNSIGNED="false" SEQUENCE="false" ENUM="false" PREVIOUS="
62       studentsnumber" NEXT="commongroup"/>
63     <FIELD NAME="commongroup" TYPE="int" LENGTH="1" NOTNULL="false"
64       UNSIGNED="false" SEQUENCE="false" ENUM="false" PREVIOUS="
65       changeallowed" NEXT="name"/>
66     <FIELD NAME="name" TYPE="char" LENGTH="255" NOTNULL="false" SEQUENCE=
67       "false" ENUM="false" PREVIOUS="commongroup"/>
68   </FIELDS>
69   <KEYS>
70     <KEY NAME="primary" TYPE="primary" FIELDS="id" COMMENT="Primary key
71       for timetable_group" NEXT="grouptimetableid"/>
72     <KEY NAME="grouptimetableid" TYPE="foreign" FIELDS="
73       timetableid" REFTABLE="timetable" REFFIELDS="id" PREVIOUS="
74       primary" />
75   </KEYS>
76   <!-- INDEXES -->
77   <INDEX NAME="timetableid" UNIQUE="false" FIELDS="timetableid"/>
78 </INDEXES -->
79 </TABLE>
80 <TABLE NAME="timetable_users" COMMENT="timetable_users table retrofitted
81   from MySQL" PREVIOUS="timetable_group" NEXT="timetable_hourtable">
82   <FIELDS>
83     <FIELD NAME="id" TYPE="int" LENGTH="10" NOTNULL="true" UNSIGNED="true"
84       SEQUENCE="true" ENUM="false" NEXT="groupid"/>
85     <FIELD NAME="groupid" TYPE="int" LENGTH="10" NOTNULL="true" UNSIGNED=
86       "false" SEQUENCE="false" ENUM="false" NEXT="userid" PREVIOUS="id"/
87     >
88     <FIELD NAME="userid" TYPE="int" LENGTH="10" NOTNULL="true" UNSIGNED="
89       false" SEQUENCE="false" ENUM="false" PREVIOUS="groupid"/>
90   </FIELDS>
91   <KEYS>
92     <KEY NAME="primary" TYPE="primary" FIELDS="id" COMMENT="Key
93       for id for timetable_hours" NEXT="groupiduserid"/>
94     <KEY NAME="groupiduserid" TYPE="unique" FIELDS="groupid, userid"
95       COMMENT="Primary key for timetable_users" NEXT="usergroupid"
96       PREVIOUS="primary"/>

```

```

70   <KEY NAME="usergroupid" TYPE="foreign" FIELDS="groupid"
71     REFTABLE="timetable_group" REFFIELDS="id" PREVIOUS="
72     groupiduserid" NEXT="userid" />
73   <KEY NAME="userid" TYPE="foreign" FIELDS="userid"
74     REFTABLE="users" REFFIELDS="id" PREVIOUS="usergroupid" />
75 </KEYS>
76 <!-- INDEXES -->
77 <INDEX NAME="userid" UNIQUE="false" FIELDS="userid"/>
78 </INDEXES -->
79 </TABLE>
80 <TABLE NAME="timetable_hourtable" COMMENT="timetable_hourtable table
81   retrofitted from MySQL" PREVIOUS="timetable_users" NEXT="
82   timetable_classroomip">
83   <FIELDS>
84     <FIELD NAME="id" TYPE="int" LENGTH="10" NOTNULL="true" UNSIGNED="
85       true" SEQUENCE="true" ENUM="false" NEXT="starthour"/>
86     <FIELD NAME="starthour" TYPE="char" LENGTH="5" NOTNULL="true"
87       SEQUENCE="false" ENUM="false" NEXT="weekday" PREVIOUS="id"
88       />
89     <FIELD NAME="weekday" TYPE="char" LENGTH="1" NOTNULL="true" SEQUENCE=
90       "false" ENUM="false" PREVIOUS="starthour" NEXT="classroomid"/>
91     <FIELD NAME="classroomid" TYPE="int" LENGTH="10" NOTNULL="true"
92       UNSIGNED="false" SEQUENCE="false" ENUM="false" PREVIOUS="weekday"
93       NEXT="endhour"/>
94     <FIELD NAME="endhour" TYPE="char" LENGTH="5" NOTNULL="true" SEQUENCE=
95       "false" ENUM="false" PREVIOUS="classroomid" NEXT="groupid"/>
96     <FIELD NAME="groupid" TYPE="int" LENGTH="10" NOTNULL="true" UNSIGNED=
97       "false" SEQUENCE="false" ENUM="false" PREVIOUS="endhour" NEXT="
98       name"/>
99     <FIELD NAME="name" TYPE="char" LENGTH="255" NOTNULL="false" SEQUENCE=
100       "false" ENUM="false" PREVIOUS="groupid"/>
101   </FIELDS>
102   <KEYS>
103     <KEY NAME="primary" TYPE="primary" FIELDS="id" COMMENT="Key
104       for id for timetable_hours" NEXT="swgce"/>
105     <KEY NAME="swgce" TYPE="unique" FIELDS="starthour, weekday, groupid,
106       classroomid, endhour" COMMENT="Primary key for timetable_hourtable"
107       NEXT="starthourendhour" PREVIOUS="primary"/>
108     <KEY NAME="starthourendhour" TYPE="foreign" FIELDS="starthour,
109       endhour" REFTABLE="timetable_hours" REFFIELDS="starthour,
110       endhour" PREVIOUS="swgce" NEXT="hourtableclassroomid"/>
111     <KEY NAME="hourtableclassroomid" TYPE="foreign" FIELDS="
112       classroomid" REFTABLE="timetable_classroom" REFFIELDS="id"
113       PREVIOUS="starthourendhour" NEXT="hourtablegroupid" />
114     <KEY NAME="hourtablegroupid" TYPE="foreign" FIELDS="groupid"
115       REFTABLE="timetable_group" REFFIELDS="id" PREVIOUS="
116       hourtableclassroomid"/>
117   </KEYS>
118   <!-- INDEXES -->
119   <INDEX NAME="classroomid" UNIQUE="false" FIELDS="classroomid" NEXT="
120     starthour"/>
121   <INDEX NAME="starthour" UNIQUE="false" FIELDS="starthour, endhour"
122     PREVIOUS="classroomid" NEXT="groupid"/>
123   <INDEX NAME="groupid" UNIQUE="false" FIELDS="groupid" PREVIOUS="
124     starthour"/>
125 </INDEXES -->
126 </TABLE>
127 <TABLE NAME="timetable_classroomip" COMMENT="timetable_classroomip"
128   PREVIOUS="timetable_hourtable" NEXT="timetable_hourtable_log">
129   <FIELDS>
130     <FIELD NAME="id" TYPE="int" LENGTH="10" NOTNULL="true"
131       UNSIGNED="true" SEQUENCE="true" ENUM="false" NEXT="
132       classroomid"/>

```

```

103      <FIELD NAME="classroomid" TYPE="int" LENGTH="10"
      NOTNULL="true" UNSIGNED="true" SEQUENCE="false"
      ENUM="false" NEXT="ip" PREVIOUS="id"/>
104      <FIELD NAME="ip" TYPE="char" LENGTH="15" NOTNULL="
      true" UNSIGNED="true" SEQUENCE="false" ENUM="false"
      " PREVIOUS="classroomid"/>
105      </FIELDS>
106      <KEYS>
107      <KEY NAME="primary" TYPE="primary" FIELDS="id"
      COMMENT="Key for id for timetable_hours" NEXT="
      classroomidip"/>
108      <KEY NAME="classroomidip" TYPE="unique" FIELDS="
      classroomid,ip" NEXT="classroomipclassroom"
      PREVIOUS="primary"/>
109      <KEY NAME="classroomipclassroom" TYPE="foreign"
      FIELDS="classroomid" REFTABLE="timetable_classroom"
      " REFFIELDS="id" PREVIOUS="classroomidip" />
110      </KEYS>
111      </TABLE>
112      <TABLE NAME="timetable_hourtable_log" COMMENT="
      timetable_hourtable_log" PREVIOUS="timetable_classroomip" NEXT="
      timetable_classroomip_log" >
113      <FIELDS>
114      <FIELD NAME="id" TYPE="int" LENGTH="10" NOTNULL="true"
      " UNSIGNED="true" SEQUENCE="true" ENUM="false"
      NEXT="starthour" />
115      <FIELD NAME="starthour" TYPE="char" LENGTH="5"
      NOTNULL="true" SEQUENCE="false" ENUM="false" NEXT="
      "weekday" PREVIOUS="id" />
116      <FIELD NAME="weekday" TYPE="char" LENGTH="1" NOTNULL="true" SEQUENCE="
      "false" ENUM="false" PREVIOUS="starthour" NEXT="classroomid"/>
117      <FIELD NAME="classroomid" TYPE="int" LENGTH="10" NOTNULL="true"
      UNSIGNED="false" SEQUENCE="false" ENUM="false" PREVIOUS="weekday"
      NEXT="endhour"/>
118      <FIELD NAME="endhour" TYPE="char" LENGTH="5" NOTNULL="true" SEQUENCE="
      "false" ENUM="false" PREVIOUS="classroomid" NEXT="groupid"/>
119      <FIELD NAME="groupid" TYPE="int" LENGTH="10" NOTNULL="true" UNSIGNED="
      "false" SEQUENCE="false" ENUM="false" PREVIOUS="endhour" NEXT="
      name"/>
120      <FIELD NAME="name" TYPE="char" LENGTH="255" NOTNULL="false" SEQUENCE="
      "false" ENUM="false" PREVIOUS="groupid" NEXT="data"/>
121      <FIELD NAME="data" TYPE="char" LENGTH="10" NOTNULL="true"
      SEQUENCE="false" ENUM="false" PREVIOUS="name" />
122      </FIELDS>
123      <KEYS>
124      <KEY NAME="primary" TYPE="primary" FIELDS="id"
      " COMMENT="Key for id for timetable_hours"
      NEXT="swgced"/>
125      <KEY NAME="swgced" TYPE="unique" FIELDS="
      starthour, weekday, groupid, classroomid,
      endhour,data" COMMENT="Primary key for
      timetable_hourtable" PREVIOUS="primary"/>
126      </KEYS>
127      </TABLE>
128      <TABLE NAME="timetable_classroomip_log" COMMENT="
      timetable_classroomip_log" PREVIOUS="timetable_hourtable_log">
129      <FIELDS>
130      <FIELD NAME="id" TYPE="int" LENGTH="10" NOTNULL="true"
      " UNSIGNED="true" SEQUENCE="true" ENUM="false"
      NEXT="hourtable_log_id"/>
131      <FIELD NAME="hourtable_log_id" TYPE="int" LENGTH="10"
      NOTNULL="true" UNSIGNED="false" SEQUENCE="false"
      ENUM="false" PREVIOUS="id" NEXT="ip"/>

```

```

132      <FIELD NAME="ip" TYPE="char" LENGTH="15" NOTNULL="
      true" UNSIGNED="true" SEQUENCE="false" ENUM="false"
      " PREVIOUS="hourtable_log_id"/>
133      </FIELDS>
134      <KEYS>
135      <KEY NAME="primary" TYPE="primary" FIELDS="id"
      " COMMENT="Key for id for timetable_hours"
      NEXT="hliip"/>
136      <KEY NAME="hliip" TYPE="unique" FIELDS="
      hourtable_log_id,ip" COMMENT="Primary key
      for timetable_classroom" PREVIOUS="
      primary"/>
137      </KEYS>
138      </TABLE>
139      </TABLES>
140      </XMLDB>

```

### 3.5 Diagrama d'estats

La manera d'interaccionar de l'usuari dependrà del seu rol i segons aquest les operacions que podrà realitzar cada usuari seran diferents. Per tant podem interpretar el següent diagrama (Figura 3.6) com a visió global del funcionament del sistema.

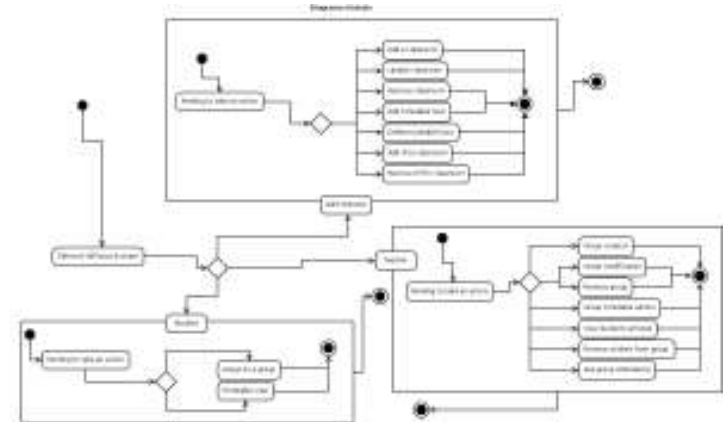


Figura 3.6: Diagrama de estats del sistema.

### 3.6 Interfície d'usuari

Al capítol sobre el Manual de l'usuari d'aquesta documentació es trobaran totes i cadascuna de les pantalles finals del sistema detallades.

## Capítol 4

# Evolució del projecte i decisions de disseny

En aquest capítol es descriu la evolució que ha tingut el projecte així com també es descriuen els problemes que s'han anat trobant i com s'han resolt. Per comparar es mostren també dos diagrames de Gantt, un amb la previsió i idea inicial de com es repartiria la feina del projecte i un de final amb el que ha estat realment la evolució d'aquest amb les variacions que els problemes han provocat.

### 4.1 Previsió de l'evolució del projecte

Inicialment es tenia prevista una evolució del projecte com segueix (Figura 4.1).

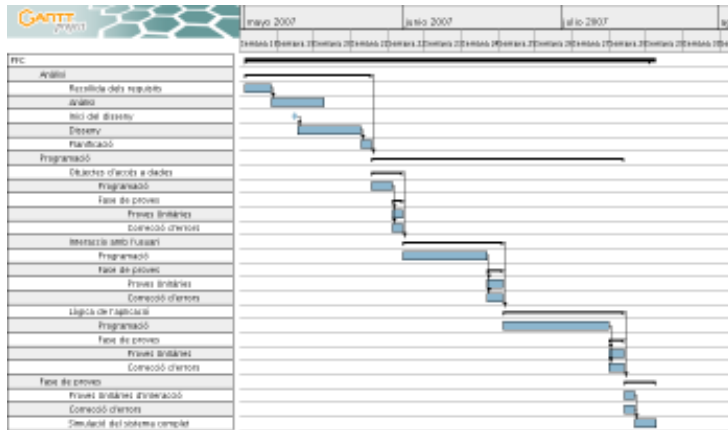


Figura 4.1: Previsió de l'evolució del projecte. Les unitats representades per les columnes són setmanes de treball.

### 4.2 Evolució del projecte

Partint de l'idea inicial del projecte, en aquesta secció presentem l'evolució real que ha tingut el projecte, els problemes trobats i les resolucions d'aquests problemes.

#### 4.2.1 Anàlisi

En aquest moment del projecte es van plasmar els requisits essencials del projecte com eren la gestió de les aules, de les hores lectives, dels grups i dels horaris d'aquests grups per a cada curs. Realment senzills si s'observa la primera revisió de l'especificació funcional.

Es va tenir que decidir, i per tant entendre, la diferència entre Mòdul d'activitat i Block de Moodle per determinar quin dels dos tipus de plugins s'hauria de implementar. Es pot veure la definició en la secció d'introducció on s'intenten definir tots dos.

Bàsicament la diferència rau en què els mòduls d'activitat serveixen per interactuar amb els estudiants mitjançant activitats a nivell de temari que han de realitzar mitjançant les quals els professors poden avaluar o fer un seguiment dels coneixements assolits pels alumnes mentre que els blocs serveixen per mostrar informació adicional sobre els cursos o l'entorn i per tant es va escollir el Block per idoneïtat amb l'ús. Per poder començar i tenir una guia per poder començar amb el desenvolupament del block es va seguir la guia de desenvolupament que podem trobar a la següent adreça <http://docs.moodle.org/en/Development:Blocks>.

Els següents subpunts defineixen com es van abordar i com s'implementen certes solucions mitjançant les eines que aporten la plataforma Moodle.

##### 4.2.1.1 Base de dades

En versions anteriors a la 1.7 la creació de les taules en bases de dades es feia mitjançant scripts SQL, tants scripts com per Sistemes Gestors de Bases de Dades es volia fer compatible el nou mòdul. A partir de la versió 1.7 es fa mitjançant fitxers xmldb, uns fitxers xml pensats per la creació de bases de dades (Veure secció 9.1, pàgina 77). Es redueix llavors la creació de les taules de la base de dades a un sol fitxer que el sistema Moodle interpreta per cadascuna de les bases de dades amb les que pot treballar.

Listing 4.1: Sentències SQL equivalents al fitxer xmldb per la creació de les taules

```
1 CREATE TABLE mdl_timetable (
2   id bigint(10) unsigned NOT NULL auto_increment ,
3   instanceid bigint(10) unsigned NOT NULL ,
4   courseid bigint(10) NOT NULL ,
5   'name' varchar(255) default NULL ,
6   PRIMARY KEY (id),
7   UNIQUE KEY mdl_time_ins_uix (instanceid),
8   KEY mdl_time_cou_ix (courseid)
9 );
10
11
12 CREATE TABLE mdl_timetable_classroom (
13   id bigint(10) unsigned NOT NULL auto_increment ,
14   capacity bigint(10) unsigned NOT NULL ,
15   'name' varchar(255) NOT NULL default ' ',
16   PRIMARY KEY (id)
17 );
18
19
20 CREATE TABLE mdl_timetable_classroomip (
21   id bigint(10) unsigned NOT NULL auto_increment ,
22   classroomid bigint(10) unsigned NOT NULL ,
23   ip varchar(15) NOT NULL ,
24   PRIMARY KEY (id),
25   UNIQUE KEY mdl_timeclas_claip_uix (classroomid,ip),
26   KEY mdl_timeclas_claix (classroomid)
27 );
28
29
30 CREATE TABLE mdl_timetable_classroomip_log (
31   id bigint(10) unsigned NOT NULL auto_increment ,
32   hourtable_log_id bigint(10) NOT NULL ,
```

```

33 ip varchar(15) NOT NULL,
34 PRIMARY KEY (id),
35 UNIQUE KEY mdl_timeclaslog_houip_uix (hourtable_log_id,ip)
36 );
37
38
39 CREATE TABLE mdl_timetable_group (
40 id bigint(10) unsigned NOT NULL auto_increment,
41 timetableid bigint(10) NOT NULL,
42 studentsnumber bigint(10) unsigned default NULL,
43 changeallowed tinyint(1) default NULL,
44 commongroup tinyint(1) default NULL,
45 'name' varchar(255) default NULL,
46 PRIMARY KEY (id),
47 KEY mdl_timegrou_tim_ix (timetableid)
48 );
49
50
51 CREATE TABLE mdl_timetable_hours (
52 id bigint(10) unsigned NOT NULL auto_increment,
53 starthour varchar(5) NOT NULL,
54 endhour varchar(5) NOT NULL,
55 PRIMARY KEY (id),
56 UNIQUE KEY mdl_timehour_staend_uix (starthour, endhour)
57 );
58
59
60 CREATE TABLE mdl_timetable_hourtable (
61 id bigint(10) unsigned NOT NULL auto_increment,
62 starthour varchar(5) NOT NULL,
63 weekday varchar(1) NOT NULL,
64 classroomid bigint(10) NOT NULL,
65 endhour varchar(5) NOT NULL,
66 groupid bigint(10) NOT NULL,
67 'name' varchar(255) default NULL,
68 PRIMARY KEY (id),
69 UNIQUE KEY mdl_timehour_staweegroclae_uix (starthour, weekday, groupid,
70 classroomid, endhour),
71 KEY mdl_timehour_staend_ix (starthour, endhour),
72 KEY mdl_timehour_cla_ix (classroomid),
73 KEY mdl_timehour_gro_ix (groupid)
74 );
75
76 CREATE TABLE mdl_timetable_hourtable_log (
77 id bigint(10) unsigned NOT NULL auto_increment,
78 starthour varchar(5) NOT NULL,
79 weekday varchar(1) NOT NULL,
80 classroomid bigint(10) NOT NULL,
81 endhour varchar(5) NOT NULL,
82 groupid bigint(10) NOT NULL,
83 'name' varchar(255) default NULL,
84 'data' varchar(10) NOT NULL,
85 PRIMARY KEY (id),
86 UNIQUE KEY mdl_timehourlog_staweegroc_uix (starthour, weekday, groupid,
87 classroomid, endhour, 'data')
88 );
89
90 CREATE TABLE mdl_timetable_users (
91 id bigint(10) unsigned NOT NULL auto_increment,
92 groupid bigint(10) NOT NULL,
93 userid bigint(10) NOT NULL,

```

```

94 PRIMARY KEY (id),
95 UNIQUE KEY mdl_timeuser_grouse_uix (groupid,userid),
96 KEY mdl_timeuser_gro_ix (groupid),
97 KEY mdl_timeuser_use_ix (userid)
98 );

```

Les versions dels fitxers xmldb equivalents a aquest script SQL de creació de taules els podem trobar a ambdues especificacions funcionals (secció 2.4, pàgina 16 i secció 3.4, pàgina 30 )

#### 4.2.1.2 Rols d'usuari

El identificador d'usuari i l'identificador de course s'obtin-drà de Moodle, de les taules d'usuari i de curs respectivament. Obtenció de la informació dels usuaris A tenir en compte els camps següents (extret de la pròpia base de dades):

mdl_user	
Camp	Tipus
	(...)
id	bigint(10)
	(...)
username	varchar(100)
firstname	varchar(100)
lastname	varchar(100)
	(...)

Els valors dels usuaris s'obtenen a partir de una de les variables globals de Moodle \$USER (Veure secció 9.2, pàgina 78 )

El model d'interacció dependrà del tipus d'usuari, que Moodle determina a través dels rols. Per determinar en cada moment quin tipus d'usuari és el que entra al sistema es pot consultar el seu rol mitjançant les funcions:

```

isadmin()
isteacher()
isstudent()

```

Un exemple de la seva utilització:

Listing 4.2: Exemple de comprovació de rol d'usuari en versions de Moodle anteriors a la 1.7

```

1 require_login($courseid);
2 if (isteacher($courseid) && confirm_sesskey($sesskey))
3 {
4 /*Accions (validació, confirmació, guardar a bbdd, etc., depenent de la pàgina
   que sigui)*/
5 }else{
6 /*Redirigeixo a pàgina del curs*/
7 }

```

Això però es feia així en versions de Moodle anteriors a la 1.7. Per permetre una millor flexibilitat en la definició dels rols això ha canviat i en la documentació d'aquesta nova versió s'especifica un nou funcionament per definir funcionalitats depenent del rol (Veure secció 9.3, pàgina 79 ). Així, ara si es vol definir una funcionalitat accessible o no depenent del rol es defineix el nom de la funcionalitat i per cadascun dels rols el nivell de seguretat que se li vol donar i al codificar es farà ús de la funció *has\_capability*

Aquí hi ha un exemple de la nova manera d'obtenir els rols per un millor enteniment:

Listing 4.3: Fitxer access.db

```

1 <?php
2
3 $mod_prueba_capabilities = array(
4

```

```

5  'mod/prueba:mostrar_hola_profesor' => array (
6
7  'captype' => 'write',
8  'contextlevel' => CONTEXT_MODULE,
9  'legacy' => array (
10 'editingteacher' => CAP_PROHIBIT,
11 'coursecreator' => CAP_PROHIBIT,
12 'teacher' => CAP_ALLOW,
13 'student' => CAP_PROHIBIT,
14 'admin' => CAP_PROHIBIT
15 )
16 ),
17
18 'mod/prueba:mostrar_hola_alumno' => array (
19
20 'captype' => 'write',
21 'contextlevel' => CONTEXT_MODULE,
22 'legacy' => array (
23 'editingteacher' => CAP_PROHIBIT,
24 'coursecreator' => CAP_PROHIBIT,
25 'admin' => CAP_ALLOW,
26 'student' => CAP_PROHIBIT
27 )
28 )
29 );
30
31
32 ?>

```

Listing 4.4: Fitxer view.php

```

1  (...)
2  /// Print the main part of the page
3
4  $context = get_context_instance(CONTEXT_MODULE, $cm->id);
5
6  if (has_capability('mod/prueba:mostrar_hola_profesor', $context)) {
7  echo 'Hola profesor';
8  }
9  else if (has_capability('mod/prueba:mostrar_hola_alumno', $context)) {
10 echo 'Hola alumno';
11 }

```

#### 4.2.1.3 Interfície d'usuari

Degut a que depenent dels rols assignats a l'usuari pot realitzar diferents funcionalitats del bloc, s'han implementat les pantalles mitjançant un sistema de pestanyes, on en cadascuna d'elles s'hi pot realitzar la funcionalitat corresponent. Per cada usuari només es mostraran les pestanyes les quals permeten l'accés a la funcionalitat aplicada al rol.

En la primera revisió de l'especificació funcional hi ha un apartat per la interfície d'usuari on hi han uns primers esboços del que es volia presentar per a cada pantalla.

En la revisió final de l'especificació funcional no hi ha un apartat concret per a la interfície d'usuari ja que en el capítol dedicat al manual de l'usuari es detallen totes i cadascuna de les pantalles que s'han implementat finalment.

#### 4.2.1.4 Disseny. Model implementat.

Per evitar que el codi es dividís únicament en fitxers .php, on cadascun d'ells tingués tota la lògica i els elements de representació per pantalla mitjançant un codi poc estructurat i que fós difícil de seguir

s'ha dissenyat un sistema que permet simular un senzill sistema amb el patró Model-Vista-Controlador (Figura 4.2) dins la estructura que Moodle obliga a seguir per implementar blocks.

Cadascuna de les accions que es poden prendre tenen un nom únic que la diferencia de la resta. Aquest mateix nom serà el que tindràn els fitxers .php que realment realitzen les operacions per poder executar l'acció, i el nom del fitxer .html que finalment es qui mostra els resultats.

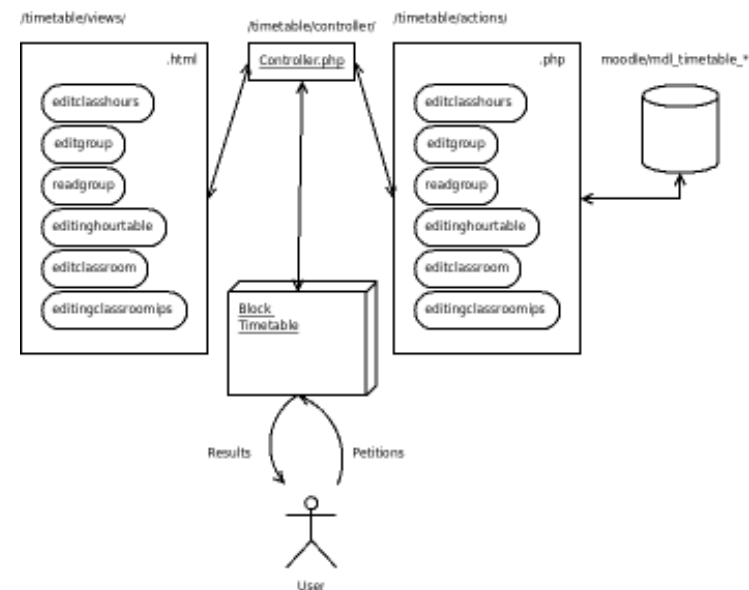


Figura 4.2: Esquema del model d'implementació.

#### 4.2.1.5 Control de l'assistència

Es va concretar posteriorment que faria falta també una opció per al professorat per poder controlar l'assistència dels alumnes a les classes.

El sistema Moodle ja emmagatzema les dades d'accés a l'entorn en una taula anomenada mdl\_log, on es guarda l'usuari, el moment en que realitza l'acció, l'acció que ha realitzat i l'adreça IP desde que s'ha realitzat. Per tant només faria falta crear aquestes dades amb les hores de classe de cada grup, per mostrar per cadascun d'ells si l'accés s'ha produït desde una adreça IP de les quals l'administrador sap que serà una adreça de l'aula en que es realitza aquella hora de classe.

El problema que hi ha és que les classes són elements vius que es poden canviar en qualsevol moment del curs de manera que la taula on s'emmagatzemen les dades de les classes per cada grup no serveix per crear les dades amb els registres de la taula mdl\_log. Per tant, s'ha d'utilitzar una taula (timetable\_hourtable\_log) per enregistrar l'històric de les classes que ja s'han efectuat, ja que si hi ha programada una classe per algun dia concret de la setmana i just abans es canvia aquesta no ha d'apareixer a la taula històrica perquè aquesta classe no s'ha arribat a solucionar.

Per cadascuna de les classes emmagatzemades en l'històric també ens interessa saber quines adreces IP eren vàlides en aquell moment per aquella aula, ja que també poden ser modificades. Això ho emmagatzemarem a la taula (timetable\_classroomip\_log).

La solució que s'ha emprat es utilitzant un recurs que ens facilita Moodle <http://docs.moodle.org/es/Cron>.

Quan s'implementen els mòduls o els blocs hi ha l'opció d'implementar una funció que serà executada

quan es faci una crida a `http://servidorexemple.com/moodle/admin/cron.php`. Aquesta funció és en el cas dels mòduls una funció `nommodul_cron()` dins el fitxer `nommodul/lib.php` i en el cas dels blocks una funció `cron()` dins el fitxer `nomblock/block_nomblock.php`. Es pot consultar això dins la documentació de Moodle a la següent adreça `http://docs.moodle.org/en/Cron#Script_overview`. Per tant, tal com diu la documentació de Moodle, s'haurà de programar una visita al script del Cron però si es vol que realment funcioni aquest script per emmagatzemar les classes realitzades correctament, és molt important que s'assegui que aquest script s'executa almenys un cop durant el moment en que s'estigui realitzant la classe.

#### 4.2.1.6 Flexibilitat amb la configuració del calendari de Moodle

Moodle permet configurar el primer dia de la setmana i els dies que es consideren cap de setmana desde la secció calendari, a l'apartat Aparència de l'Administració del lloc (Figura 4.3).



Figura 4.3: Administració del lloc, Aparència, Calendari.

Aquest block està construït de manera que sempre consulta amb Moodle aquestes dades mitjançant les facilitats que ens proporciona la llibreria `calendar`.

Listing 4.5: Importació de la llibreria PHP necessària per manejar les dades del calendari.

```
1 require_once($CFG->dirroot.'/calendar/lib.php');
```

Per tant qualsevol canvi en qualsevol moment serà considerat a l'hora de representar els horaris dels alumnes.

En el codi s'utilitza la següent iteració per recorre els dies de la setmana i treballar amb ells amb l'ordre que s'ha establert a la configuració de Moodle.

Listing 4.6: Iteració per recorre i processar els dies de la setmana en l'ordre definit per la configuració.

```
1 for($i = $display->minwday; $i <= $display->maxwday; ++$i)
```

Dins la llibreria del `calendar` aquest objecte `$display` defineix els seus atributs `minwday` i `maxwday` com segueix:

Listing 4.7: `calendar/lib.php` Definició del objecte `display`.

```
1 $display = &New stdClass;
2 $display->minwday = get_user_preferences('calendar_startwday',
   CALENDAR_STARTING_WEEKDAY);
3 $display->maxwday = $display->minwday + 6;
```

Per comprovar si es tracta d'un dia de cap de setmana s'utilitza la següent instrucció

Listing 4.8: Iteració per recorre i processar els dies de la setmana en l'ordre definit per la configuració.

```
1 if (CALENDAR_WEEKEND & (1 << ($i % 7)))
```

Aquesta constant `CALENDAR_WEEKEND` es defineix així:

Listing 4.9: `calendar/lib.php` Definició de la constant `CALENDAR_WEEKEND`.

```
1 // This is a packed bitfield: day X is "weekend" if $field & (1 << X) is true
2 // Default value = 65 = 64 + 1 = 2^6 + 2^0 = Saturday & Sunday
3 define('CALENDAR_DEFAULT_WEEKEND', 65);
```

I per obtenir el literal associat al valor del dia setmanal s'utilitza la crida habitual per obtenir literals el fitxer de llenguatge apropiat.

Listing 4.10: Crida per obtenir els literals per als noms dels dies de la setmana

```
1 get_string(CALENDAR_DAYS[$i % 7], 'calendar')
```

#### 4.2.1.7 Internacionalització i ajuda

Timetable ha estat traduït o almenys s'ha intentat que els literals i els fitxers d'ajuda siguin comprensibles en els següents idiomes: català, castellà, anglès i francès.

Moodle està pensat per poder ser utilitzat i traduït al màxim nombre de llengües possibles.

Qualsevol persona pot col·laborar en aquestes traduccions o fer petites modificacions per a ús local tal i com s'indica a la següent adreça `http://docs.moodle.org/ca/Traducci%C3%B3`.

En aquesta pàgina també indica com, a banda de col·laborar amb les traduccions en paquets de llenguatge generals de Moodle, col·laborar també amb les traduccions de mòduls i plugins de tercers. Aquestes instruccions serveixen per crear també els fitxers d'idioma per a nous mòduls i plugins.

Al mateix temps, però, s'han afegit també uns fitxers d'ajuda en format html i les seves traduccions. Per què siguin accessibles amb les mateixes condicions que la resta de fitxers d'ajuda del sistema, aquest contingut s'ha de copiar a la carpeta `lang/XX_utf8/help/nomdelbloc/` però de Moodle en comptes de la pròpia del component. on `XX_utf8` és el llenguatge escollit per la traducció i `nomdel-bloc` és el nom del bloc, en el nostre cas `ca`, es, en i fr per els idiomes i `timetable` com a nom. Copiant aquests fitxers d'ajuda a la ruta especificada ens permet poder cridar-los amb la funció `helpbutton`, mostrant-se amb l'idioma corresponent.

Listing 4.11: Crida als fitxers d'ajuda on `$currenttab` és el nom del fitxer.

```
1 helpbutton($currenttab, get_string('help', 'block_timetable'), "timetable",
   true, true);
```

Aquesta instrucció obre la informació de l'ajuda en una finestra apart, què és el comportament habitual a l'hora de mostrar l'ajuda dins de Moodle.

#### 4.2.1.8 Versió de Moodle

Al moment de començar aquest projecte no es volia fixar una versió mínima de Moodle però les millores i la solvència d'alguns defectes, gràcies a les col·laboracions dels desenvolupadors en algunes de les

progressives versions versions han obligat a fixar una versió mínima. En un principi és va començar a desenvolupar amb la versió 1.6 de Moodle. A l'hora de consultar informació de com crear les taules de la base de dades, es va consultar la informació existent sobre la creació del fitxer xmldb () (Secció 4.2.1.1, pàgina 36 ). Es va escollir que la creació de la base de dades es faria amb aquest fitxer. També, al consultar com es gestionen els rols a Moodle. És va trobar la documentació referent al tema de la versió 1.7 en la que comenta que s'utilitza un fitxer php per gestionar els permisos dels rols d'usuari (Secció 4.2.1.2, pàgina 38). Al escollir aquesta manera de gestionar aquests factors es va fixar per el moment la versió 1.7 de Moodle. Més endavant, durant el desenvolupament del projecte es va trobar un problema amb el registre d'usuaris, adreça IP i moment en que es realitzen les accions sobre el sistema que es va solucionar a la versió 1.8 de Moodle (Secció 4.2.3.2, pàgina 44). Aquest problema afectava directament el projecte i es va forçar a versió mínima la 1.8 de Moodle.

4.2.2 Programació

4.2.2.1 Objectes d'accès a dades

En un principi es va pensar que s'hauria d'implementar una capa per facilitar l'accés a les dades, mitjançant unes classes que accedirien a la base de dades per recuperar cadascuna de les instàncies necessàries per gestionar les dades. Moodle en aquest cas ens facilita les coses i ja incorpora les funcions necessàries per recuperar dades de la base de dades i mapejar-les sobre objectes amb camps corresponents a les columnes amb el mateix nom. En cas de recuperar de la base de dades un conjunt de registres i no només un de sol aquestes funcions ja recuperen una array d'aquests objectes. Dins la llibreria dmlib.php podem trobar totes les funcions necessàries per poder accedir a la base de dades d'una manera transparent al usuari (Veure secció 9.4, pàgina 80 ).

4.2.2.2 Interacció amb l'usuari i Lògica de l'aplicació

A l'hora de seguir amb la implementació de l'aplicació es va canviar el plantejament inicial realitzat en la previsió inicial (Figura 4.1) del desenvolupament del projecte de implementar primer tota la part d'interacció amb l'usuari i després tota la lògica de la aplicació per el desenvolupament de cadascuna de les accions que es poden realitzar. És a dir, en comptes de subdividir la tasca de la programació amb subtasques per la lògica i subtasques per l'interfície es va optar per subdividir-ho amb subtasques per cadascuna de les accions que es poden prendre dins el sistema. Tal com mostra el diagrama (Figura 4.4) de la següent manera:

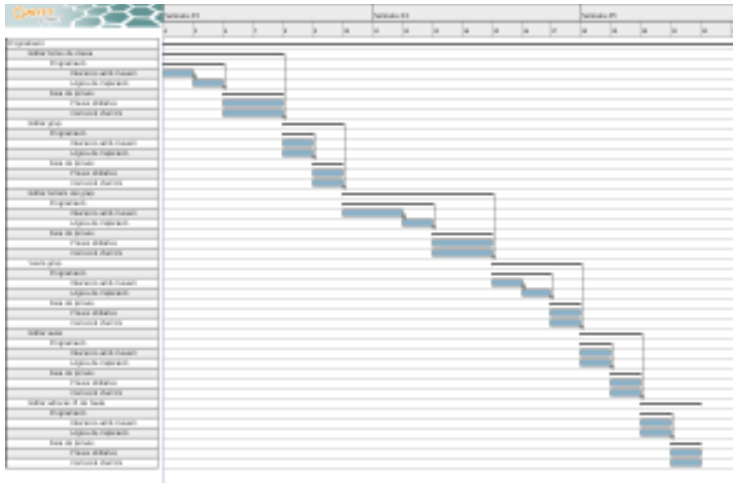


Figura 4.4: Evolució en el desenvolupament de la programació del projecte.

4.2.3 Fase de proves. Problemes i resolució.

Tant en la fase de proves (fase final de la planificació), com en cadascuna de les subfases de Proves unitàries dins el desenvolupament, a part d'errors comuns de programació, s'han anat trobat diferent problemes, i s'han solucionat tal com s'esmenta en els apartats següents.

4.2.3.1 Taules amb camp Id

Un dels primers problemes en que varem trobar es que algunes de les taules com ara mdl\_timetable\_hourttable, on es guarden les hores de classe per cada dia de la setmana per cada grup, o la taula mdl\_timetable\_hours, on es guarden únicament les hores en que es fa classe, o la taula mdl\_timetable\_classroomip, on es guarda la relació entre les adreces IP vàlides i les aules corresponents; s'havien dissenyat sense un camp concret que ens servís d'identificador de clau primària de la taula ja que no era un camp sol el que determinaria la clau primària si no un conjunt (hora d'inici i hora de la setmana, identificador del grup i identificador de l'aula per mdl\_timetable\_hourttable, en mdl\_timetable\_hours únicament es guarda l'hora d'inici i la hora de fi per cada possible classe i la relació entre adreces IP i aules només hauria de guardar l'adreça IP i l'identificador de l'aula). Això suposa un problema si s'utilitzen les funcions que dona l'entorn ja que es suposa que totes les taules han de tenir un camp anomenat id, que serveixi de clau primària. Es pot trobar una discussió sobre el tema als forums de Moodle: <http://moodle.org/mod/forum/discuss.php?d=38183#p176751> <http://docs.moodle.org/en/IdColumnReasons> (Veure secció 9.5, pàgina 82 ) En els debat sobre el tema es fa referència a un paràmetre dins la funció per per indicar quina és el camp clau de la taula. Però tal com es pot observar en la pròpia definició d'aquesta funció és un parametre "obsolet" i n'ignora el seu ús.

```
insert_record($table, $dataobject, $returnid=true,
$primarykey='id') X-Ref
Insert a record into a table and return the "id" field if
required

If the return ID isn't required, then this just reports success
as true/false.
$dataobject is an object containing needed data

param: string $table The database table to be checked against.
param: object $dataobject A data object with values for one or
more fields in the record
param: bool $returnid Should the id of the newly created
record entry be returned? If this option is not requested then
true/false is returned.
param: string $primarykey (obsolete) This is now forced to be
'id'.
```

Per tant s'ha hagut d'afegir a aquestes taules un camp anomenat id i declarar-lo com a clau primària.

4.2.3.2 Funció time() i zones horàries

Per el desenvolupament del control de l'assistència es va crear un sistema que crea les dades de les classes realitzades amb les dades que ja emmagatzema Moodle a la taula mdl\_log. Moodle guarda en aquesta taula la informació de les accions que es prenen sobre Moodle, l'adreça IP i el moment, un timestamp, en que es realitzen. En les primeres proves això es va provar en un servidor montat en la mateixa màquina on es desenvolupava i es provava i no va suposar cap problema. Però més endavant és va provar en un servidor Moodle montat sobre un servidor de hosting gratuït situat a Nova York, mentre que la configuració de Moodle constava que el servidor havia de mostrar l' hora de fus hoari GMT +1. Aquesta diferència interferia en el camp time que s'emmagatzema a la taula mdl\_log, ja que la instrucció que obtenia el timestamp i l'emmagatzemava era errònia. Aquest error i la seva resolució es pot veure com va ser discutit a la següent adreça <http://tracker.moodle.org/browse/MDL-13959> on es pot observar a part de l'error comentat com aclareix com s'han d'utilitzar les funcions per obtenir un timestamp correcte per al moment necessari (Veure secció 9.6,



pàgina 83 ).

La resolució del problema amb la instrucció `usergetdate()` i els consells sobre la seva utilització ens porten a utilitzar-la com segueix:

```
1 $now3 = usergetdate(time());
```

Aquesta instrucció ens serveix per obtenir l'hora del usuari administrador, que per una correcta execució no hauria de ser diferent de la configuració de Moodle.

### 4.2.3.3 Visualització del registre d'assistència

A la pantalla de visualització del control d'assistència dels alumnes per un grup, si els registres retornats de les classes realitzades per aquell grup eren una quantitat força alta, és a dir, que per aquell grup i les dates seleccionades hi havien força classes realitzades, les dades marxaven de la finestra, obligant a l'usuari a utilitzar la barra de desplaçament horitzontal del navegador web, perdent de vista així la resta de dades del grup, com ara la llista de alumnes o les dates en que s'ha fet la consulta. Per evitar això, per la representació de les dades s'ha obtat per jugar amb els estils de les capes dins de l'html forçant així que les dades si ocupen més espai que el reservat amb el `<div>`, aquesta capa mostri la barra de desplaçament i no sigui el navegador web el que la mostri.

Listing 4.12: codi de `readgroup.html` on es juga amb els estils de les capes per mostrar la barra de desplaçament horitzontal.

```
1 <div id="tablaLogs" >
2   <div id="Layer1" style="position:relative;float:left;">
3     <!-- Pinta una taula amb la llista d'usuaris-->
4   </div>
5
6   <div id="Layer1" style="position:relative;width:80%;overflow:auto;
7     float:left;">
8     <!-- Pinta la taula de la llista de classes, on les files
9       seran els
10      accessos dels usuaris amb IP i les columnes les
        classes realitzades-->
9   </div>
10 </div> <!--TablaLogs"-->
```

## Capítol 5

# Guia d'instal·lació

El procés d'instal·lació és l'habitual dels blocs de Moodle:  
<http://moodle.org/mod/forum/discuss.php?d=38783>,  
[http://docs.moodle.org/en/Installing\\_contributed\\_modules\\_or\\_plugins](http://docs.moodle.org/en/Installing_contributed_modules_or_plugins).  
S'haurà de copiar la carpeta `timetable` dins la carpeta `blocks` allà on correspongui, segons la instal·lació de Moodle. S'haurà d'accedir com l'usuari administrador del lloc (normalment `admin`) e instal·lar el bloc clicant a Notificacions dins el bloc d'Administració del lloc.  
Posteriorment es mostrarà la informació corresponent a la correcta creació de les taules, tal com es mostra a les següents línies:

Listing 5.1: Resultat de la instal·lació del Block dins al sistema Moodle

```
1 No warnings - Scroll to the continue button
2 timetable
3
4 (mysql): SHOW TABLES
5
6 (mysql): SHOW TABLES
7
8 (mysql): SHOW TABLES
9
10 (mysql): SHOW TABLES
11
12 (mysql): SHOW TABLES
13
14 (mysql): SHOW TABLES
15
16 (mysql): SHOW TABLES
17
18 (mysql): SHOW TABLES
19
20 (mysql): SHOW TABLES
21
22 (mysql): SHOW TABLES
23
24 (mysql): SHOW TABLES
25
26 (mysql): SHOW TABLES
27
28 (mysql): SHOW TABLES
29
30 (mysql): SHOW TABLES
31
32 (mysql): SHOW TABLES
33
34 (mysql): CREATE TABLE mdl_timetable_hours ( id BIGINT(10) unsigned NOT NULL
      auto_increment, starthour VARCHAR(5) NOT NULL DEFAULT '', endhour VARCHAR
      (5) NOT NULL DEFAULT '', CONSTRAINT PRIMARY KEY (id) )
```

```

35  Òxit
36
37  (mysql): ALTER TABLE mdl_timetable_hours COMMENT='timetable_hours table
38  retrofitted from MySQL'
39  Òxit
40
41  (mysql): CREATE UNIQUE INDEX mdl_timehour_staend_uix ON mdl_timetable_hours (
42  starthour, endhour)
43  Òxit
44
45  (mysql): CREATE TABLE mdl_timetable_classroom ( id BIGINT(10) unsigned NOT
46  NULL auto_increment, capacity BIGINT(10) unsigned NOT NULL, name VARCHAR
47  (255) NOT NULL DEFAULT '', CONSTRAINT PRIMARY KEY (id) )
48  Òxit
49
50  (mysql): ALTER TABLE mdl_timetable_classroom COMMENT='timetable_classroom
51  table retrofitted from MySQL'
52  Òxit
53
54  (mysql): CREATE INDEX mdl_time_cou_ix ON mdl_timetable (courseid)
55  Òxit
56
57  (mysql): CREATE UNIQUE INDEX mdl_time_ins_uix ON mdl_timetable (instanceid)
58  Òxit
59
60  (mysql): CREATE TABLE mdl_timetable_group ( id BIGINT(10) unsigned NOT NULL
61  auto_increment, timetableid BIGINT(10) NOT NULL, studentsnumber BIGINT(10)
62  unsigned DEFAULT NULL, changeallowed TINYINT(1) DEFAULT NULL, commongroup
63  TINYINT(1) DEFAULT NULL, name VARCHAR(255) DEFAULT NULL, CONSTRAINT
64  PRIMARY KEY (id) )
65  Òxit
66
67  (mysql): ALTER TABLE mdl_timetable_group COMMENT='timetable_group table
68  retrofitted from MySQL'
69  Òxit
70
71  (mysql): CREATE INDEX mdl_timegrou_tim_ix ON mdl_timetable_group (timetableid)
72  Òxit
73
74  (mysql): CREATE TABLE mdl_timetable_users ( id BIGINT(10) unsigned NOT NULL
75  auto_increment, groupid BIGINT(10) NOT NULL, userid BIGINT(10) NOT NULL,
76  CONSTRAINT PRIMARY KEY (id) )
77  Òxit
78
79  (mysql): ALTER TABLE mdl_timetable_users COMMENT='timetable_users table
80  retrofitted from MySQL'
81  Òxit
82
83  (mysql): CREATE UNIQUE INDEX mdl_timehour_staweegroclae_uix ON
84  mdl_timetable_hourtable (starthour, weekday, groupid, classroomid, endhour)
85  Òxit
86
87  (mysql): CREATE INDEX mdl_timehour_staend_ix ON mdl_timetable_hourtable (
88  starthour, endhour)
89  Òxit
90
91  (mysql): CREATE INDEX mdl_timehour_cla_ix ON mdl_timetable_hourtable (
92  classroomid)
93  Òxit
94
95  (mysql): CREATE INDEX mdl_timehour_gro_ix ON mdl_timetable_hourtable (groupid)
96  Òxit
97
98  (mysql): CREATE TABLE mdl_timetable_classroomip ( id BIGINT(10) unsigned NOT
99  NULL auto_increment, classroomid BIGINT(10) unsigned NOT NULL, ip VARCHAR
100 (15) NOT NULL DEFAULT '', CONSTRAINT PRIMARY KEY (id) )
101 Òxit
102
103 (mysql): ALTER TABLE mdl_timetable_classroomip COMMENT='timetable_classroomip
104 ,
105 Òxit
106
107 (mysql): CREATE UNIQUE INDEX mdl_timeclas_claip_uix ON
108 mdl_timetable_classroomip (classroomid, ip)
109 Òxit
110
111 (mysql): CREATE INDEX mdl_timeclas_cla_ix ON mdl_timetable_classroomip (
112 classroomid)
113 Òxit
114
115 (mysql): CREATE TABLE mdl_timetable_hourtable_log ( id BIGINT(10) unsigned
116 NOT NULL auto_increment, starthour VARCHAR(5) NOT NULL DEFAULT '', weekday
117 VARCHAR(1) NOT NULL DEFAULT '', classroomid BIGINT(10) NOT NULL, endhour
118 VARCHAR(5) NOT NULL DEFAULT '', groupid BIGINT(10) NOT NULL, name VARCHAR
119 (255) DEFAULT NULL, data VARCHAR(10) NOT NULL DEFAULT '', CONSTRAINT
120 PRIMARY KEY (id) )
121 Òxit
122
123 (mysql): ALTER TABLE mdl_timetable_hourtable_log COMMENT='
124 timetable_hourtable_log
125 Òxit

```

```

79  (mysql): CREATE INDEX mdl_timeuser_gro_ix ON mdl_timetable_users (groupid)
80  Òxit
81
82  (mysql): CREATE INDEX mdl_timeuser_use_ix ON mdl_timetable_users (userid)
83  Òxit
84
85  (mysql): CREATE TABLE mdl_timetable_hourtable ( id BIGINT(10) unsigned NOT
86  NULL auto_increment, starthour VARCHAR(5) NOT NULL DEFAULT '', weekday
87  VARCHAR(1) NOT NULL DEFAULT '', classroomid BIGINT(10) NOT NULL, endhour
88  VARCHAR(5) NOT NULL DEFAULT '', groupid BIGINT(10) NOT NULL, name VARCHAR
89  (255) DEFAULT NULL, CONSTRAINT PRIMARY KEY (id) )
90  Òxit
91
92  (mysql): ALTER TABLE mdl_timetable_hourtable COMMENT='timetable_hourtable
93  table retrofitted from MySQL'
94  Òxit
95
96  (mysql): CREATE UNIQUE INDEX mdl_timehour_staweegroclae_uix ON
97  mdl_timetable_hourtable (starthour, weekday, groupid, classroomid, endhour)
98  Òxit
99
100 (mysql): CREATE INDEX mdl_timehour_staend_ix ON mdl_timetable_hourtable (
101 starthour, endhour)
102 Òxit
103
104 (mysql): CREATE INDEX mdl_timehour_cla_ix ON mdl_timetable_hourtable (
105 classroomid)
106 Òxit
107
108 (mysql): CREATE INDEX mdl_timehour_gro_ix ON mdl_timetable_hourtable (groupid)
109 Òxit
110
111 (mysql): CREATE TABLE mdl_timetable_classroomip ( id BIGINT(10) unsigned NOT
112 NULL auto_increment, classroomid BIGINT(10) unsigned NOT NULL, ip VARCHAR
113 (15) NOT NULL DEFAULT '', CONSTRAINT PRIMARY KEY (id) )
114 Òxit
115
116 (mysql): ALTER TABLE mdl_timetable_classroomip COMMENT='timetable_classroomip
117 ,
118 Òxit
119
120 (mysql): CREATE UNIQUE INDEX mdl_timeclas_claip_uix ON
121 mdl_timetable_classroomip (classroomid, ip)
122 Òxit
123
124 (mysql): CREATE INDEX mdl_timeclas_cla_ix ON mdl_timetable_classroomip (
125 classroomid)
126 Òxit
127
128 (mysql): CREATE TABLE mdl_timetable_hourtable_log ( id BIGINT(10) unsigned
129 NOT NULL auto_increment, starthour VARCHAR(5) NOT NULL DEFAULT '', weekday
130 VARCHAR(1) NOT NULL DEFAULT '', classroomid BIGINT(10) NOT NULL, endhour
131 VARCHAR(5) NOT NULL DEFAULT '', groupid BIGINT(10) NOT NULL, name VARCHAR
132 (255) DEFAULT NULL, data VARCHAR(10) NOT NULL DEFAULT '', CONSTRAINT
133 PRIMARY KEY (id) )
134 Òxit
135
136 (mysql): ALTER TABLE mdl_timetable_hourtable_log COMMENT='
137 timetable_hourtable_log
138 Òxit
139
140

```

```

121 (mysql): CREATE UNIQUE INDEX mdl_timehourlog_staweegroc_uix ON
      mdl_timetable_hourtable_log (starthour, weekday, groupid, classroomid,
      endhour, data)
122 Èxit
123
124 (mysql): CREATE TABLE mdl_timetable_classroomip_log ( id BIGINT(10) unsigned
      NOT NULL auto_increment, hourtable_log_id BIGINT(10) NOT NULL, ip VARCHAR
      (15) NOT NULL DEFAULT '', CONSTRAINT PRIMARY KEY (id) )
125 Èxit
126
127 (mysql): ALTER TABLE mdl_timetable_classroomip_log COMMENT='
      timetable_classroomip_log'
128 Èxit
129
130 (mysql): CREATE UNIQUE INDEX mdl_timeclaslog_houip_uix ON
      mdl_timetable_classroomip_log (hourtable_log_id, ip)
131 Èxit
132
133 S'han configurat correctament les taules Horari

```

Per últim si es vol accedir als fitxers d'ajuda, per qualsevol dels idiomes amb els quals s'han escrit aquests, i per simplificar aquest últim pas, s'haurà de copiar el contingut de la carpeta moodle/block-s/timetable/lang dins la carpeta moodle/lang sobreescrivint si s'escaigués els fitxers existents. També és necessari que el cron s'estigui executant, configurat tal com indica la documentació de Moodle <http://docs.moodle.org/es/Cron>. La configuració és l'habitual però s'haurà de tenir en compte que si es vol que es guardi l'històric de classes realitzades el cron s'haurà d'executar almenys un cop durant la classe.

La carga de este script no es muy alta, así que un intervalo de 5 minutos es razonable generalmente, pero si esto le preocupa, puede ampliar el periodo de tiempo a 15 minutos o incluso 30 minutos. Es mejor no establecer un intervalo de tiempo demasiado largo, ya que el retrasar el envío de mensajes de correo puede reducir la actividad del curso.

La documentació considera que 5 minuts es raonable, i es una bona configuració ja que es molt improbable que es programi una classe de duració igual o inferior a 5 minuts. Tot i això si fos el cas de que s'haguessin programat classes de duració tant curta, només caldria reduir el temps entre execucions del cron perquè s'executes almenys un cop durant la classe. Per saber quin és el funcionament del bloc consulteu el capítol de Manual d'usuari.

## Capítol 6

# Manual d'usuari

### 6.1 Introducció

A partir de la visualització inicial del bloc, la pantalla es divideix aproximadament com segueix. Cadascun dels usuaris, depenent del seu rol, tindrà accés a unes o altres opcions:

- Rol administrador (Figura 6.1 i Figura 6.2):

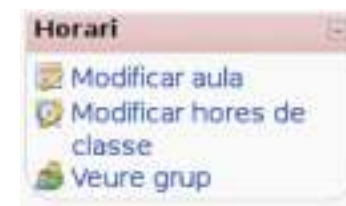


Figura 6.1: Aspecte del bloc per al rol administrador



Figura 6.2: Pestanyes vistes per al rol administrador

- Rol professor (Figura 6.3 i Figura 6.4):

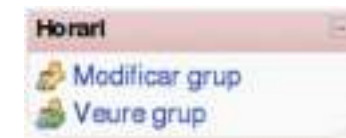


Figura 6.3: Aspecte del bloc per al rol professor



Figura 6.4: Pestanyes vistes per al rol professor

- Rol alumne (Figura 6.5 i Figura 6.6):

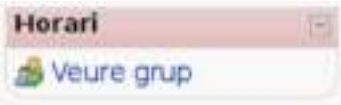


Figura 6.5: Aspecte del bloc per al rol estudiant

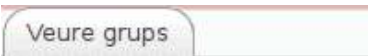


Figura 6.6: Pestanyes vistes per al rol estudiant

Desde qualsevol de les pestanyes, a la dreta d'aquestes, es pot accedir a l'ajuda (Figura 6.7), un resum sense el detall gràfic d'aquest document Manual de l'usuari.



Figura 6.7: Situació de l'accés a l'ajuda

Si l'usuari està apuntat a algun grup i aquest grup té classes programades es mostrarà la informació de la pròxima classe del usuari com indica la figura (Figura 6.8)



Figura 6.8: Si hi han classes programades per algun grup dels que està apuntat l'usuari es mostrarà la informació de la pròxima classe.

## 6.2 Editant les aules

Des d'aquesta pàgina (Figura 6.9) pot afegir, modificar i eliminar les aules d'aquesta institució.



Figura 6.9: Pantalla per editar aules

- **Afegir una aula** - des d'aquesta secció pot afegir una nova aula. Només són necessaris dos paràmetres per afegir una nova aula:
  - **nom** - el nom s'utilitza per identificar cadascuna de les aules.
  - **capacitat** - la màxima capacitat de les aules segons l'espai disponible.

Si s'afegeix una aula i no s'ha introduït cap valor per la seva capacitat (Figura 6.10),



Figura 6.10: Afegint una aula sense indicar-li la capacitat

l'aula s'acaba afegint a la llista d'aules amb el valor 0 com a capacitat (Figura 6.11).

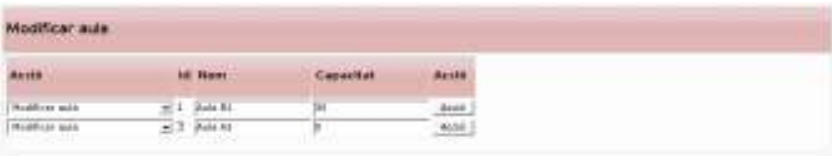


Figura 6.11: Aula creada amb capacitat 0

I si s'afegeix una aula sense cap valor per al nom (Figura 6.12),

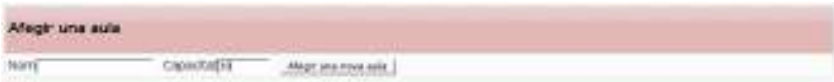


Figura 6.12: Afegir una aula sense indicar-hi cap nom

l'aula s'acaba afegint sense cap valor al seu nom ja que s'identifica amb un identificador únic per a cada aula afegida (Figura 6.13).

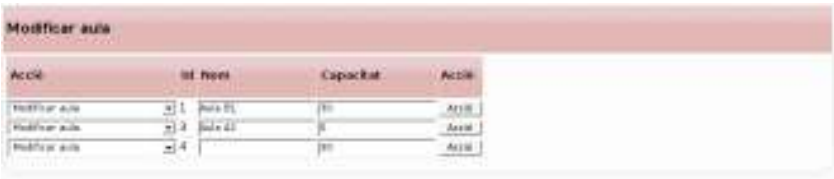


Figura 6.13: Aula creada sense nom

- **Editar aula** - des d'aquesta secció pot modificar o eliminar una aula que ja existeix. Aquesta és una llista de totes les aules disponibles. Si no s'han afegit aules apareixerà un missatge mostrant que no hi han aules disponibles. Es poden fer tres accions canviant el valor de la combo d'acció.
  - **Actualizar aula** - el nom i la capacitat poden ser modificats per cada fila de la llista però només pot ser actualitzat si aquesta acció és seleccionada a la combo d'acció i el botó d'acció és apretat.
  - **Eliminar aula** - l'aula pot ser eliminada si aquesta acció és seleccionada i el botó d'acció és apretat. Una pantalla com la següent (Figura 6.14) li demanarà que confirmi l'acció.



Figura 6.14: Confirmació per esborrar una aula.

- **Editar les adreces IP de l'aula** - amb aquesta opció es podran gestionar les adreces IP vàlides per cadascuna de les aules.
- **Editar les adreces IP de l'aula** - des d'aquesta secció se's capaç d'afegir un rang d'adreces IP utilitzades a l'aula (Figura 6.15).



Figura 6.15: Pantalla per gestionar adreces IP vàlides per una aula.

A la primera secció es poden prendre les mateixes accions que a la secció d'edició de aules però només per l'aula seleccionada. A la segona secció és mostrada una llista de les adreces IP afegides (Si no s'han afegit adreces IP es mostrarà un text indicant-ho, com es pot veure a la Figura 6.16) i serà capaç d'afegir una IP al rang o esborrar-la.

- **Afegir una IP** - una adreça IP pot ser afegida desde la primera secció d'aquesta segona secció, escrivint-la i prement el botó per Afegir IP. Si la IP no té un format correcte es mostrarà un missatge, tal i com mostra la següent imatge (Figura 6.16).



Figura 6.16: Missatge quan no s'han introduït adreces IP vàlides. El format per l'adreça IP ha de ser correcte.

- **Esborrar una IP** - a sota de la secció reservada per afegir una IP es mostrarà el rang d'adreces IP. Es pot esborrar cadascuna d'aquestes prement el botó per esborrar IP.

### 6.3 Editant hores de classe

Des d'aquesta pàgina (Figura 6.17) pot afegir i eliminar les hores de classe que aquesta institució està oferint.



Figura 6.17: Definició d'hores lectives.

- **Afegir una hora de classe** - des d'aquesta secció es pot afegir una nova hora de classe. Ha d'introduir hora d'inici i hora final i clicar el botó Afegir una nova hora de classe i aquesta hora s'afegirà a la llista de baix. Hora final ha de ser major que hora d'inici, sinó es mostrarà un missatge d'error a la part superior de les seccions tal i com mostra la següent imatge (Figura 6.18).



Figura 6.18: Missatge d'error en cas que les hores introduïdes no siguin correctes.

Per espais de temps de vint minuts entre hores es mostrarà una línia blanca.

- **Eliminar hora de classe** - A la part inferior de la secció per a afegir una hora de classe trobarà una llista ordenada de les hores de classe afegides, sinó es mostrarà un missatge que indicarà que no s'han afegit hores de classe. Per cada fila hi ha un botó d'esborrar. Se li demanarà que confirmi l'acció d'esborrar mitjançant la següent pantalla (Figura 6.19):



Figura 6.19: Petició de confirmació per esborrar una hora lectiva.

## 6.4 Editant Grups

Des d'aquesta pàgina (Figura 6.20) es pot afegir, modificar i eliminar els grups i els seus horaris per cada curs d'aquesta institució.

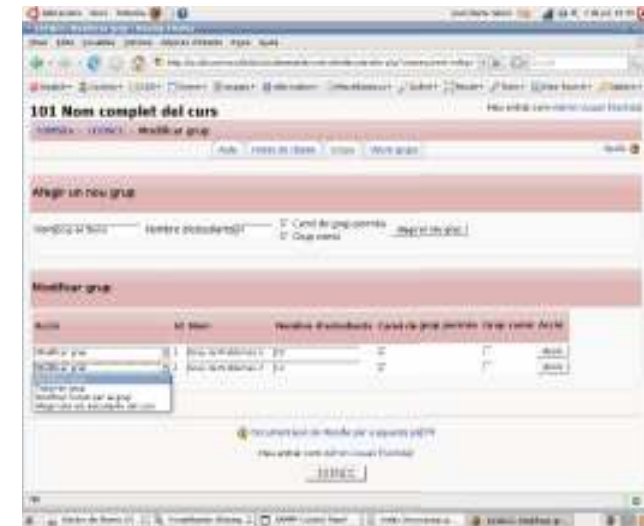


Figura 6.20: Pantalla per la gestió dels grups.

- **Afegir un grup** - des d'aquesta secció pot afegir un nou grup apretant el botó Afegir un nou grup . Els paràmetres necessaris per definir un grup són:
  - **nom** - el nom s'utilitza per identificar un grup d'entre els altres grups.
  - **número d'alumnes** - el número màxim d'alumnes permesos per apuntar-se al grup. Si s'omet aquest valor s'acaba creant el grup amb 0 places (Figura 6.21).



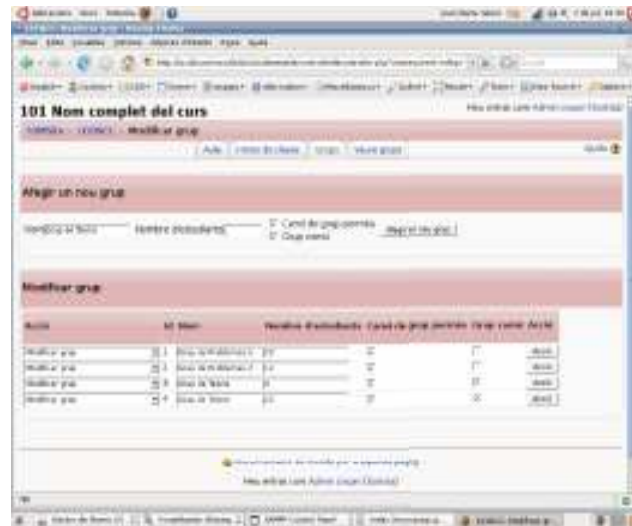


Figura 6.21: Ometent el paràmetre de número d'alumnes es crearà el grup amb 0 places.

- **canvi de grup permès** - el professor pot decidir si un alumne pot esborrar-se del grup ell mateix o ha de ser el professor qui l'esborri del grup.
- **grup comú** - si un grup és comú, això permet poder-se apuntar a aquest grup mentre l'alumne pot ésser apuntat en algun altre. Per un grup no comú un usuari només pot apuntar-se en un d'ells, i si l'usuari escull apuntar-se a un altre grup serà esborrat del primer grup.
- **Editar grup** - des d'aquesta secció pot modificar o eliminar un grup que ja existeix. Aquesta és una llista de tots els grups disponibles. Si no hi ha grups disponibles un missatge serà mostrat indicant-ho (Figura 6.22).



Figura 6.22: Pantalla per la gestió dels grups.

Es poden pendre quatre tipus d'accions canviant el valor de la combo d'acció:

- **Actualitzar grup** - Els paràmetres nom, el número d'alumnes, canvi de grup permès i grup comú es poden modificar per cada fila de la llista però només poden ser actualitzats si aquesta acció és seleccionada a la combo d'acció i el botó d'acció és apretat.
- **Eliminar grup** - el grup pot ser eliminat si aquesta acció és seleccionada i el botó d'acció apretat. Un missatge li demanarà que confirmi l'acció (Figura 6.23).



Figura 6.23: Confirmació per esborrar un grup.

- **Afegir tots els estudiants del curs** - tots els alumnes del curs seran introduïts a la llista un cop el botó d'acció sigui pressionat. Un missatge li demanarà que confirmi l'acció (Figura 6.24).



Figura 6.24: Confirmació per la incorporació de tots els alumnes del curs en aquest grup.

- **Editar una taula d'hores del grup** - aquesta acció et porta a la pantalla d'edició de l'horari del grup.
- **Editar una taula d'hores del grup** - des d'aquesta secció (Figura 6.25) pot modificar l'horari del grup seleccionat.

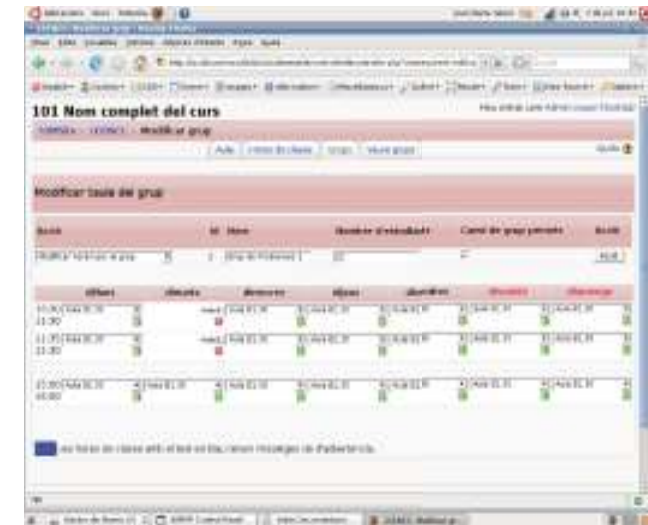


Figura 6.25: Pantalla per a crear l'horari per cadascun dels grups.

A la primera secció pot fer les mateixes accions que a la secció d'Edició de grup però només pel grup seleccionat. A la segona secció hi ha una representació de la setmana ( dies de cap de setmana < en text vermell) és mostrat amb totes les hores que la institució està donant classes. Per cadascuna d'aquestes caselles es poden fer una d'aquestes dues accions:

- **Afegir** - Si aquesta acció és possible es mostra una combo amb les aules disponibles a la institució per aquesta hora. Una classe pot ser afegida seleccionant l'aula apropiada i clicant l'icona afegir. Si la capacitat de l'aula és menor que el número d'alumnes que tenim previst un missatge d'alerta en color blau apareixerà a la secció superior i la classe serà afegida en text blau (Figura 6.26).

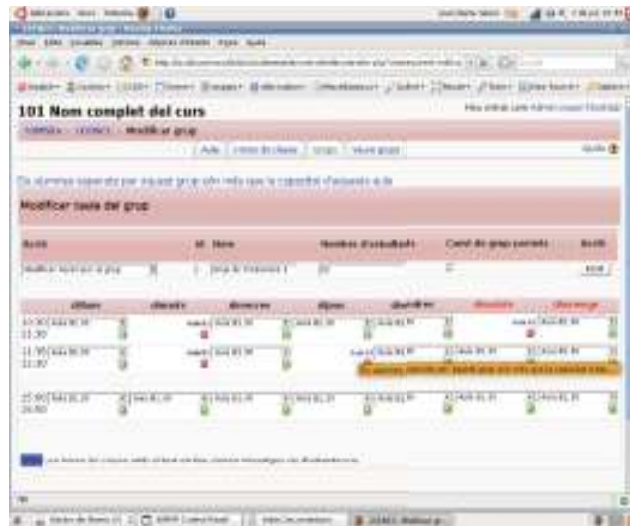


Figura 6.26: Missatge si l'aula té una capacitat menor que el nombre d'alumnes del grup assignat.

Si la classe és afegida en un dia considerat cap de setmana un missatge d'alerta en color blau apareixerà a la secció superior i la classe serà afegida en text blau (Figura 6.27).

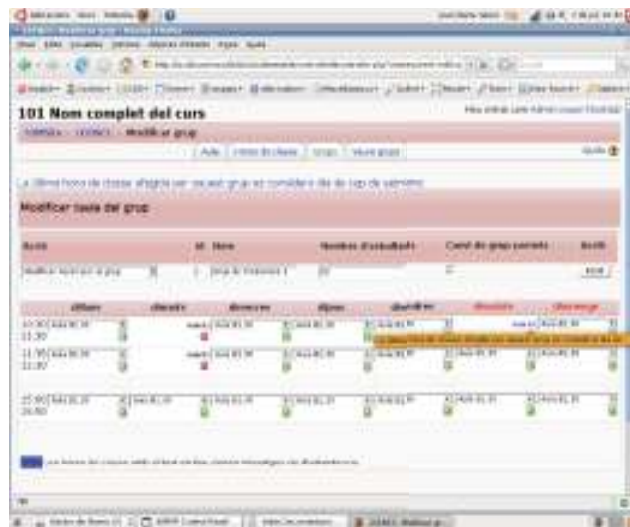


Figura 6.27: Missatge si la classe és assignada en un dia considerat no lectiu pel centre.

Si passen tots dos casos allora es mostren tots dos missatges allora (Figura 6.28).

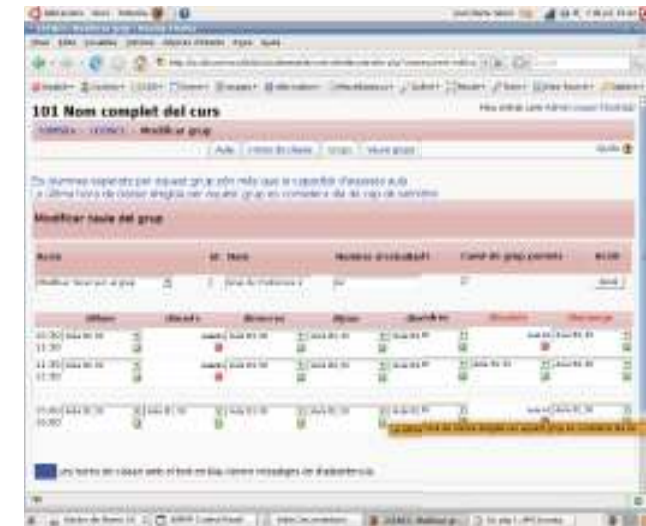


Figura 6.28: Missatge amb les dues advertències allora.

– **Eliminar** - Si una classe va ser afegida la pot eliminar clicant la icona Eliminar.

## 6.5 Veient els grups

Des d'aquesta pàgina (Figura 6.29) pot veure els alumnes apuntats al grup i l'horari del grup seleccionat.

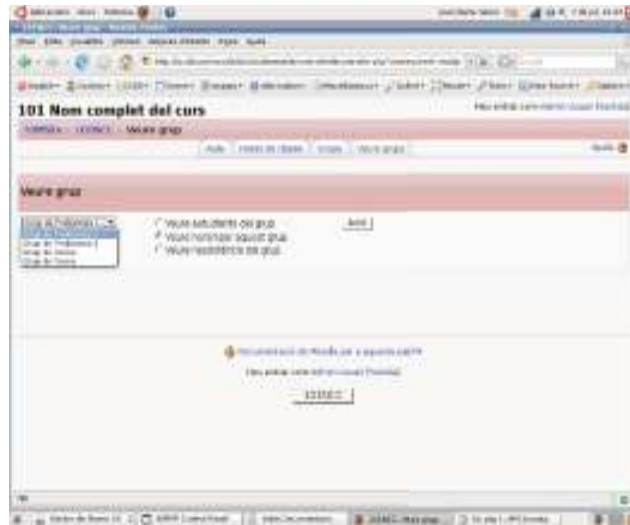


Figura 6.29: Pantalla per a la visualització d'informació dels grups vista amb el rol d'administrador.

Depenent del rol de l'usuari es veuen unes o altres accions a realitzar (Figura 6.30):



Figura 6.30: Pantalla per a la visualització d'informació dels grups vista amb el rol d'estudiant.

- **Veure el grup d'alumnes** - seleccionant un grup al combo i l'acció veure alumnes del grup

veurà una llista de tots els alumnes apuntats (Figura 6.31) sinó un missatge indicarà que no s'hi han apuntat alumnes (Figura 6.32).



Figura 6.31: Llistat dels alumnes apuntats al grup vist amb el rol d'estudiant.



Figura 6.32: Missatge si no hi han alumnes apuntats al grup.

Si té permisos pot excloure l'alumne del grup (Figura 6.33).



Figura 6.33: Llistat dels alumnes apuntats al grup vist amb el rol d'administrador.



Figura 6.34: Confirmació a l'hora de treure un alumne d'un grup.

- **Veure la taula d'hores del grup** - seleccionant un grup al combo i l'acció veure la taula d'hores del grup veurà un horari pel grup seleccionat (Figura 6.35).



Figura 6.35: Visualització del horari pel grup amb l'opció d'apuntar-shi (Botó a la part superior dreta).

Vostè es pot apuntar al grup seleccionat pulsant el botó a la part superior dreta del horari.







Figura 6.39: Missatge d'error si el rang és invàlid (en vermell a la part superior) i missatge si entre les dates escollides no hi han hagut registres.

## Capítol 7

# Avaluacions

Durant el període de desenvolupament s'ha pogut comprovar que el bloc té un funcionament correcte sobre moltes plataformes i navegadors web. A continuació és detalla quins han estat aquests escenaris de proves, tant la part on ha estat instal·lat el servidor Moodle com en quins navegadors s'ha provat.

### 7.1 Servidor

Aquest bloc de Moodle es va començar a desenvolupar en un servidor local XAMPP sobre Windows XP.

XAMPP és un paquet de programari lliure que conté el servidor HTTP Apache, base de dades de MySQL i eines necessàries per utilitzar el PHP i el llenguatge de programació Perl. El programa es llança sota el GNU de Llicència Pública General i d'un servidor web, d'ús fàcil, capaç de servir pàgines dinàmiques. Actualment, XAMPP està disponible per a Windows, Linux, Solaris i Mac OS X (el X en el seu nom podria estar dret per a un qualsevol d'aquests sistemes operatius).

XAMPP és pot obtenir d'aquesta url <http://www.apachefriends.org/en/xampp.html>.

Es van començar les proves i el desenvolupament amb la versió 1.7 de Moodle, i més tard tal com indica el capítol sobre la versió de Moodle necessària (Secció 4.2.1.8, pàgina 42) amb la versió 1.8, concretament amb la versió 1.8.7.

Localment s'ha provat, posteriorment, aquest bloc amb una versió de XAMPP corrent sobre Linux Ubuntu (Primer amb la versió 7.10 i després sobre la versió 8.4). Aquest servidor XAMPP és compost per:

- Servidor Web Apache 2.2.8
- Servidor de Base de dades MySQL 5.0.51a
- PHP Versió 5.2.5

Per continuar amb les proves i el desenvolupament es va passar a instal·lar, també primer la versió de Moodle 1.7 i posteriorment la 1.8 (Moodle 1.8.3 primer i Moodle 1.8.7 després) en un servidor de hosting gratuït amb les següents especificacions.

- Sistema Operatiu Linux
- Servidor Web Apache Versió 2.2.10 (Unix)
- PHP Versió 5.2.6
- Servidor de Base de dades MySQL Version 5.0.51

Va ser en aquest servidor de hosting gratuït, situat a Nova York on es va descobrir el problema de la versió mínima (Secció 4.2.3.2, pàgina 44). Aquest servidor es pot trobar a <http://jmtalarn.byethost10.com/moodle>.

## 7.2 Client

Per confirmar el correcte funcionament en el màxim nombre de clients, s'ha comprovat un funcionament correcte amb el següent llistat de navegadors web:

- Internet Explorer 6 amb Windows XP
- Internet Explorer 7 amb Windows XP
- Firefox 2 amb Windows XP
- Firefox 3 amb Windows XP
- Safari amb Windows XP
- Firefox 2 amb Ubuntu 7.10
- Firefox 2 amb Ubuntu 8.4
- Firefox 3 amb Ubuntu 8.4
- Konqueror amb Ubuntu 8.4

## Capítol 8

## Conclusions

Moodle és una servidor de continguts per a centres educatius, on, vist la experiència i sobre molts comentaris que es poden trobar als forums de Moodle, el seu èxit es deu principalment sobre la possibilitat d'instalar-ho en múltiples plataformes sense masses complicacions.

Tanmateix, aquest projecte ha servit per comprovar que la principal avantatge del software lliure no recau en la seva gratuïtat, tot i que sí que es tracta d'una gran avantatge, sinó en la comunitat que l'envolta.

La comunitat Moodle, com moltes altres comunitats de software lliure, és una comunitat amb moltíssima activitat i una gran quantitat de recursos disponibles per a consultar sobre qualsevol dels àmbits (instal·lació, desenvolupament, ús, etc.) que pot implicar l'implementació d'una sol·lució Moodle mitjançant els fòrums, documents i seguiment d'incidències.

Tota aquesta informació està total o parcialment traduïda a moltíssimes llengües, fet que també repercuteix en la seva ampla difusió ja que el fa accessible a molta gent.

Es en aquesta comunitat d'on s'ha extret gairebé tot el suport per al desenvolupament d'aquest projecte.

Aquest desenvolupament ha servit principalment en veure quina pot ser la complexitat en el desenvolupament de les unitats de software. I com tota planificació de qualsevol projecte, encara que sigui molt importat i d'obligatorietat, pot ser afectada i modificada per múltiples factors i acabant desviant la previsió inicial; recursivament, una bona previsió serveix per preveure aquestes situacions o si més no, de quina manera poden ser corregides.



## Capítol 9

# Annex. Extractes de la documentació utilitzada.

En aquest capítol s'inclouen extractes de la documentació utilitzada per l'elaboració d'aquest bloc de Moodle. En aquest document es troben referències a aquest capítol per clarificar aquells punts on podria ser necessària una lectura del material utilitzat per la presa de decisions o resolució de problemes.

### 9.1 Definint el fitxer xmldb

#### Development:XMLDB defining an XML structure



##### Justification

Before Moodle 1.7, all the DB install and upgrade was developed twice (once to handle MySQL installations and another to handle PostgreSQL installations). This approach, although working, has caused some headaches in the past, mainly because it was really difficult to keep both lines of development 100% on sync.

Some developers do they work against one RDBMS and it was complex to develop to the other one (two test environments, skills on both databases, slower development cycle...). And all this was happening with only two supported RDBMS!

One of the main objectives of Moodle 1.7 is to extend the the number of supported RDBMS to other flavours (more exactly, to Oracle and MSSQL). And the old approach (one line of development for each DB) could become an absolute nightmare.

Because of this we have planned to build one structure to define all the DB objects used by Moodle. This structure will provide the necessary level of abstraction to be shared by all the RDBMS systems, so the "multiple lines of development" explained in the previous paragraph will be out forever, giving us one robust and well defined way to handle DB objects independently of the exact RDBMS being used.

##### Implementation

Initially all our best wishes were to use the AdoDB XML Schema (<http://phplens.com/lens/adodb/docsdadict.htm#xmlschema>).

As Moodle is using ADOdb libraries to communicate with databases it sounded like the natural approach to solve the problem. But, finally, two reasons prevented us to use it:

1. Although working, it seems to be one feature in progress, with important changes/evolutions arriving at the time of write this document.
2. Its lack of support for "prefixes" (one Moodle key feature, to allow multiple instances to run in the same server), would force us to create some awful tricks to generate the objects.

So, finally, we decided to build our own XML files, with everything we need to define every object present in the DB.

#### The XMLDB editor

Although the XML is pretty simple to read (and to write), one of the major drawbacks was its easy and errorprone adoption by the developers. Also some problems with versioning systems getting crazy with XML files (thanks ML!) pointed us to the requirement to use one high-density format (it means, physically long lines) in our XML files.

After some intense thoughts we decided to build one specialised editor for our XML format. This editor should be easy to use and provide support for all the objects present one Moodle DB. And it's done (and will support future enhancements easily, we hope).

The XMLDB Editor makes the edition of tables/fields/keys/indexes practically a trivial task, allowing the developer to spend the time coding and improving things instead of fighting against XML files and the errors caused by manual editing (of course, the developer is free to use such extra-time as desired, beers, dance,books, music...) :-)

All the new install.xml files, present under each db directory in Moodle can be edited (and we recommend it) with just some clicks and keystrokes. Those install.xml will contain all the info needed to generate the specific objects needed for each RDBMS supported. Obviously, such files, are the neutral replacement for all the \*.sql files used until now.

#### Launching

Just login to your server as an administrator and, under the Miscellaneous tab of the Administration Block, you'll see a new link pointing to the "XMLDB Editor".

One important note is that, to be able to handle files properly, the web server needs write access to all those "db" directories where the "install.xml" files reside (and to the files themselves, of course). :-)

That's all!

#### Use

We really think the XMLDB Editor is pretty easy to use, so here you won't see a complete guide to use it. We highly recommend you to play with it for a while, viewing how it works and how it modifies the install.xml files.

It's organised in a top-bottom structure, where you start loading (or creating) a new XMLDB file. Then, you can edit such file and its general structure will be showed. This structure have two type of elements, tables and statements and the XMLDB Editor allows you to add, edit, delete, and move them easily. Also, for initial creation of tables, one small but effective reverse-enginery tool has been developed (only under MySQL) allowing you to retrofit any table from the DB to the XMLDB Editor.

While editing tables you will see their fields, keys and indexes and you'll be able to handle all them easily.

Note that some fields can be no-editable. It uses to be because they are being used in some way (part of one key or index) and the idea is to warn you about that.

Fields can be edited and you can specify their name, type, length, decimals, null-ability, defaults and so one. Exactly the same for both keys and indexes.

While editing statements, you must think about them like "collections of sentences". Once you select the type (only inserts are allowed for now) and table you are interested you'll be able to introduce the exact values easily, being able to duplicate them easily to gain some speed if you have a lot of sentences in your development. Sentences can be edited and deleted easily too.

One interesting feature is that all the XMLDB Editor pages allow you to enter one comment about the item being modified (table, index, key, field, statement...). Use it at your entire needs, sure it helps other developers to understand a bit more the DB model.

Please, don't forget to read and understand the next section, where we talk about some important guidelines to create and handle XMLDB files.

### 9.2 Variables globales

Moodle maneja la información más recurrida mediante variables globales a las cuales tiene acceso desde todos los ficheros que desee (si no conoces mucho sobre PHP, te sugiero que busques información sobre el tema en webs como [www.lawebdelprogramador.com](http://www.lawebdelprogramador.com) o [www.webalia.com](http://www.webalia.com)). Estas variables globales son las siguientes:

**\$CFG:** Es relativo a la configuración del curso, guarda los valores sobre la dirección raíz del sistema (wwwroot) o el prefijo de las tablas de la BDD (prefix) entre otras cosas.

**\$SESSION:** Guarda los datos sobre cada sesión abierta (creo que se refiere a las cookies que crea al iniciar cada una de las sesiones, pero esto deberías confirmarlo).

**\$THEME:** Esta variable se refiere al modelo de interfaz que estas usando, los que se encuentran en el directorio /moodle/theme

**\$SITE:** Sinceramente no recuerdo para que vale.

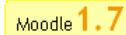
**\$COURSE:** Esta es muy importante ya que sirve para conocer toda la información sobre los cursos sin necesidad de buscar en cada momento la información del curso actual.

**\$USER:** Igualmente, esta variable contiene toda la información útil sobre cada usuario. Por ejemplo para acceder a su identificador se puede acceder como `$USER->id`

Con estas variables tendrás casi toda la información que puedas necesitar en un momento dado, pero de todas formas puedes hacer consultas y demás usando las funciones que ya ha implementado Moodle o tus propias consultas. Moodle además guarda un log de todo lo que ocurre en el sitio mediante una tabla de la base de datos (tabla log como te comenté). Los resultados se muestran por ejemplo cuando se accede a la consulta de registros como administrador. Si cuando implementes tus módulos, bloques, o lo que quieras hacer, quieres que se guarde un log, es suficiente con que uses la función que te comentaba en el post anterior (`add_to_log`). No recuerdo muy bien la sintaxis que tiene, pero te sugiero que busques en cualquier archivo de moodle que guarde información en el registro (sobre cursos, accesos al sitio o cualquier otra cosa) y observes su comportamiento, porque es realmente sencillo.

### 9.3 Definició de rols. Fitxer access.db.

#### Programming Interface



Although the Roles system may look complicated at first glance, implementing it in Moodle code is fairly simple. You need to define each capability once, so that Moodle can upgrade existing roles to take advantage of it. You do this in an `access.php` inside the `db` folder of any module (eg see `mod/forum/db/access.php`). The array contains entries like this (note the descriptions for the legacy roles which provides forward compatibility):

```
1  'mod/forum:viewforum' => array(
2      'captype' => 'read',
3      'contextlevel' => CONTEXT_MODULE,
4      'legacy' => array(
5          'guest' => CAP_PREVENT,
6          'student' => CAP_ALLOW,
7          'teacher' => CAP_ALLOW,
8          'editingteacher' => CAP_ALLOW,
9          'coursecreator' => CAP_ALLOW,
10         'admin' => CAP_ALLOW
11     )
12 ),
```

To load/change these capabilities you need to bump the module version. There's no need to provide changes or differences as Moodle will scan the whole array and sort it out. On each page you need to find the context the user is working in, using the `get_context_instance()` function. For example, in the forum module:

```
1  $context = get_context_instance(CONTEXT_MODULE, $cm->id);
```

Then, whenever you want to check that the current user has rights to do something, call `has_capability()` like this:

```
1  if (!has_capability('mod/forum:viewforum', $context)) {
2      print_error('nopermissiontoviewforum');
3  }
```

If you just want to assert a capability and then finish with an error message if it's not met (as we did above), then a shorter way it to use `require_capability()` like this:

```
1  require_capability('mod/forum:viewforum', $context);
```

Note that there are extra parameters you can specify to get a custom error message, otherwise users get an automated "No permissions" message that lists the permission they were missing.

As a result of the new Roles System, all calls to `isadmin()`, `iscoursecreator()`, `isteacheredit()`, `isteacher()`, `istudent()`, and `isguest()` will have to be replaced with calls to `has_capability()` or `require_capability()`. However, these functions will be retained for some backward compatibility with old code, using the legacy capabilities to try and work out what to do.

### 9.4 Funcions d'accès a dades.

```
get_record($table, $field1, $value1, $field2="", $value2="",
           $field3="", $value3="", $fields='') X-Ref
Get a single record as an object
param: string $table The table to select from.
param: string $field1 the first field to check (optional).
param: string $value1 the value field1 must have (required if
field1 is given, else optional).
param: string $field2 the second field to check (optional).
param: string $value2 the value field2 must have (required if
field2 is given, else optional).
param: string $field3 the third field to check (optional).
param: string $value3 the value field3 must have (required if
field3 is given, else optional).
return: mixed a fieldset object containing the first matching
record, or false if none found.
```

```
get_record_sql($sql, $expectmultiple=false, $nolimit=false) X-Ref
Get a single record as an object using an SQL statement
The SQL statement should normally only return one record. In
debug mode
you will get a warning if more record is returned (unless you
set $expectmultiple to true). In non-debug mode, it just returns
the first record.
param: string $sql The SQL string you wish to be executed,
should normally only return one record.
param: bool $expectmultiple If the SQL cannot be written to
conveniently return just one record,
param: bool $nolimit sometimes appending ' LIMIT 1' to the SQL
causes an error. Set this to true
return: Found record as object. False if not found or error
```

<b>get_record_select(\$table, \$select="", \$fields='*') X-Ref</b>
Gets one record from a table, as an object
param: string \$table The database table to be checked against. param: string \$select A fragment of SQL to be used in a where clause in the SQL call. param: string \$fields A comma separated list of fields to be returned from the chosen table. return: object false Returns an array of found records (as objects) or false if no records or error occurred.
<b>get_records(\$table, \$field="", \$value="", \$sort="", \$fields='*', \$limitfrom="", \$limitnum="") X-Ref</b>
Get a number of records as an array of objects.
If the query succeeds and returns at least one record, the return value is an array of objects, one object for each record found. The array key is the value from the first column of the result set. The object associated with that key has a member variable for each column of the results.
param: string \$table the table to query. param: string \$field a field to check (optional). param: string \$value the value the field must have (required if field1 is given, else optional). param: string \$sort an order to sort the results in (optional, a valid SQL ORDER BY parameter). param: string \$fields a comma separated list of fields to return (optional, by default param: int \$limitfrom return a subset of records, starting at this point (optional, required if \$limitnum is set). param: int \$limitnum return a subset comprising this many records (optional, required if \$limitfrom is set). return: mixed an array of objects, or false if no records were found or an error occurred.
<b>get_records_select(\$table, \$select="", \$sort="", \$fields='*', \$limitfrom="", \$limitnum="") X-Ref</b>
Get a number of records as an array of objects.
Return value as for @see function get_records.
param: string \$table the table to query. param: string \$select A fragment of SQL to be used in a where clause in the SQL call. param: string \$sort an order to sort the results in (optional, a valid SQL ORDER BY parameter). param: string \$fields a comma separated list of fields to return param: int \$limitfrom return a subset of records, starting at this point (optional, required if \$limitnum is set). param: int \$limitnum return a subset comprising this many records (optional, required if \$limitfrom is set). return: mixed an array of objects, or false if no records were found or an error occurred.

<b>get_records_list(\$table, \$field="", \$values="", \$sort="", \$fields='*', \$limitfrom="", \$limitnum="") X-Ref</b>
Get a number of records as an array of objects.
Return value as for @see function get_records.
param: string \$table The database table to be checked against. param: string \$field The field to search param: string \$values Comma separated list of possible value param: string \$sort Sort order (as valid SQL sort parameter) param: string \$fields A comma separated list of fields to be returned from the chosen table. If specified, return: mixed an array of objects, or false if no records were found or an error occurred.
<b>get_records_sql(\$sql, \$limitfrom="", \$limitnum="") X-Ref</b>
Get a number of records as an array of objects.
Return value as for @see function get_records.
param: string \$sql the SQL select query to execute. The first column of this SELECT statement param: int \$limitfrom return a subset of records, starting at this point (optional, required if \$limitnum is set). param: int \$limitnum return a subset comprising this many records (optional, required if \$limitfrom is set). return: mixed an array of objects, or false if no records were found or an error occurred.
<b>insert_record(\$table, \$dataobject, \$returnid=true, \$primarykey='id') X-Ref</b>
Insert a record into a table and return the "id" field if required
If the return ID isn't required, then this just reports success as true/false. \$dataobject is an object containing needed data
param: string \$table The database table to be checked against. param: object \$dataobject A data object with values for one or more fields in the record param: bool \$returnid Should the id of the newly created record entry be returned? If this option is not requested then true/false is returned. param: string \$primarykey (obsolete) This is now forced to be 'id'.

## 9.5 Debat sobre l'obligatorietat del camp ID.

Hmm, the consequences of using non-Moodle standard design may be decreased ability for others to test/help with your code and integrate it into the standard system and increased chance of forking.

And in any case, id-less are sooooo last century, that we don't care about them any more. So the 4th parameter is now deprecated.

Did this ever happen? I am just looking at insert\_record from HEAD, and the 4th parameter is still sitting there with no indication that it is deprecated.

I think we should edit the function comment ASAP to mark this parameter deprecated, possibly with a link to this thread as explanation.

Then, shortly after 1.6 is released, we add code so that if you are in debug mode, and this function is called with a 4th parameter, you get a warning (probably generated with trig-

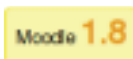
```
ger_error('Call to insert_record() with a 4th parameter. This is strongly deprecated.');
```

I only can say that, from a relational-DB perspective, I agree 100% with your posts. I remember, some time ago, when I arrived to Moodle, that I was really surprised about all those "id" artificial PKs (not necessary nor used at all). More yet, my first DB contributions (the backup tables) didn't contain such artificial "id" at all and "true" PKs were used instead.

But problems arrive when any of the get\_record\_XXX() functions are used, because all them returns associative arrays (GetAssoc()) where the first field in the returned recordset becomes the key of the array. So having those artificial "id" fields for each table is the only mechanism to fetch all the records from any table without losing some of them by the "key overwritten" effect. Following your example, something so simple like: get\_records('user\_teachers') won't work if the id field is missing. Only one occurrence of every teacher (the first field in the select, userid) will be returned, missing a lot of records. I'm pretty sure that we could bypass this limit in the long term (current recordsets implementation is a good start) but it would force us to review all the get\_records\_XXX() calls before, because a lot of them, simply will break and others could be using the GetAssoc() key feature in its own benefit so, modifying the type of arrays returned by the functions isn't an immediate alternative.

I really think that it's the reason to have that mandatory id column inside each table. And the proposed change about killing the 4th parameter in the insert\_record() function was only to enforce it more...

## 9.6 Correcció de problemes amb la funció usergetdate().



Also you need to update your Moodle to the latest 1.8.5+ or 1.9+ release to get the usergetdate() (and other) functions working properly.

You need to use Moodle functions to guarantee that everything is ok, without mixing php functions (mktime, date...) with moodle functions (usergetdate).

So, to get the GMT timestamp of one date... you need to execute:

```
1 $timestamp = make_timestamp(2008, 4, 9, 14, 0, 0, 'America/New_York');
2
```

And then you can handle that GMT timestamp as you prefer, for example:

- To print the date of that GMT timestamp in different cities (timezones):

```
1 echo userdate($timestamp, '', 'America/New_York');
2 echo userdate($timestamp, '', 'Spain/Madrid');
3
4 - To get the date components of that GMT timestamp in different
5 cities (timezones):\
6 \begin{lstlisting}[style=PHP]
   print_object( usergetdate($timestamp, 'America/New_York'));
```

```
7 | print_object( usergetdate($timestamp, 'Europe/Madrid'));
```

# Bibliografia

- [1] Manual de PHP <http://www.php.net/docs.php>
- [2] Moodle <http://moodle.org/>
- [3] Moodle Docs <http://docs.moodle.org/overview/>
- [4] Using Moodle <http://moodle.org/course/view.php?id=5>
- [5] Using Moodle - Forums <http://moodle.org/mod/forum/index.php?id=5>
- [6] PHP Cross Reference of Moodle 1.9 <http://xref.moodle.org/nav.html?index.html>
- [7] Moodle Technical Documentation (phpdocumentor) <http://phpdocs.moodle.org/>
- [8] Moodle Tracker <http://tracker.moodle.org/secure/Dashboard.jspa>
- [9] XAMPP <http://www.apachefriends.org/en/xampp.html>
- [10] Article de XAMPP a la Viquipèdia <http://ca.wikipedia.org/wiki/XAMPP>
- [11] Manual de Latex [http://es.wikibooks.org/wiki/Manual\\_de\\_LaTeX](http://es.wikibooks.org/wiki/Manual_de_LaTeX)