Universidad de Valladolid

E. T. S. de Ingeniería Informática

TRABAJO FIN DE GRADO

Grado en Ingeniería Informática

“Aportando Seguridad en las Calificaciones de Moodle”

Autor:

Dª. Sofía Oraá Pérez

Tutores:

Dª. Carmen Hernández Díez

D. Óscar Delgado Mohatar

Quiero expresar mi más sincero agradecimiento a todas las personas involucradas directa o indirectamente en la realización de este proyecto, tanto por su esfuerzo personal como por sus ánimos en este periodo. En particular a Dª. Carmen Hernández Diez y al Dr. Óscar Delgado Mohatar, por brindarme la oportunidad de poder desarrollar este trabajo y a todos los profesores que nos han ayudado a lo largo de todo este recorrido. Por supuesto, también quiero agradecer a mi familia y amigos, muchas gracias por vuestro tiempo y paciencia.

TABLA DE CONTENIDOS:

1 INTRODUCCIÓN

1.1 Objetivos del proyecto

El objetivo principal del proyecto es el diseño e implementación de una aplicación capaz de controlar las notas almacenadas en Moodle para cualquier tipo de titulación y asignatura, tratando de evitar que pase desapercibido para la persona responsable del curso, una modificación ilícita de las calificaciones, ya sea a través de la interfaz gráfica de la herramienta o directamente en el servidor que contiene la base de datos.

El segundo objetivo será la implementación de medidas de seguridad extra en las calificaciones almacenadas utilizando cifrado.

Como objetivo complementario tenemos la creación de un manual de usuario de requisitos previos en Moodle, por ejemplo, la creación de usuarios (globales al sistema, profesores y alumnos asignados a cursos concretos), cursos y notas. Asimismo, se explicará detalladamente las tablas utilizadas en el diseño de los algoritmos de acceso a la plataforma para obtener la información necesaria.

Nos interesa que la aplicación sea fundamentalmente útil, que ayude a los responsables de los cursos a mantener la seguridad en sus calificaciones. Por ello, consideramos que la utilización de una aplicación externa al sistema para contrastar las modificaciones en las notas de los alumnos permitirá:

- Almacenar las notas finales de actividades y cursos de manera segura sin que el profesor tenga que comprobar por sí mismo que no ha habido una modificación ilegal de las notas almacenadas.
- Avisar al profesor, gestor o personal encargado, en caso de que se produzca un cambio en las notas, ya sea autorizado o no, permitiendo revelar los cambios no deseados, además de mantener un log de las incidencias detectadas en el sistema.
- El almacenamiento de las notas cifradas durante varios años para futuras reclamaciones tal y como estipula la ley.

PHP es el lenguaje utilizado en el desarrollo de Moodle. Si la aplicación fuera lo suficientemente útil, la parte clave de la documentación sería traducida al inglés para poder compartirla con otros usuarios que utilicen la plataforma Moodle.

1.2 Cronograma

1.2.1 Introducción

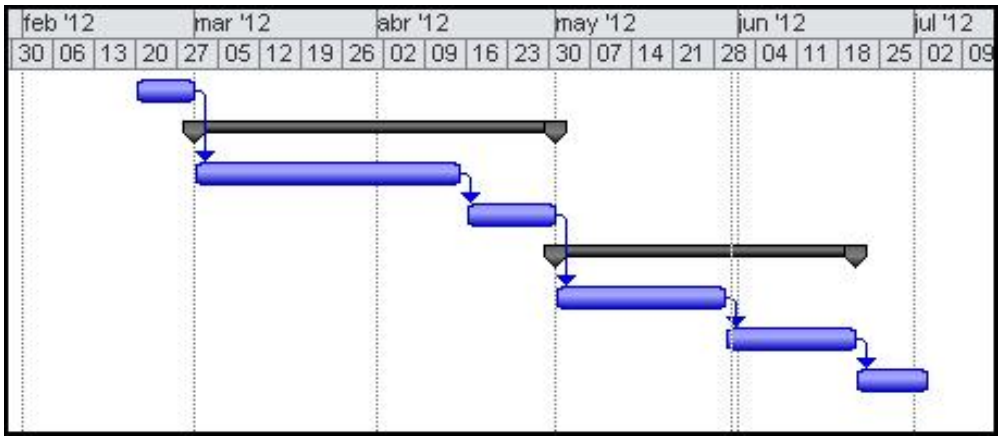
Se debe de tener en cuenta que al no tener demasiada experiencia en la planificación es muy probable que las estimaciones no sean las adecuadas. Basándonos, por tanto, en la poca experiencia previa en proyectos largos trataremos de estimar un tiempo adecuado.

1.2.2 Plan de Fases

El desarrollo del proyecto se dividirá en cuatro fases, cada una dividida en iteraciones dependiendo de la complejidad de la misma.

Fase	Inicio	Elaboración	Construcción	Transición
Número de Iteraciones	1	2	2	1

	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras
1	Inicio	8 días	lun 20/02/12	mié 29/02/12	
2	**Elaboración**	**43 días**	**jue 01/03/12**	**lun 30/04/12**	
3	Iteración 1	32 días	jue 01/03/12	sáb 14/04/12	1
4	Iteración 2	11 días	lun 16/04/12	lun 30/04/12	3
5	**Construcción**	**37 días**	**mar 01/05/12**	**mié 20/06/12**	
6	Iteración 1	21 días	mar 01/05/12	mar 29/05/12	4
7	Iteración 2	16 días	mié 30/05/12	mié 20/06/12	6
8	Transición	8 días	jue 21/06/12	lun 02/07/12	7



1.2.3 Objetivos e Hitos

Fase	Descripción	Hito
Inicio	Objetivos de la aplicación. Calendarización.	Documento de plan de proyecto.
Elaboración	Identificación de requisitos. Estudio de la Bases de Datos de Moodle. Realización de casos de uso de análisis. Arquitectura de sistema. Diseño de algoritmos. Realización de casos de uso de diseño.	Documento de Base de Datos de Moodle. Documento de análisis. Documento de diseño.
Construcción	Implementación del sistema. Pruebas. Pruebas en entorno Real.	Versión final de sistema. Documento de Pruebas. Manual de Instalación.
Transición	Redacción Anexos. Redacción Final del Documento TFG. Realización de la Presentación.	Documento final del TFG. Presentación.

1.3 Estructura del documento

La información de esta documentación se estructura como sigue:

- En el Capítulo 1 encontramos los objetivos principales de nuestro sistema y el cronograma de su construcción.
- En el Capítulo 2 mostramos información relevante sobre la estructura de la Base de Datos de Moodle. Esta documentación ha sido extraída de la bibliografía técnica consultada y de la página oficial [docs.moodle.org], así como de las numerosas pruebas del sistema en una máquina local.
- En el Capítulo 3 se detallan brevemente nociones básicas sobre seguridad de la información.
- Los Capítulos 4 y 5 recogen el Análisis y el Diseño del sistema respectivamente según la estructura que propone UPEDU. En el Capítulo 5 se añadirá un apartado especial con la explicación del diseño de los algoritmos empleados para acceder a la plataforma Moodle.
- Los Capítulos 6, 7, 8 y 9 contienen el documento de implementación, las pruebas, las conclusiones y las líneas futuras de mejora del trabajo de fin de grado respectivamente.
- Por último encontramos el seguimiento del proyecto y la bibliografía.
- Anexos: Contenido del CD adjunto; Manual Básico de Moodle; Ventajas de la Plataforma Moodle.

2 MOODLE

2.1 Introducción.

Moodle es un Sistema de Gestión de Cursos de Código Abierto (Open Source Course Management System, CMS), conocido también como Sistema de Gestión del Aprendizaje (Learning Management System, LMS) o como Entorno de Aprendizaje Virtual (Virtual Learning Environment, VLE).

Es una herramienta muy extendida sobre todo en el entorno de la educación. Dispone de características que le permiten ser utilizada por cientos de miles de estudiantes al mismo tiempo, tanto universitarios, como de educación infantil y primaria. Muchas instituciones lo utilizan como plataforma para formación en línea mientras que otras lo utilizan como apoyo a la formación presencial, este método se denomina blended learning.

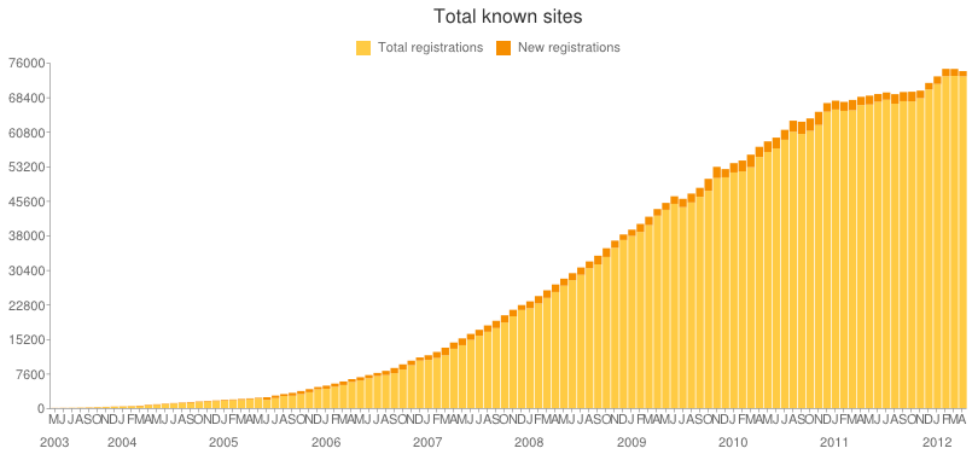


Tabla de Sitios Moodle Disponibles en el Mundo

En esta gráfica se observa la cantidad de sitios Moodle registrados en el mundo actualmente, con unos 6.144.522 cursos en total y 57.893.031 usuarios aproximadamente. España es el segundo País del mundo con más sitios registrados solamente superado por Estados Unidos.

2.2 Inseguridad en la plataforma Moodle.

Moodle es una plataforma insegura; de hecho, existen varias páginas Web dedicadas exclusivamente a los errores detectados en la herramienta [http://educhalk.org/blog/], y un subforo entero en la página oficial [http://moodle.org/mod/forum/view.php?id=7301]. La seguridad implementada en Moodle para proteger tanto las notas, como otro tipo de información, no impide ciertos ataques, entre ellos las inserciones directas de código SQL en la base de datos. Esta aplicación se diseñó para añadir una protección extra al sistema con cifrado y revisión de los cambios en las notas almacenadas.

Las inserciones de código SQL pueden ser de tres tipos: inserción, borrado y actualización. Será competencia del profesor asignado el comprobar tanto el borrado como la inserción ilegal de actividades con calificación asignada. Preguntar a los profesores cada vez que se cree una nueva actividad que admita calificación sería tedioso, por lo que el profesorado deberá conocer el número de actividades que ha colocado y corregido. Cualquier nota nueva que sea insertada asociada a una actividad creada por un intruso no será detectada por la aplicación, por lo que requerirá que los profesores avisen del problema al Administrador de la plataforma. Esas nuevas actividades aparecerán en la interfaz gráfica de calificación y aunque su estado sea oculto para los alumnos, en la interfaz de los profesores seguirán apareciendo y podrán ser detectadas por ellos mismos como entrada ilegal.

Consideramos que las inyecciones de código que actualicen la base de datos cambiando una nota existente serán las más frecuentes; por lo que nuestra aplicación estará diseñada para detectarlas, tanto si se realizan desde la interfaz gráfica de la plataforma Moodle como si se realizan directamente sobre la tabla de la base de datos.

2.3 Configuración Previa de Moodle.

Para utilizar Moodle basta con instalarlo en un servidor Web, en un ordenador personal o en un servidor proporcionado por una compañía de hospedaje de páginas Web. Una vez instalado y para optimizar la aplicación implementada se requerirá que se sigan las siguientes indicaciones.

Condiciones obligatorias para el correcto funcionamiento de la aplicación:

- Los Administradores del sitio, a partir de la versión 2.X, no son considerados un rol normal; son cuentas con permisos por defecto. Para evitar que la aplicación las detecte como intrusos será obligatorio que se les asocie al rol “Gestor”.

Condiciones necesarias para mejorar la eficiencia de la aplicación:

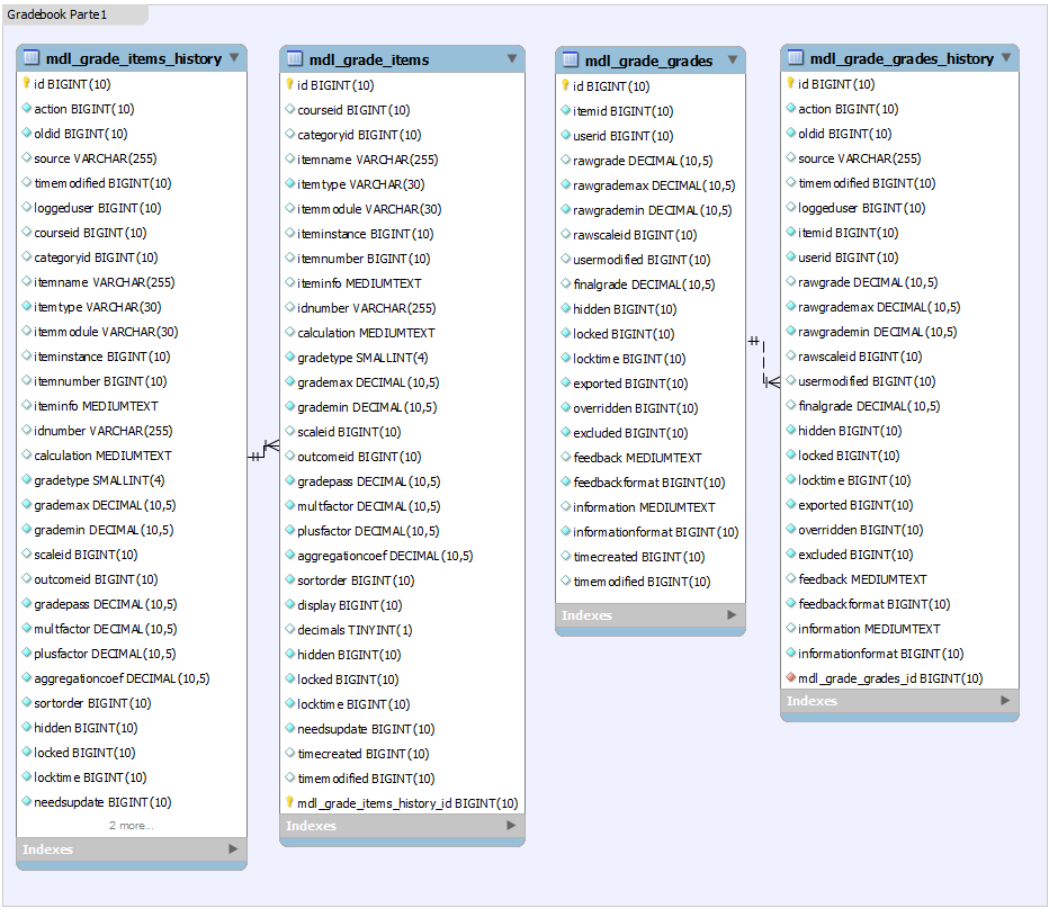
- Selección de una fecha de fin de curso. Una vez transcurridos 30 días de la finalización del curso la aplicación dejará de solicitar las notas dándolo por finalizado, considerando que las notas, salvo aquellas donde no haya sido resuelto el conflicto, almacenadas y cifradas en su base de datos son las correctas.
  Para poder activar el rastreo de finalización del curso será necesario que uno de los Administradores/Gestores de la plataforma, habilite el rastreo de finalización de los cursos y luego asigne una fecha de fin del curso.

2.4 Base de Datos Moodle.

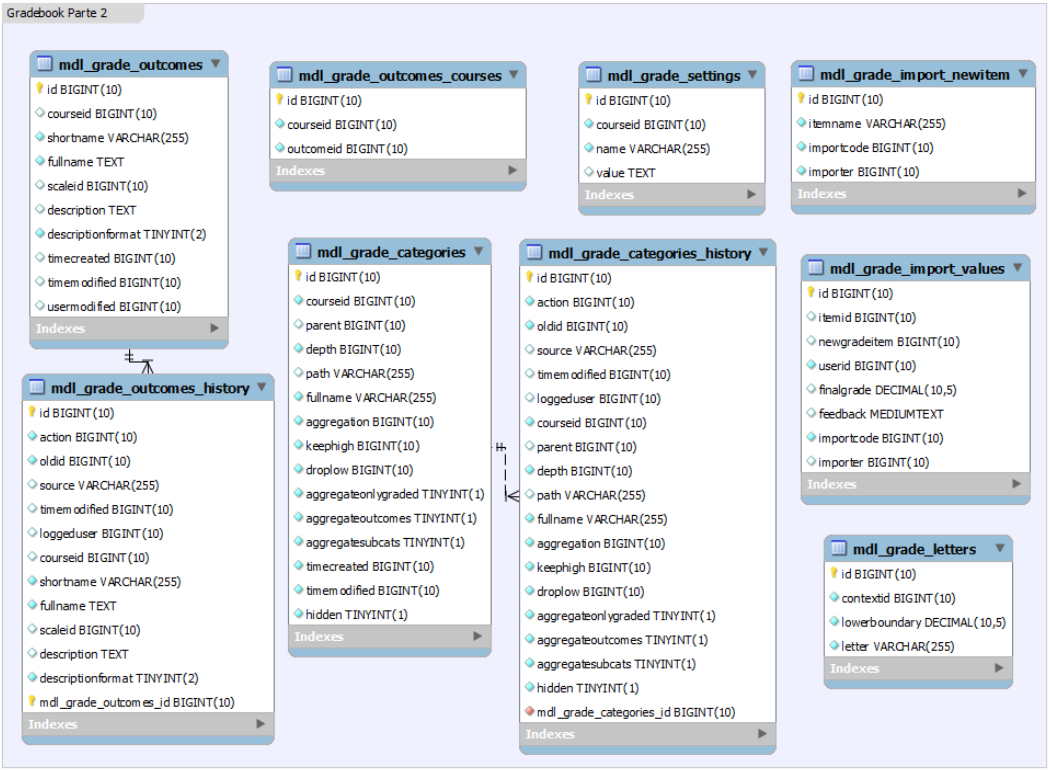
La base de datos de Moodle puede dividirse en varias áreas. A continuación se detallarán las más relevantes con respecto al proyecto, incluyendo las tablas que las componen y los campos contenidos en ellas.

2.4.1 Calificaciones

El sistema de calificaciones de Moodle comprende las siguientes tablas:



Tablas de Calificaciones en Moodle Parte 1



Tablas de Calificaciones en Moodle Parte 2

2.4.1.1 Tabla GRADE_ITEMS

Esta tabla almacena información sobre todas las actividades que pueden ser calificadas en los cursos. Si una actividad (por ejemplo, un cuestionario) tiene más de una forma de calificación asociada (por ejemplo varios resultados y notas numéricas), entonces le corresponderá más de una fila en esta tabla.

Los campos contenidos en esta tabla son los siguientes:

Campo	Descripción
id	**Identificador único de la actividad en la base de datos. (PK)**
courseid	**Identificador del curso al que pertenece la actividad calificable.**
categoryid	Categoría a la que pertenece el curso al que está asociada esta actividad.
itemname	**Nombre de la actividad.**
itemtype	**Tipo de actividad. Las más importantes son:**

	- **'mod': actividades en general.** - **'blocks': bloque actual.** - **'manual'.** - **'course': curso global.** - **'category': categoría global.**
itemmodule	Cuando la atividad es de tipo 'mod', el tipo de módulo: 'forum', 'quiz', 'csv', etc.
iteminstance	Identificador numérico del campo anterior, itemmodule.
itemnumber	Etiqueta única dentro de un curso que permite identificar un elemento de calificación; es útil para identificar los datos en exportaciones y para referirse a la nota en el cálculo de la calificación final.
iteminfo	Información y anotaciones asociadas a ese ítem.
idnumber	Número arbitrario proporcionado por el módulo responsable (opcional pero único en el curso).
calculation	Fórmula usada para procesar las notas y obtener la nota final asociada.
gradetype	Tipo de nota asociada. Valores: - 0 = Ninguna calificación asociada. - 1 = Valor numérico. - 2 = Escala de valores. - 3 = Nota en formato texto.
grademax	Nota máxima que se puede obtener en la actividad.
grademin	Nota mínima que se puede obtener en la actividad
scaleid	Cuando la nota es de tipo escala de valores, gradetype=2, identificador de la escala sobre la que se apoya la calificación.
outcomeid	Identificador de la competencia asociada a la actividad. Mide la evaluación del desempeño general del alumno; por ejemplo, puede referirse a la participación y a la asistencia.
gradepass	Nota mínima para aprobar la actividad.
multfactor	Factor por el que se multiplican las calificaciones para obtener la nota final.
plusfactor	Factor que se suma a las calificaciones para obtener la nota final.
aggregationcoef	Peso asociado a la nota de la actividad para el cálculo de la calificación final.
sortorder	Orden de creación de las actividades dentro de un curso. La nota global del curso tiene asociada el número 1.
display	Forma de mostrar en el interfaz gráfico de Moodle las calificaciones.

	Pueden mostrarse las notas reales, en porcentaje (haciendo referencia al mínimo o máximo), con letras (A, B, C, etc.), o solamente la nota global del curso.
hidden	Indica si la actividad está oculta para los alumnos o es visible para todos los participantes: - 0 = No está oculta. - 1 = Oculta. - > 1 = Número correspondiente a una fecha. La actividad permanecerá oculta hasta la fecha indicada.
locked	Indica si la actividad está bloqueada. Cuando una actividad está bloqueada cualquier nota asociada no puede ser modificada y cualquier petición de cambio será ignorada: - 0 = No está bloqueada. - > 0 = Número correspondiente a una fecha. Fecha en la actividad fue bloqueada.
locktime	Fecha en la que se debe bloquear automáticamente una actividad. - 0 = La actividad no se autobloquea. - > 0 = Número correspondiente a una fecha. La actividad se bloqueará en la fecha indicada.
needsupdate	Cuando este campo se pone a 1, todas las notas finales asociadas a esta actividad deben ser recalculadas. Si la actividad es de tipo curso, itemtype=course, algunas de las notas de actividades pertenecientes a ese curso deben ser recalculadas previamente.
timecreated	Fecha correspondiente a la primera vez que la actividad fue creada.
timemodified	Fecha de la última modificación de la actividad.

2.4.1.2. Tabla GRADE_GRADES

Esta tabla almacena una nota por cada actividad y usuario (alumno) existentes. La nota en crudo se almacena de la misma manera si se importa directamente desde un fichero con un formato determinado que si se almacena manualmente usando la interfaz gráfica de Moodle.

Las notas máxima, mínima, y la escala aplicadas para obtener la nota final, se almacenan asociadas a las notas individuales para que el profesor pueda variarlas en cada actividad si lo desea. Todos los resultados son normalizados y calculados para obtener la nota final, que estará relacionada con los campos máximo, mínimo y la escala correspondiente almacenados en la tabla grade_item en la fila que corresponda. La nota final será recalculada cada vez que haya un cambio en la nota en crudo o cambie algo en la fila a la que se refiera de la tabla grade_item.

El valor de la nota final obtenida será redondeado para mostrarlo en el interfaz gráfico de Moodle.

Campo	Descripción
id	Identificador único de la nota en la base de datos. (PK)
itemid	**Identificador de la actividad a la que pertenece la nota.**
userid	**Identificador del usuario (alumno) al que pertenece la nota.**
rawgrade	La calificación en crudo que el profesor introdujo en el sistema de forma manual o importándola desde un fichero.
rawgrademax	Valor máximo de la nota permitido cuando fue creada.
rawgrademin	Valor mínimo de la nota permitido cuando fue creada.
rawscaleid	Identificador de la escala en la que está apoyado el cálculo de la nota final.
usermodified	**Identificador del usuario que modificó la nota la última vez.**
finalgrade	**Nota final almacenada en el sistema, obtenida tras realizar los cálculos necesarios sobre el campo "rawgrade", por ejemplo aplicarle la escala correspondiente.**
hidden	Indica si la calificación está oculta para los alumnos o es visible para todos los participantes: - 0 = No está oculta. - 1 = Oculta. - > 1 = La calificación permanecerá oculta hasta la fecha indicada.
locked	Indica si la calificación está bloqueada: - 0 = No está bloqueada. - > 0 = Fecha en la actividad fue bloqueada.
locktime	Fecha en la que se debe bloquear automáticamente una calificación. - 0 = La calificación no se autobloquea. - > 0 = La calificación se bloqueará en la fecha indicada.
exported	Indica si la calificación ha sido exportada: - 0 = La calificación no ha sido exportada. - > 0 = Fecha correspondiente a la última exportación de la nota.
excluded	Indica si la calificación se debe tener en cuenta para el cálculo de la nota global del curso: - 0 = La calificación no ha sido excluida. - > 0 = Fecha correspondiente a la exclusión de la nota del cálculo de la nota global.

overridden	Indica si la calificación ha sido sobrescrita: - 0 = La calificación no ha sido remplazada. - > 0 = Fecha correspondiente a la última vez que se remplazó la nota.
feedback	Código opcional elegido por el profesor explícitamente para que devuelva como realimentación al escribir una nota.
feedbackformat	Formato de texto para el campo “feedback”.
information	Texto de información adicional y opcional para la nota.
informationformat	Formato de texto para el campo “information”.
timecreated	Fecha correspondiente a la primera vez que la calificación fue creada.
timemodified	**Fecha de la última modificación de la nota.**

2.4.1.3. Tabla GRADE_CATEGORIES

Esta tabla contiene información sobre las categorías usadas para agrupar las notas de las actividades pertenecientes a un curso. En cada categoría se indicará la forma de agrupar las notas para obtener la nota global del curso (media, mediana, suma…).

Campo	Descripción
id	Identificador único de la categoría en la base de datos. (PK)
courseid	Identificador del curso al que pertenece la categoría.
parent	Identificador de la categoría padre de la que hereda dentro de la jerarquía de categorías existentes.
depth	Nivel de profundidad en la que se encuentra situada la categoría dentro de la jerarquía existente (1, 2, 3…).
path	Muestra la ruta de categorías dentro de la jerarquía hasta llegar a la actual.
fullname	Nombre de la categoría de notas.
aggregation	Constante que representa una de las estrategias de agregación disponibles para obtener la nota global del curso (‘none’, ‘mean’, ‘median’, ‘sum’, etc.).
keephigh	Conservar sólo el número indicado en este campo de notas, empezando por las que tengan una calificación más alta, para calcular el resultado final.
droplow	Desechar el número indicado en este campo de notas, empezando por las que tengan una calificación más baja.
aggregateonlygraded	Agregar sólo las notas existentes.
aggregateoutcomes	Agregar los resultados junto a las notas normales.
aggregatesubcats	Agregar sólo las actividades pertenecientes de manera directa a la categoría o todas las actividades pertenecientes a las subcategorías excluyendo los totales.
timecreated	Fecha correspondiente a la primera vez que la categoría fue creada.
timemodified	Fecha de la última modificación de la categoría.

2.4.1.4. Tabla GRADE_OUTCOMES

Esta tabla contiene una serie de resultados en distintas competencias utilizados para evaluar al alumno que indican si ha demostrado entender la actividad o el curso realizados. Estas competencias miden el desempeño general del alumno, por ejemplo, pueden referirse a la participación y a la asistencia, siguiendo una serie de principios de evaluación. Por tanto, la calificación global de un curso puede ir acompañada de un conjunto de estas competencias específicas.

Campo	Descripción
id	Identificador único del posible resultado de una competencia en la base de datos. (PK)
courseid	Identificador del curso al que está asociado ese resultado. Si el valor almacenado fuera NULL el resultado de la competencia afectaría a todo el sistema.
shortname	Nombre corto o código utilizado para referirse al resultado de la competencia.
fullname	Nombre completo del resultado de la competencia.
scaleid	Identificador de la escala en la que se encuentran los posibles resultados de la competencia.
description	Descripción completa del resultado de la competencia.
timecreated	Fecha correspondiente a la primera vez que el resultado de la competencia fue creado.
timemodified	Fecha de la última modificación del resultado de la competencia
usermodified	Identificador del usuario que modificó por última vez el resultado.

2.4.1.5. Tabla GRADE_OUTCOMES_COURSES

Tabla que relaciona los resultados de las competencias disponibles en el sistema con un curso concreto para que se puedan aplicar en él.

Campo	Descripción
id	Identificador único de la relación entre los resultado de una competencia y los cursos en la base de datos. (PK)
courseid	Identificador del curso al que se le asigna el resultado.
outcomeid	Identificador de los resultados de la competencia que han sido asignados al curso.

2.4.1.6. Tabla GRADE_IMPORT_NEWITEM

Tabla que almacena información sobre el fichero utilizado para importar notas en una actividad o curso.

Campo	Descripción
id	Identificador único del fichero utilizado para importar calificaciones en la base de datos. (PK)
itemname	Nombre del fichero.
importcode	Código utilizado para la importación de calificaciones.

2.4.1.7. Tabla GRADE_IMPORT_VALUES

Tabla que contiene los valores de las notas importadas en el sistema a través de un fichero.

Campo	Descripción
id	Identificador único de la calificación importada. (PK)
itemid	Identificador de la actividad a la que pertenece la nota.
newgradeitem	Calificación en crudo que el profesor introdujo en el sistema importándola desde el fichero.
userid	Identificador del usuario que importó la nota.
finalgrade	Calificación final de la actividad importada desde el fichero.
feedback	Código opcional elegido por el profesor explícitamente para que devuelva como realimentación al escribir una nota.
importcode	Código utilizado para la importación de calificaciones.

2.4.1.8. Tabla GRADE_LETTERS

Esta tabla contiene la relación existente entre las notas numéricas habituales y las letras de calificación utilizadas en otros países. La equivalencia entre las letras y el valor numérico más alto y el más bajo es:

Letra	A	A-	B+	B	B-	C+	C	C-	D+	D	F
Más Alta	100,00	92,99	89,99	86,99	82,99	79,99	76,99	72,99	69,99	66,99	59,99
Más Baja	93,00	90,00	87,00	83,00	80,00	77,00	73,00	70,00	67,00	60,00	0,00

Campo	Descripción
id	Identificador único de la letra de calificación. (PK)
contextid	Identificador del contexto al que se le aplica la letra de calificación (‘sistema’, ‘categoría de cursos’, ‘curso’)
lowerboundary	La calificación numérica más baja equivalente a esa letra. La calificación numérica más alta se obtiene como la nota siguiente a la más baja del anterior nivel. Si no hubiera nivel siguiente, la nota más alta será equivalente a la almacenada en el campo grademax de la tabla grade_item.
letter	El valor de la letra de calificación que se mostrará en la interfaz gráfica de Moodle. Puede ser un carácter o una cadena de caracteres (por ejemplo, ‘OK’, ‘A’, ‘10%’, etc.).

2.4.1.9. Tablas HISTORY

Alguna de las tablas anteriormente explicadas posee una tabla asociada con el mismo nombre añadiéndole el sufijo "_history". Se usan como histórico de los datos almacenados en las tablas del mismo nombre, indicando los valores que han tomado sus campos a lo largo del tiempo, y que más tarde se podrán usar, por ejemplo, para realizar auditorías. El uso de estas tablas en lugar de los ficheros de log existentes en Moodle se justifica por la mejora en el rendimiento a la hora de localizar los cambios sufridos por las notas.

Las tablas pertenecientes a este grupo son:

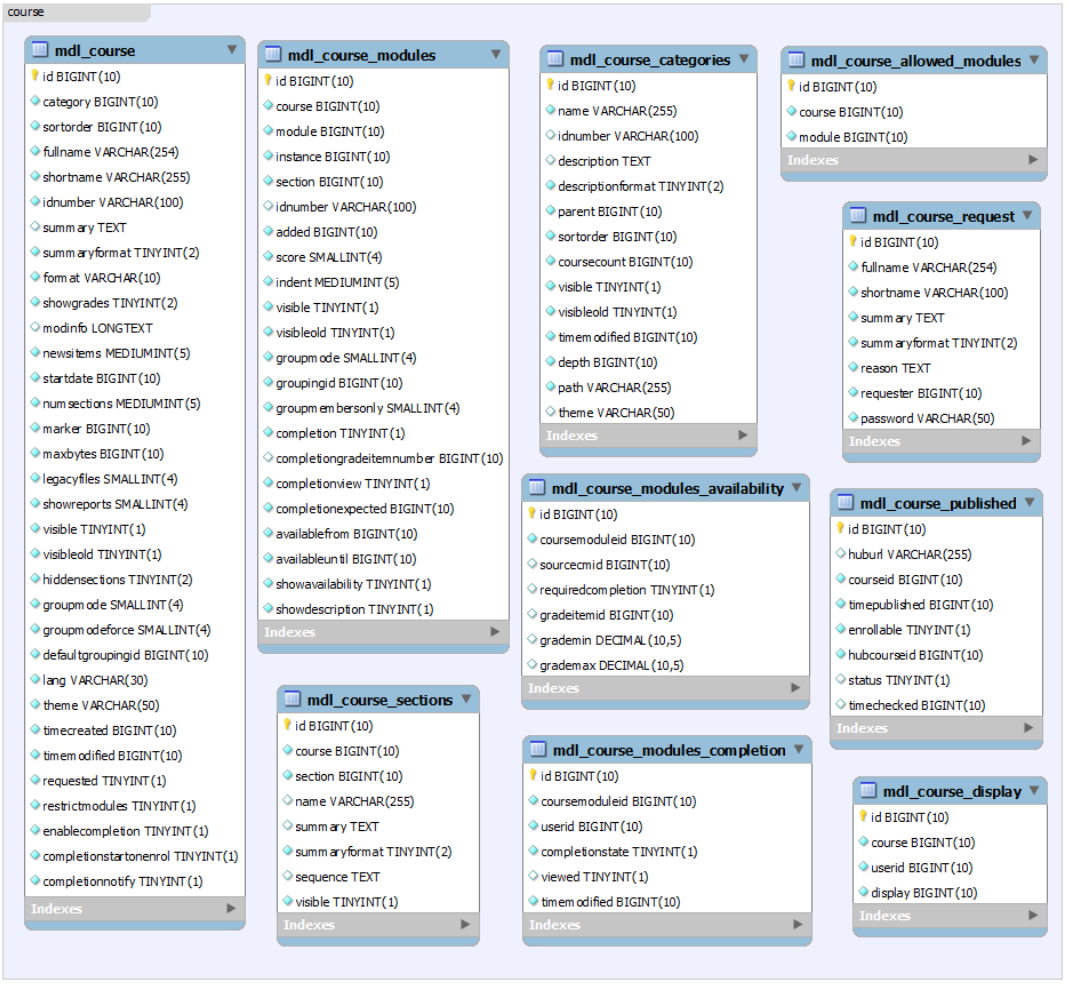
- grade_categories_history.
- grade_grades_history.
- grade_items_history.
- grade_outcomes_history.

Se hace copia de seguridad de todos los campos existentes en las tablas de la base de datos del mismo nombre añadiendo tres campos nuevos en cada una de las entradas:

Campo	Descripción
action	Acción utilizada para crear o modificar la entrada ('insert', 'update', 'delete').
oldid	Clave primaria de la entrada en la tabla original.
source	Extensión que originó el cambio. En nuestro caso 'gradebook'.

2.4.2 Cursos

Para explicar este apartado la información sobre cursos de Moodle se ha dividido en dos bloques en función de su contenido, la primera parte muestra la información sobre los cursos, sus secciones y las actividades contenidas y la segunda los criterios de finalización de los cursos.



Tablas de Cursos en Moodle

2.4.2.1. Tabla COURSE

Esta tabla contiene información de todos los cursos existentes en Moodle.

Campo	Descripción
id	**Identificador único del curso. (PK)**
category	Identificador de la categoría a la que pertenece el curso. Por defecto su valor es “Miscelánea”.
sortorder	Número utilizado para ordenar los cursos de acuerdo a su fecha de creación. Al primer curso creado le corresponde el número 1001.
fullname	**Nombre completo del curso. Se muestra en la parte superior de cada página del curso y en la lista de cursos de la página principal.**
shortaname	Nombre corto del curso. Muchas instituciones asignan nombres cortos a sus cursos para efectos administrativos. Los nombres cortos deben ser significativos puesto que se utilizarán en los diferentes lugares donde los que un nombre completo sería inadecuado, por ejemplo, en la línea "asunto" de un correo.
idnumber	Número identificativo de un curso. Usado única y exclusivamente cuando se compara el curso con un sistema externo a Moodle.
summary	Descripción resumida del curso.
summaryformat	Formato del resumen.
format	Formato de presentación del curso. Para conocer los tipos de formato ir a la tabla siguiente.
showgrades	Indica si se muestran las calificaciones de las actividades del curso a los estudiantes o si se mantienen ocultas.
modinfo	Información sobre las actividades.
newsitems	Número de noticias nuevas sobre las actividades que se muestra, como máximo, en la página principal del curso.
startdate	Fecha de inicio del curso.
numsections	Número de secciones, bloques, que tendrá el curso. Dividido según el formato en semanas o en temas.
marker	Indica si el curso está agregado a favoritos.

maxbytes	Número máximo de Bytes que puede subir un estudiante en cada entrega de una actividad.
legacyfiles	Términos y condiciones de uso del curso.
showreports	Mostrar informes de actividad.
visible	Indica la visibilidad del curso: - 1= El curso está visible para todos. - 0= El curso está sólo visible para los profesores.
visibleold	Estado previo de visibilidad.
hiddesections	Forma de mostrar las secciones ocultas: - 0= Las secciones ocultas se muestran de forma colapsada. - 1= Las secciones ocultas se muestran totalmente visibles.
groupmode	Indica si los participantes del curso se dividen en grupos. - 0= no se usan grupos. - 1= grupos separados, los estudiantes sólo podrán ver a los participantes de su propio grupo. - 2= grupos visibles para todos los participantes, los estudiantes podrán ver a todos los participantes del curso independientemente del grupo al que pertenezcan.
groupmodeforce	Forzar el modo de agrupación de los participantes. - 0= No forzar. - 1= Forzar. El modo de agrupación de los participantes del curso se aplica a todas las actividades del mismo.
defaultgroupingid	Identificador del grupo por defecto en el que se añadirán los participantes cuando sean matriculados.
lang	Idioma por defecto del curso.
theme	Estilo del Moodle.
timecreated	Fecha en la que el curso fue creado.
timemodified	Fecha en la que el curso fue modificado por última vez.
requested	Indica si la creación del curso ha sido solicitada al Administrador por parte de algún otro usuario.
restrictmodules	Las actividades no son visibles para los estudiantes.

enablecompletion	Si esta opción está activada, se activa el rastreo de finalización del curso.
completionstartonenrol	Indica que la finalización del curso se produce tras un tiempo determinado desde que el alumno fue matriculado.
completionnotify	Cuando un alumno ha acabado un curso, si esta opción está habilitada, se le envía una notificación.

Los posibles formatos de presentación de un curso son:

Formato	Descripción
LAMS	Este formato convierte la interfaz LAMS, Learning Activity Management System, en el aspecto central del curso. LAMS es un entorno de creación visual e intuitivo para la creación de secuencias de actividades de aprendizaje, las cuáles pueden incluir una serie de tareas individuales y trabajos en pequeños grupos.
SCORM	Este formato muestra un paquete SCORM, Sharable Content Object Reference Model, en la primera sección de la página principal del curso. SCORM es un conjunto de estándares y especificaciones que permite crear objetos pedagógicos estructurados.
Social	Este formato se orienta en torno al foro central, que aparece en la página principal. Además, podría utilizarse como tablón de anuncios de un departamento.
Temas	El curso se organiza en secciones o temas, cada uno con una serie de actividades.
Semanal	El curso se organiza por semanas, con fecha de inicio y fin, cada una con sus propias actividades, aunque algunas de ellas pueden durar más de una semana antes de cerrarse.
CSS/Sin tablas	El curso se organiza semana por semana, sin usar tablas en el formato.

2.4.2.2. Resto de Tablas de Curso

Del resto de tablas de cursos, puesto que no son relevantes para la aplicación, sólo se indicará su funcionalidad:

- Tabla COURSE_MODULES: Contiene información sobre las actividades de un curso.
- Tabla COURSE_SECTIONS: Contiene la información sobre los bloques existentes en el curso, ya sea por semanas o por temas.
- Tabla COURSE_CATEGORIES: Contiene información sobre la categoría a la que está asociado el curso.
- Tabla COURSE_MODULES_AVAILABILITY: Contiene información sobre las actividades y los requisitos para pasarlas o realizarlas.
- Tabla COURSE_MODULES_COMPLETION: Contiene el estado de las diferentes actividades del curso.
- Tabla COURSE_ALLOWED_MODULES: Relaciona las actividades en la tabla course_modules con los cursos de la tabla course.
- Tabla COURSE_REQUEST: Contiene información sobre los cursos solicitados para su creación al administrador por un usuario.
- Tabla COURSE_PUBLISHED: Contiene información sobre la publicación de los cursos.
- Tabla COURSE_DISPLAY: Indica si un curso está visible para todos los participantes o si está oculto para los alumnos.

Tablas de Criterios de Finalización de los Cursos en Moodle

Una de las posibilidades que ofrece Moodle es el uso de ciertos criterios para rastrear la finalización de los cursos. Esta opción es opcional y requiere que el Administrador active el rastreo globalmente en el sistema y particularmente en cada uno de los cursos donde deba aplicarse.

Encontramos siete criterios distintos que permiten la finalización de un curso, puede ser obligatorio que se cumplan todos, o sólo alguno, dependiendo de lo que el Administrador haya indicado:

- 1. Autocompletar (self): El alumno, en cualquier momento, decidirá cuando finalizar el curso.
- 2. Fecha de Finalización (date): El curso permanecerá abierto hasta la fecha elegida. Una vez se alcance, la fecha el curso se marcará como finalizado automáticamente para todos los alumnos.
- 3. Desmatriculación (unenrol): Cuando un alumno se da de baja en el curso, automáticamente el curso se marca como completado para él.
- 4. Completar ciertas Actividades (mod): El curso finalizará cuando el alumno complete la o las actividades indicadas en este criterio.
- 5. Tras un tiempo determinado desde la matriculación (enrolperiod): El curso finalizará trascurrido un tiempo específico desde la matriculación del alumno.

- 6. Nota de corte (grade): El curso finalizará cuando el alumno consiga una nota que exceda la nota mínima necesaria para aprobar el curso.
- 7. Finalización Manual (manual): El curso finaliza para un alumno cuando así lo indique el profesor o el profesor sin permisos de edición responsable del curso o un gestor de la plataforma. Puede elegirse que cualquiera de ellos lo haga, o que necesite de la conformidad de todos.

Para más información sobre los criterios de finalización véase el apartado 2.5.1.4 Tabla COURSE_COMPLETION_CRITERIA.

Cuando un curso se cierra existen tres posibles estados finales distintos:

- Cierre privado (Private closing): cambiará la visibilidad del curso, el nuevo estado será "no visible" y sólo los profesores encargados y los gestores podrán acceder a él.
- Cierre protegido (Protected closing): la visibilidad del curso no cambia. Los alumnos matriculados en él podrán seguir accediendo al curso pero no se les permitirá modificar nada.
- Cierre público (Public closing): un curso visible para cualquiera, seguirá permitiendo la entrada a los invitados aunque no se les permitirá modificar nada.

2.4.2.3. Tabla COURSE_COMPLETION_AGGR_METH

Esta tabla indica la forma en la que se completa cada uno de los cursos cuando el rastreo de finalización está activado.

Campo	Descripción
id	Identificador único del criterio de finalización. (PK)
course	Identificador del curso al que está asociado el criterio de finalización.
criteriatype	Número entero que representa un rol que tenga permitido marcar un curso como finalizado para un estudiante en concreto o una actividad que tras ser completada marque el curso como finalizado para el alumno. Si el campo tuviera el valor NULL el criterio de finalización afectaría a todos los participantes a la vez.
method	Forma en la que se aplican los distintos métodos de finalización del curso: - 1='all', todos los criterios seleccionados para la finalización deben cumplirse para que el curso se cierre. - 2='any', el primero de los criterios asociados que se cumpla cerrará el curso automáticamente. - 3='fraction', cuando se complete el tanto por ciento indicado el curso se cerrará automáticamente. - 4='unit', cuando se complete el 100% del curso el curso finaliza.

Value	Relacionado con el campo method: - NULL = si el método es 'all' o 'any'. - Un valor entre 0 y1 = si el método es 'fraction'. - Entero > 0 = si el método es 'unit'.

2.4.2.4. Tabla COURSE_COMPLETION_CRIT_COMPL

Esta tabla contiene la relación entre los usuarios y los métodos de finalización que han completado.

Campo	Descripción
id	Identificador único de la relación entre los usuarios y el criterio de finalización que han completado. (PK)
userid	Identificador del usuario que ha completado el criterio.
course	Identificador del curso del que se ha completado el criterio.
criteriaid	Identificador del criterio que se ha completado.
gradefinal	Nota final obtenida en el curso en caso de que el tipo de criterio sea 'grade'.
unenroled	Fecha en el que el usuario se dio de baja en caso de que el tipo de criterio sea 'unenrol'.
deleted	Indica si el curso al que está asociada esta relación ha sido borrado.
timecompleted	Fecha en la que el usuario completó el criterio.

2.4.2.5. Tabla COURSE_COMPLETION_CRITERIA

Tabla que almacena los criterios de finalización que tiene asociados un curso.

Campo	Descripción
id	Identificador único del criterio de finalización. (PK)
course	**Identificador del curso al que se asocia el criterio de finalización.**
criteriatype	**Tipo de criterio de finalización:** - **1= 'self'.** - **2= 'date'.** - **3= 'unenrol'.** - **4= 'mod'.** - **5= 'enrolperiod'.** - **6= 'grade'.**

	- **7= 'role'.**
module	Si criteriatype='mod'. Nombre de la actividad, por ejemplo, 'scorm', 'quiz', 'feedback'…
moduleinstance	Si criteriatype='mod'. Identificador de la actividad.
enrolperiod	Si criteriatype='enrolperiod'. Número de segundos tras la inscripción.
timeend	**Si criteriatype='date'. Fecha en la que finaliza el curso.**
gradepass	Si criteriatype='grade'. Nota mínima requerida para completar el curso.
role	Si criteriatype='role'. Tipo de rol que puede marcar el curso como completado; puede ser un gestor, un profesor o un profesor sin permisos de edición.

2.4.2.6. Tabla COURSE_COMPLETION_NOTIFY

Esta tabla contiene datos de aquellos roles a los que se envía una notificación cuando uno de los criterios de finalización del curso se ha completado.

Campo	Descripción
id	Identificador único del aviso de notificación. (PK)
course	Identificador del curso al que está asociado el aviso.
role	Identificador del rol al que se le enviará una notificación cuando se complete un criterio en ese curso.
message	Mensaje en HTML que se enviará al usuario que tenga ese rol en el curso.
timesent	Fecha en la que la notificación fue enviada.

2.4.2.7. Tabla COURSE_COMPLETIONS

Tabla que guarda el momento en el que un usuario completa uno de los criterios de finalización activados en un curso.

Campo	Descripción
id	Identificador único del momento en el que un usuario completa uno de los criterios de finalización activados en un curso. (PK)
userid	Identificador del usuario que ha completado el criterio.
course	Identificador del curso al que estaba asociado el criterio.
deleted	Indica si el curso al que está asociado el registro ha sido borrado

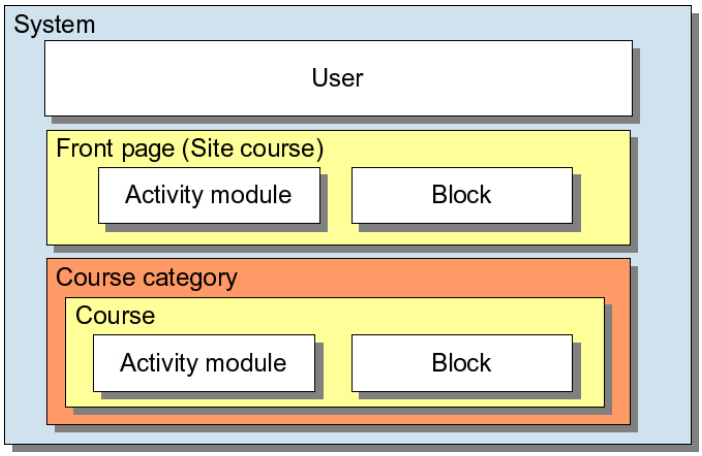
timenotified	Fecha en que la que el usuario fue informado.
timeenroled	Fecha en la que el usuario fue matriculado en el curso.
timecompleted	Fecha en la que el usuario complete el criterio.

2.4.3 Contextos

Un contexto (context) es un espacio de Moodle en el cual se pueden asignar roles a los usuarios. Al asignar un rol a un usuario en un contexto determinado le estamos garantizando los permisos propios de ese rol en ese contexto y en todos los contextos de rango inferior. Por ejemplo, si asignamos a un usuario el rol de profesor a una categoría de cursos, tendrá ese rol para todos los cursos que contenga la categoría; asimismo, si se asigna a un usuario el rol estudiante en un curso, poseerá ese rol en todo el curso incluyendo todos los bloques y actividades del curso.

Los contextos se organizan de forma jerárquica y sus permisos se transfieren desde los contextos 'superiores' a los 'inferiores'. El orden jerárquico es el siguiente:

Nombre del Contexto	Nivel Superior	Espacio del Contexto	Nivel del Contexto
CONTEXT_SYSTEM	----	Permisos en todo el sistema.	10
CONTEXT_USER	System	Permisos para ver perfiles de Usuario y para modificar el propio.	30
CONTEXT_COURSECAT	System	Permisos para toda la categoría.	40
CONTEXT_COURSE	**Categoría**	**Permisos para todo el curso.**	**50**
CONTEXT_GROUP	Curso	Permisos sobre todo el grupo de usuarios.	60
CONTEXT_MODULE	Curso Portal del Sitio	Permisos para la actividad.	70
CONTEXT_BLOCK	Curso Portal del Sitio	Permisos para el bloque de actividades.	80



Tablas de Contextos en Moodle

2.4.3.1. Tabla CONTEXT

Tablas de Contextos en Moodle

Esta tabla contiene los distintos contextos (espacios) que posee Moodle a los que se les puede asociar un rol concreto.

Campo	Descripción
id	Identificador único del contexto. (PK)
contextlevel	**Nivel del contexto (Los valores se encuentra definidos en la tabla superior).**
instanceid	**Identificador de la categoría, curso, actividad o bloque, dependiendo del nivel del contexto asociado.**
path	Muestra la ruta partiendo del sistema y siguiendo la jerarquía para alcanzar la instancia a la que se refiera el contexto.

depth	Profundidad del campo path.

2.4.4 Roles

Un rol consiste en una lista de permisos sobre cada una de las funcionalidades existentes en Moodle, como pueden ser borrar discusiones, añadir actividades, etc., en uno de los contextos predefinidos.

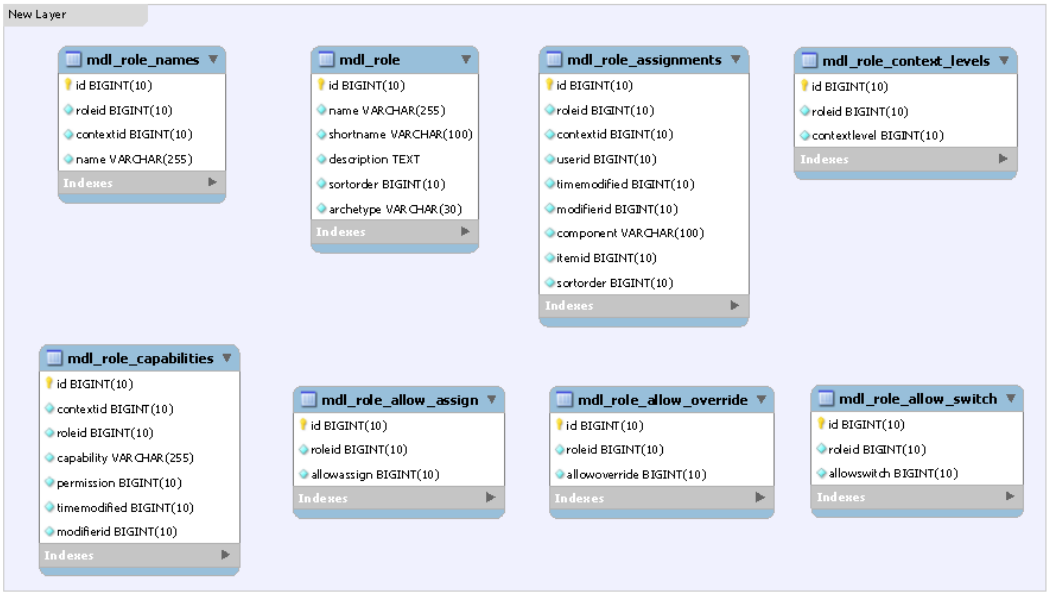
Existen ocho roles por defecto en Moodle:

Id	Nombre Rol	Descripción
1	**Gestor (Manager)**	**Pueden acceder a cualquier curso existente en el sistema y modificarlo.**
2	Creador de curso (Course creator)	Pueden crear nuevos cursos, una vez creado el curso queda matriculado como Profesor (Editing Teacher).
3	**Profesor (Editing teacher)**	**Pueden realizar cualquier acción dentro de un curso, incluyendo cambiar actividades y calificar a los estudiantes.**
4	**Profesor sin permiso de edición (Teacher)**	**Pueden realizar cualquier tipo de acción dentro del curso y calificar a los estudiantes, sin embargo, no tienen permitido modificar las actividades.**
5	Estudiante (Student)	Poseen menos permisos que los profesores, entre ellos se incluyen rellenar actividades, participar en los foros…
6	Invitado (Guest)	Poseen los privilegios mínimos y normalmente no están autorizados para participar en los cursos.
7	Usuario identificado (User)	Cualquier usuario identificado en cualquier parte del sistema. Usado si se quieren dar permisos especiales a todos los usuarios que pueden autentificarse en el sistema.
8	Usuario identificado en la página principal (Front page)	Cualquier usuario identificado en la página principal. Usado en caso de que se quieran dar permisos especiales a todos los usuarios que cuando estén en la página principal.

Desde la versión 2.0 de Moodle, Administrador dejó de ser un rol normal como los 8 definidos en la tabla anterior. Desde la versión 2.x es un caso especial; ya no se pueden definir los permisos que tienen, se le asignan automáticamente al instalar el sistema.

La mejor forma de dar permisos a los usuarios sobre todo el sistema es asignarles el rol de Gestor. Gestor es un rol real que permite hacer las operaciones de identificación necesarias para en el módulo a implementar en el Trabajo de Fin de Grado.

El sistema de roles en Moodle comprende las siguientes tablas:



Tablas de Roles en Moodle

2.4.4.1 Tabla ROLE

Esta tabla contiene los diferentes roles existentes en Moodle. Cuando se instala el sistema, se crean los roles por defecto, ya explicados en la tabla anterior. En cualquier momento se pueden añadir otros roles.

Campo	Descripción
id	**Identificador único del rol en el sistema. (PK)**
name	Nombre del rol.
shortname	Nombre corto usado para referirse al rol.
description	Descripción del rol. Indica que funcionalidades puede realizar en Moodle y en que contexto.
sortorder	Orden en el que se han añadido los roles a la tabla.

achertype	Determina los permisos por defecto de un rol.

2.4.4.2 Tabla ROLE_ASSIGNMENTS

En esta tabla se le asigna a un usuario un rol (una serie de permisos) en un contexto determinado.

Campo	Descripción
id	Identificador único de la asociación entre el usuario, el rol y el contexto. (PK)
roleid	**Identificador del rol con el que el usuario ha sido matriculado en el contexto.**
contextid	**Identificador del contexto en el que el usuario ha sido matriculado con ese rol.**
userid	**Identificador del usuario que ha sido matriculado con ese rol en ese contexto.**
timemodified	Fecha en la que se asoció al usuario con el rol y el contexto
modifierid	Identificador del usuario que asoció al usuario con el rol y el contexto.
component	----
itemid	[Opcional] Identificador de la actividad a la que está asociada el rol.
sortorder	Orden en el que se crearon los roles.

2.4.4.3 Tabla ROLE_CAPABILITIES

Esta tabla contiene los permisos para cada función en cada una de las definiciones de los roles (si el contexto es todo el sistema) o la anulación de un rol (si el contexto es cualquier otro).

Campo	Descripción
id	Identificador único de los permisos de la función para ese rol. (PK)
contextid	Identificador del contexto al que afecta la definición de esa función.
roleid	Identificador del rol al que afecta la definición de esa función.
capability	Nombre de la función que se está definiendo.
permission	Permisos que se le asignan.
timemodified	Fecha en la que se modificaron los permisos de la función para ese rol.

modifierid	Identificador del usuario que modificó los permisos de la función en ese rol.

2.4.4.4 Tabla ROLE_CONTEXT_LEVELS

Los niveles de contexto donde los roles pueden ser asignados.

Campo	Descripción
id	Identificador único de la relación entre el rol y el nivel de contexto al que se ha aplicado. (PK)
roleid	Identificador del rol.
contextlevel	Nivel del contexto.

2.4.4.5 Tabla ROLE_NAMES

Esta tabla contiene el nombre de cada una de las asociaciones entre los niveles del contexto y los roles.

Campo	Descripción
id	Identificador único de la relación entre el rol y el nivel de contexto al que se ha aplicado. (PK)
roleid	Identificador del rol.
contextlevel	Nivel del contexto.
name	Nombre de la relación entre el nivel del contexto y los roles.

2.4.4.6 Tabla ROLE_ALLOW_ASSIGN

Esta tabla contiene información sobre que roles pueden asignar otros roles a usuarios en el sistema. Por ejemplo, un Profesor puede matricular usuarios con el rol Alumno en sus cursos.

Campo	Descripción
id	Identificador único de la relación entre el rol y el rol que puede asignar. (PK)
roleid	Identificador del rol.
allowassign	Identificador del rol que puede asignar.

2.4.4.7 Tabla ROLE_ ALLOW_OVERRIDE

Esta tabla contiene la relación de roles que pueden anular a otros y que se usen sus permisos en vez de los del otro rol.

Campo	Descripción
id	Identificador único de la relación entre el rol y el rol al que puede anular. (PK)
roleid	Identificador del rol.
allowoverride	Identificador del rol al que puede anular.

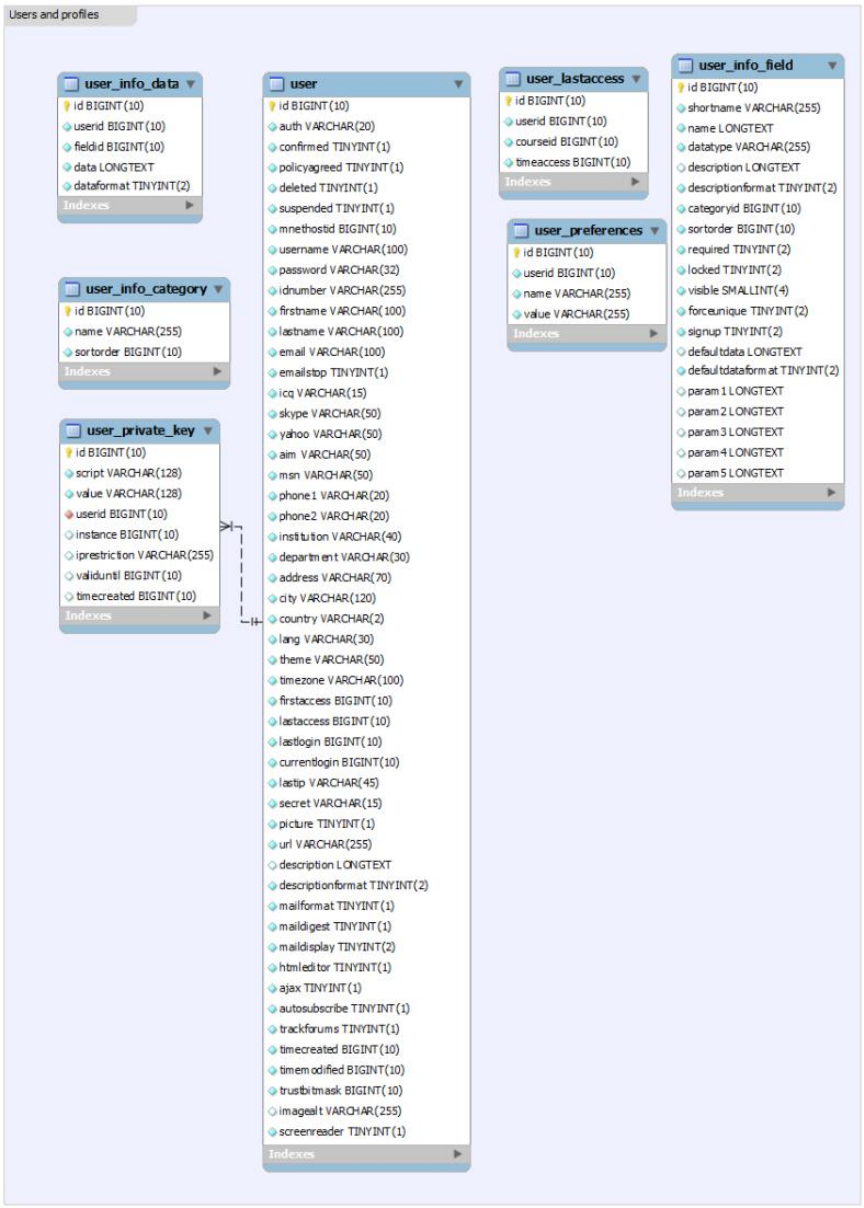
2.4.4.8 Tabla ROLE_ ALLOW_ SWITCH

Esta tabla contiene la relación de roles que pueden intercambiarse por otros en determinados contextos. Por ejemplo, un profesor en su curso puede entrar como alumno para ver la plataforma de la misma forma en la que la ven sus estudiantes.

Campo	Descripción
id	Identificador único de la relación entre el rol y el rol por el que puede intercambiarse. (PK)
roleid	Identificador del rol.
allowoverride	Identificador del rol por el que puede intercambiarse.

2.4.5 Usuarios

La información de los Usuarios en Moodle se encuentra contenida en las siguientes tablas:



Tablas de Usuarios en Moodle

2.4.5.1 Tabla USER

Esta tabla contiene información sobre todos los usuarios registrados en la plataforma Moodle.

Campo	Descripción
id	**Identificador único del usuario en el sistema. (PK)**
auth	Manera en la que se creó la cuenta de usuario. Los valores más comunes son: - Manual: Creación manual de la cuenta desde interfaz gráfica. - LDAP: Permite comprobar la existencia de los usuarios contra un servidor LDAP con los datos; de esta forma no hay que mantener actualizada la información en dos servidores separados (Moodle y LDAP). - Email: Permite a los usuarios crearse una nueva cuenta de usuario siempre que dispongan de una cuenta de correo electrónico; es decir, pueden automatricularse en la plataforma. No obstante, el acceso a los cursos todavía estaría controlado. - Enrolment keys: Distintas opciones de autentificación.
confirmed	Indica que la información de usuario está almacenada en el sistema pero todavía no se ha completado su registro; es necesaria una confirmación por parte del usuario.
policyagreed	Indica si el usuario ha leído y aceptado las condiciones de uso de la plataforma.
deleted	Indica si la cuenta ha sido borrada del sistema y el usuario ya no tiene permitida la entrada.
suspended	Indica si la cuenta ha sido suspendida y el usuario no tiene permitida la entrada en el sistema.
mnethostid	Permite al usuario acceder a otro sistema Moodle utilizando el mismo nombre y contraseña que en el actual.
username	Nombre del usuario en Moodle, login, no permite mayúsculas.
password	Contraseña del usuario en Moodle. La contraseña debe tener al menos 8 carácter(es), al menos 1 dígito(s), al menos 1 minúscula(s), al menos 1 mayúscula(s), al menos 1 carácter(es) no alfanumérico(s).

idnumber	--------
firstname	**Nombre real del Usuario.**
lastname	**Apellido(s) reales del Usuario.**
email	**Email del Usuario al que se enviarán los avisos.**
emailstop	Uso del correo electrónico para recibir avisos de los cursos en los que esté matriculado: - 0=El uso del correo electrónico está activado. - 1=El uso del correo electrónico está desactivado.
icq	[Opcional] Cuenta de ICQ.
skype	[Opcional] Cuenta de Skype.
yahoo	[Opcional] Cuenta de Yahoo.
aim	[Opcional] Cuenta de Aim.
msn	[Opcional] Cuenta de Microsoft Messenger.
phone1	[Opcional] Teléfono del Usuario.
phone2	[Opcional] Otro teléfono del Usuario.
institution	Institución a la que pertenece el usuario.
departament	Departamento al que pertenece el usuario.
addres	Dirección del usuario.
city	Ciudad en la que reside el usuario.
country	País en el que vive el usuario.
lang	Idioma por defecto del usuario.
theme	Tema de Moodle que usa el Usuario.
timezone	Zona horaria del usuario.
firstaccess	Fecha del primer acceso del Usuario a la plataforma Moodle.
lastaccess	Fecha del último acceso del Usuario a la plataforma Moodle.
lastlogin	Ultima vez que el usuario se autentificó en la plataforma Moodle.
currentlogin	Tiempo que ha estado conectado el Usuario al sistema en el último acceso.
lastip	IP desde la que ha accedido el Usuario la última vez que accedió al sustema.
secret	Cuando un Usuario olvida su contraseña y solicita que se le envíe

	una nueva a su correo, la contraseña enviada se almacena en este campo.
picture	Indica si el Usuario tiene una imagen o no almacenada en su perfil.
url	URL de la página de inicio del usuario.
description	Descripción del usuario; más tarde se mostrará en su perfil para que otros usuarios puedan consultarla.
descriptionformat	Formato de la descripción.
mailformat	Formato del email.
maildigest	Forma de enviar los avisos sobre nuevas entradas en los foros a los usuarios. - 0= Sin resumen. Se le envía al usuario un solo email por cada nueva entrada en el foro. - 1= Completo. Se le envía al usuario un email diario con todas las entradas completas del día. - 2= Asuntos. Se le envía al usuario un email diario solamente con los asuntos de las entradas del día.
maildisplay	Mostrar el correo del Usuario al resto de participantes de la plataforma. - 0= Ocultar email a todos; - 1= Permitir que cualquiera vea el email. - 2= Permitir ver el email sólo a otros participantes del curso.
htmleditor	Permitir la edición de mensajes en formato HTML.
ajax	Usar Ajax para las características Web avanzadas.
autosubscribe	Suscribir automáticamente a los usuarios a los foros del curso. - 0= Suscribir al usuario al foro cuando envíe un mensaje. - 1= No suscribir automáticamente a los foros a los usuarios.
trackforums	Activar el rastreo de foro, se marcan los mensajes como ya leídos cuando se acceda al hilo.
timecreated	Fecha en la que el usuario fue creado.
timemodified	Fecha en la que el usuario fue modificado por última vez.
trustbitmask	Máscara de bits usada en la creación del informe general de seguridad.
imagealt	Texto alternativo a una imagen, usado en el lector de pantallas para personas con discapacidad visual.

screenreader	Activar o desactivar el lector de pantalla (accesibilidad para personas con discapacidad visual).

2.4.5.2 Resto de Tablas de Usuarios

Del resto de tablas de usuarios, puesto que no son relevantes para la aplicación, sólo se indicará su funcionalidad:

- Tabla USER_INFO_FIELD: en esta tabla se almacenan campos extra para la cuenta de usuario en caso de que el Administrador los necesite.
- Tabla USER_INFO_DATA: valor asignado a cada uno de los campos creados en la tabla user_info_fields.
- Tabla USER_INFO_CATEGORY: permite colocar los campos de la tabla user_info_fields en categorías.
- Tabla USER_PRIVATE_KEY: contiene un script con un certificado de seguridad para la autentificación de los usuarios.
- Tabla USER_LASTACCESS: almacena la fecha del último acceso del usuario, de esta forma se podrá indicar cuanto tiempo ha pasado desde su última visita a la plataforma.
- Tabla USER_PREFERENCES: almacena las preferencias de los usuarios.

3 SEGURIDAD

3.1 Introducción

La tendencia, cada vez más dominante, hacia la interconectividad y la interoperabilidad de las redes, de las máquinas de computación, de las aplicaciones, e incluso, de las empresas, ha situado a la seguridad de los sistemas de información como un elemento central en todo el desarrollo de la sociedad.

Su ámbito de aplicación abarca el desarrollo, la integración, la operación, la administración, el mantenimiento y la evolución de los sistemas y las aplicaciones, es decir, todo el ciclo de vida de los productos o unidades de negocios.

La seguridad de los sistemas de información es una disciplina en continua evolución. La meta final de la seguridad es permitir que una organización cumpla con todos sus objetivos de negocio o misión, implementando sistemas que tengan un especial cuidado y consideración hacia los riegos relativos a las TIC de la organización, a sus socios comerciales, clientes, Administración pública, suministradores…

3.1.1 Conceptos Básicos Generales

- Seguridad de la Información: Medidas preventivas y reactivas del hombre, de las organizaciones y de los sistemas tecnológicos que permiten resguardar y proteger la información buscando mantener la confidencialidad, la disponibilidad y la integridad de la misma. Este concepto no debe ser confundido con el de Seguridad Informática.
- Seguridad Informática: Área de la informática que se enfoca en la protección de la infraestructura computacional y todo lo relacionado con esta, incluyendo la información contenida.
- Tecnologías de la información y la comunicación (TIC): Conjunto de tecnologías que permiten la adquisición, producción, almacenamiento, tratamiento, comunicación, registro y presentación de información, en forma de voz, imágenes y datos contenidos en señales de naturaleza acústica, óptica o electromagnética.
- Riesgo: Todo tipo de vulnerabilidades, amenazas que pueden ocurrir sin previo aviso y producir numerosas pérdidas para las empresas.
- Criptología: disciplina científica que se dedica al estudio de la escritura secreta; es decir, estudia los mensajes que, procesados de cierta manera, se convierten en difíciles o imposibles de leer por entidades no autorizadas.

Actualmente, la criptología moderna se enfoca en el diseño y evaluación de un creciente conjunto de métodos y técnicas para la protección de la información. Consta de tres partes:

  - Criptografía: trata del diseño de algoritmos, protocolos y sistemas que se utilizan para proteger la información contra amenazas específicas.

  - Criptoanálisis: trata de descifrar las comunicaciones cifradas sin conocer las claves adecuadas.
  - Estenografía: trata de ocultar la información y/o las comunicaciones en sí, en vez de ocultar sólo su contenido o significado, como hace la criptografía.
- Cifrado: procedimiento que, utilizando un algoritmo con cierta clave, transforma un mensaje, sin atender a su estructura lingüística o significado, de tal forma que sea incomprensible o, al menos, difícil de comprender, a toda persona que no tenga la clave secreta del algoritmo que se usa para poder descifrarlo.
- Texto en claro o texto plano: cualquier tipo de información que resulta legible y comprensible. Mensaje que va a ser cifrado.
- Criptograma: cualquier información que se encuentre convenientemente cifrada y no resulte legible ni comprensible más que para el destinatario legítimo de la misma.

3.1.2 Criptoanálisis

- Criptoanálisis: parte de la criptología que se dedica al estudio de sistemas criptográficos con el fin de encontrar sus debilidades y romper su seguridad sin tener el conocimiento de la información secreta. Se ocupa de conseguir capturar el significado de textos cifrados sin tener autorización para ello. Podríamos decir que el criptoanálisis tiene un objetivo opuesto al de la criptografía. Su objetivo es buscar el punto débil de las técnicas criptográficas para explotarlo y así reducir o eliminar la seguridad que teóricamente aportaba dicha técnica.
- Ataque: Cualquier intento de criptoanálisis en un sistema que intente explotar algún tipo de debilidad del mismo; tendrá éxito cuando el atacante consiga romper la seguridad que la técnica criptográfica aporta al sistema. Pueden ser de dos tipos:
  - Ataques al propio sistema: en los que se pretende deducir una debilidad en el diseño del sistema.
  - Ataques a la implementación del sistema: donde la debilidad que se pretende obtener proviene de alguna de las aplicaciones; este tipo de ataques suele tener más éxito.
- Criptosistemas: sistemas criptográficos. La seguridad que proporciona un criptosistema se puede clasificar en dos grandes categoría:
  - Seguridad incondicional: No importa que potencia de computación tenga disponible el atacante, ya que el criptosistema no puede romperse.
  - Seguridad computacional o condicional: En este caso, el criptosistema sí puede romperse utilizando una determinada potencia de computación o bien resolviendo algún problema de los considerados muy difíciles. Puede significar una de las dos siguiente cosas:
    - El criptosistema o cifrado no puede romperse en el caso en el que el oponente posea recursos de computación limitados; por ejemplo, que el tiempo necesario para realizar los cálculos sea mayor que la edad del

universo, o que se necesite más tiempo para romperlo que el tiempo que necesita conservar la confidencialidad el documento cifrado.

- Seguridad probable: se proporciona la evidencia de la seguridad computacional, reduciendo la seguridad del criptosistema a algún problema matemático bien estudiado y que sea muy difícil, como por ejemplo, el problema de la factorización de números enteros o el logaritmo discreto.

El criptoanálisis utiliza métodos matemáticos para probar que la implementación de la protección de la información no es capaz de alcanzar un objetivo de seguridad o de resistir un ataque de una lista de amenazas dadas en la especificación del diseño de seguridad. Esto ocurre cuando los parámetros de seguridad declarados se han sobrestimado o si las interrelaciones entre las diferentes amenazas no se han entendido correctamente o no se han tenido en cuenta.

Los criterios de diseño para determinar la política de seguridad obtenidos a partir de un ataque permiten crear un sistema que sea inmune contra él. La criptografía trata de probar que los diseños obtenidos son seguros utilizando todo el conocimiento disponible acerca de posibles ataques. El criptoanálisis cuidadoso examina posibles amenazas reales para encontrar nuevos ataques y probar que el diseño no es seguro, sino que es "rompible". Muchos criptosistemas como DES, RC4 o MD5 que se creían seguros, se han visto atacados y se han dejado de utilizar para proteger sistemas.

El criptoanálisis suele ser un proceso duro, a menudo tedioso, repetitivo y de gran coste económico. El éxito nunca está asegurado y los recursos son siempre limitados. Consecuentemente, también se deben considerar otros enfoques, a veces más efectivos, para obtener información oculta o las claves secretas. Cuando la fuerza de un desarrollo criptográfico excede con mucho el esfuerzo requerido para obtener la misma información de otra forma, el cifrador será lo suficiente fuerte para resistir los ataques.

El criptoanálisis implica una combinación de razonamiento analítico, de aplicación de técnicas de inteligencia artificial (redes neuronales, algoritmos genéticos, sistemas expertos…), utilización de algoritmos discretos, aplicación de métodos estadísticos sofisticados, y la ordenación y reordenación de los datos para revelar características o manifestaciones no aleatorias (por ejemplo, contadores de frecuencia, repeticiones, patrones o fenómenos simétricos), etc.

3.1.2.1 Tipos de ataques

Algunas de las técnicas más importantes de ataque mediante el criptoanálisis son:

- Clasificación según la actitud del atacante

- Ataques pasivos: El atacante no altera la comunicación, sólo la escucha o monitoriza, para obtener información. Son difíciles de detectar ya que no implican alteración de los datos.
- Ataques activos: Suponen alguna modificación del flujo de datos o la creación de flujos falsos.

Ataques de intruso intermedio (Man-In-The-Middle): El atacante se coloca entre las partes legítimas que se comunican. Ese ataque es relevante en protocolos de intercambio de claves y comunicación criptográfica. La idea es que cuando dos partes se intercambian claves para comunicaciones seguras, un adversario se coloca entre ellas en la línea de comunicaciones. El adversario realiza un intercambio de clave por separado con cada parte y cada una de ellas terminará utilizando una clave diferente, ambas conocidas por el atacante. El adversario descifrará las comunicaciones con la clave adecuada y las cifrará con la otra clave para enviarla a la otra parte. Las partes creerán que se están comunicando de forma segura, pero de hecho, el adversario está escuchando y entendiendo todo.

- Clasificación según el tipo de conocimiento:
  - Criptoanálisis sólo con texto cifrado: El atacante posee una cadena de texto cifrado. No conoce nada acerca de los contenidos del mensaje y debe trabajar a partir sólo del texto cifrado. En la práctica, es posible adivinar parte del texto sin cifrar, ya que muchos tipos de mensaje tienen cabeceras de formato fijo.
  - Criptoanálisis con texto en claro conocido: El atacante posee una cadena de texto en claro y la correspondiente cadena de texto cifrado. Con esto el atacante puede plantear ecuaciones que le permitan deducir la clave o al menos reducir el número de pruebas necesarias para un ataque de fuerza bruta.
  - Criptoanáliss con texto en claro elegido: El oponente ha obtenido un acceso temporal a la máquina de cifrado. Puede elegir una cadena de texto en claro y con ella, construir la correspondiente cadena de texto cifrado.
  - Criptoanálisis con texto cifrado elegido: El oponente ha obtenido un acceso temporal a la máquina de descifrado. Puede elegir una cadena de texto cifrado y construir con ella la cadena correspondiente en texto claro.
- Clasificación según el objetivo en criptoanálisis:
  - Ruptura total: el atacante deduce la clave secreta. Por ejemplo con ataques de búsqueda exhaustiva o fuerza bruta para encontrar la clave de un cifrador, útil cuando el tamaño de la clave a atacar es reducido. Se realiza generando de forma aleatoria todos los valores posibles de las claves de acceso y transformándolas.
  - Deducción global: el atacante descubre un algoritmo funcionalmente equivalente para el cifrado y descifrado de mensajes, pero no obtiene la clave.
  - Deducción local: el atacante descubre textos planos o cifrados adicionales a los conocidos previamente. Por ejemplo, ataques de diccionario para romper las contraseñas de los sistemas operativos en los que no se pretende obtener la clave,

sino directamente el texto en claro, ya que el método de cifrado es público y aunque no se disponga de la clave, se puede reproducir.
    - Deducción de información: el atacante descubre alguna información que no era conocida previamente.
    - Distinción del algoritmo: el atacante puede distinguir la información cifrada de una permutación al azar.

3.2 Principios de la seguridad informática

La seguridad de la información tiene como finalidad el proteger la confidencialidad, la integridad y la disponibilidad de la información. Los objetivos principales de seguridad son los siguientes:

- Confidencialidad de datos y de la información del sistema: requisito que intenta que la información privada o secreta no se revele a individuos no autorizados. La protección de la confidencialidad se aplica a los datos almacenados durante su procesamiento, mientras se transmiten y se encuentran en tránsito.
- Integridad: se encarga de garantizar que la información del sistema no haya sido alterada por usuarios no autorizados, evitando la pérdida de consistencia. Presenta dos facetas:
    - Integridad de datos: propiedad de que los datos no hayan sido alterados de forma no autorizada, mientas se almacenan, procesan o transmiten.
    - Integridad del sistema: cualidad que posee un sistema cuando realiza la función deseada, de manera no deteriorada y libre de manipulación no autorizada.
- Autenticación o autentificación: propiedad que permite identificar al generador de la información. Por ejemplo, cuando se reciba un mensaje de alguien, estar seguro de quien es quien lo ha mandado y no una tercera persona haciéndose pasar por él (suplantación de identidad).
- Disponibilidad y accesibilidad de los sistemas y datos, sólo para su uso autorizado: requisito necesario para garantizar que el sistema trabaje puntualmente, con prontitud y que no se deniegue el servicio a ningún usuario autorizado. La disponibilidad protege al sistema contra determinados problemas como los intentos deliberados o accidentales de realizar un borrado no permitido de datos, de causar cualquier tipo de denegación del servicio o de acceso a los datos y de los intentos de utilizar el sistema o los datos para propósitos no autorizados.
- No repudio: servicio de seguridad que permite probar la participación de las partes en una comunicación. Este servicio se encuentra estandarizado en la ISO-7498-2.
    - No Repudio de origen: El emisor no puede negar el envío del mensaje porque el destinatario recibe una prueba infalsificable del origen del envío, lo cual evita que el emisor, de negar tal envío, tenga éxito ante el juicio de terceros. La prueba la crea el propio emisor y la recibe el destinatario.
    - No Repudio de destino: El receptor no puede negar que recibió el mensaje porque el emisor tiene pruebas de la recepción. Este servicio proporciona al emisor la

prueba de que el destinatario legítimo de un envío realmente lo recibió evitando que el receptor lo niegue posteriormente. En este caso la prueba irrefutable la crea el receptor y la recibe el emisor.

El no repudio evita que el emisor o el receptor nieguen la transmisión de un mensaje. Así, cuando se envía un mensaje, el receptor puede comprobar que, efectivamente, el supuesto emisor envió el mensaje. De forma similar, cuando se recibe un mensaje, el emisor puede verificar que, de hecho, el supuesto receptor recibió el mensaje.

3.3 Tipos de Cifrado

- Criptografía: parte de la criptología que se ocupa de las técnicas, bien sea aplicadas al arte o la ciencia, que alteran las representaciones lingüísticas de mensajes, mediante técnicas de cifrado y/o codificado, para hacerlos ininteligibles a intrusos (lectores no autorizados) que intercepten dichos mensajes. Se ocupa del estudio de los algoritmos, protocolos y sistemas que se utilizan para proteger la información y dotar de seguridad a las comunicaciones y a las entidades que se comunican.

El colectivo de herramientas criptográficas básicas va en aumento. Se pueden identificar algoritmos de cifrado, códigos de autenticación, funciones unidireccionales, funciones hash, esquemas de compartición de secretos, esquemas de firmas electrónica, generados de bits pseudo-aleatorios, sistemas de pruebas de conocimiento nulo, etc. A partir de estas herramientas elementales, es posible crear otras herramientas y servicios más complejos como los algoritmos de cifrado basados en umbrales, los protocolos de autenticación, los protocolos de establecimiento de clave y una creciente variedad de protocolos orientados a la aplicación, entre los que destacan los sistemas de pago electrónico, los sistemas de votación electrónica por Internet y los de comercio electrónico.

Un buen sistema de cifrado pone toda la seguridad en la clave y ninguna en el algoritmo. En otras palabras, no debería ser de ninguna ayuda para un atacante conocer el algoritmo que se está usando, sólo si el atacante obtuviera la clave, le serviría conocer el algoritmo.

3.3.1 Algoritmos Simétricos

Los criptosistemas clásicos son de tipo simétrico, también denominados de clave secreta o de secreto compartido. Utilizan una única clave secreta tanto para cifrar como para descifrar. Las dos partes que se comunican han de ponerse de acuerdo de antemano sobre la clave a usar. Una vez ambas tienen acceso a esta clave, el remitente cifra un mensaje usándola, lo envía al destinatario, y éste lo descifra con la misma.

Este tipo de cifrados **garantiza confidencialidad** de los datos.

Los cifradores simétricos pueden dividirse en tres grandes grupos:

- Cifradores monoalfabéticos: letras del texto sin cifrar se transforman en letras de forma única. Un ejemplo sería los cifradores por sustitución.
- Cifradores polialfabéticos: las letras del alfabeto del texto en claro se transforman en letras del espacio del texto cifrado dependiendo de su posición en el texto.
- Cifradores de flujo: genera un flujo de clave largo como el mensaje que se utiliza para cifrar el texto en claro bit a bit o byte a byte.
- Cifradores de bloque: dividen el texto en claro en bloques de tamaño prefijado (por ejemplo 64 bits) y los cifran bloque a bloque.

Ventajas:

- Son sencillos. Los ordenadores los manejan fácil y rápidamente.
- Hay claves simétricas muy sofisticadas y seguras.

Inconvenientes:

- El principal problema con los sistemas de cifrado simétrico no está ligado a su seguridad, sino al intercambio de claves. Una vez que el remitente y el destinatario hayan intercambiado las claves pueden usarlas para comunicarse con seguridad, pero ¿qué canal de comunicación que sea seguro han usado para transmitirse las claves?.
- Otro problema es el número de claves que se necesitan. Si tenemos un número n de personas que necesitan comunicarse entre sí, se necesitan n/2 claves para cada pareja de personas que tengan que comunicarse de modo privado. Esto puede funcionar con un grupo reducido de personas, pero sería imposible llevarlo a cabo con grupos más grandes.

3.3.1.1 AES

AES (Avanced Encryption Standard) también conocido como Rijndael, es un esquema de cifrado por bloques adoptado como un estándar de cifrado por el gobierno de los Estados Unidos. Fue diseñado por dos criptólogos belgas, Jon Daemen y Vincent Rijmen.

AES sustituyó a DES (Data Encryption Standard); es una red de sustitución-permutación. Es rápido tanto en software como en hardware, es relativamente fácil de implementar y requiere poca memoria. Como nuevo estándar de cifrado, se está utilizando actualmente a gran escala.

3.3.1.2 IDEA

IDEA fue creado en 1990 por Xuejia Lai y L.Massey. Se trata de un algoritmo simétrico de cifrado en bloques de 64 bits. Su funcionamiento se basa en operaciones sencillas como multiplicaciones de enteros, sumas y XOR. IDEA trabaja con claves de 128 bits de longitud.

3.3.2 Algoritmos Hash

Los criptosistemas de resumen, más conocidos como funciones o algoritmos hash, constituyen un tipo especial de criptosistemas simétricos.

Un algoritmo tipo hash acepta como entrada un mensaje de longitud arbitraria, y tras efectuar sobre él los cálculos determinados por el algoritmo, devuelve una cadena de caracteres que representa el hash del mensaje al que aplicamos el algoritmo. Este hash no puede ser denominado criptograma dado que **no es posible el proceso de descifrado que nos devolvería el mensaje original**.

Las características de los criptosistemas hash son:

- Unidireccional: Conocido un hash, es computacionalmente imposible la reconstrucción del mensaje original.
- Compresión: A partir de un mensaje de cualquier longitud se obtiene un hash de tamaño fijo, normalmente menor que el del mensaje original.
- Difusión: El resumen es una función compleja de todos los bits del mensaje.
- Colisión simple: se conoce como resistencia débil a las colisiones el hecho de que, dado un mensaje cualquiera, es computacionalmente **imposible** encontrar otro mensaje cuyo hash sea igual.
- Colisión fuerte: se conoce como resistencia fuerte a las colisiones el hecho de que sea computacionalmente **difícil** encontrar dos mensajes cuyo hash sea idéntico.

Estas características hacen de los criptosistemas hash el medio perfecto para la autentificación de todo tipo de información, con usos que van desde la autenticación de ficheros descargados a través de Internet, hasta checksum de paquetes TCP/IP. Es tan sencillo como conocer el hash de la información y una vez obtenido realizar de nuevo la función hash para comprobar las cadenas de salida.

El ataque del cumpleaños sobre funciones criptográficas unidireccionales resumen o hash sucede cuando no se cumple la propiedad de las funciones hash que dice que es imposible encontrar una pareja de mensajes que produzcan la misma cadena de caracteres tras aplicarles la función hash, es decir, produzcan una colisión.

A continuación explicamos algunos de los principales algoritmos criptográficos de tipo hash:

3.3.2.1 MD5

MD5 (abreviatura de Message-Digest Algorithm 5, Algoritmo de Resumen del Mensaje 5) es un algoritmo de reducción criptográfico de 128 bits. Se utiliza extensamente en el mundo del software para proporcionar la seguridad de que un archivo descargado de Internet no se ha alterado. Comparando una suma MD5 publicada con la suma de comprobación del archivo

descargado, un usuario puede tener la confianza suficiente de que el archivo es igual que el publicado por los desarrolladores.

MD5 fue ideado por el matemático Ron Rivest, y supone la evolución de los algoritmos MD2 y MD4. Se trata de una función criptográfica de tipo hash que acepta como entrada un mensaje de cualquier longitud y devuelve como salida una cadena de 128 bits. Su fácil implementación y su gran popularidad le hacen uno de los principales algoritmos hash de la red, usado principalmente en comprobación de ficheros en Internet.

3.3.2.2 SHA

SHA (Secure Hash Algorithm, Algoritmo de Hash Seguro) es un sistema de funciones hash criptográficas relacionadas de la Agencia de Seguridad Nacional de los Estados Unidos y publicadas por el National Institute of Standards and Technology (NIST).

El primer miembro de la familia SHA fue publicado en 1993 y es denominado oficialmente SHA, aunque también se le conoce como SHA-0 para evitar confusiones con sus sucesores. Dos años más tarde el primer sucesor de SHA fue publicado con el nombre de SHA-1. Existen cuatro variantes más que se han publicado desde entonces cuyas diferencias se basan en un diseño algo modificado y rangos de salida incrementados: SHA-224, SHA-256, SHA-384, y SHA-512 (se denomina SHA-2 a toda la familia).

SHA-1 fue ideado por el NIST en 1914 como ampliación del SHA. Se trata de una función criptográfica de tipo hash que acepta una entrada de 2^64 bits como máximo (2048 Terabytes) y devuelve como salida una cadena de 160 bits. SHA-1 es ligeramente más lento que MD5, pero también es computacionalmente más complejo y su salida es de mayor longitud, por lo que se considera de forma global más seguro.

3.3.2.3 HMAC

Probar la integridad de la información transmitida o almacenada es una necesidad fundamental en la computación y las comunicaciones. Los mecanismos que proporcionan esa integridad basadas en una clave secreta se denominan Message Authentication Codes (MAC). Generalmente, estos MAC son utilizados entre dos partes que comparten una clave secreta para validar la información que se transmiten entre ellas. Si a este mecanismo de seguridad le asociamos además un cifrado basado en funciones hash obtenemos HMAC.

HMAC puede ser combinado con cualquier función hash de cifrado, por ejemplo, MD5 o SHA-1; además añade para el cálculo del resultado y la verificación del mensaje una clave secreta; esta clave puede ser de cualquier tamaño aunque el tamaño mínimo recomendado es de 15 y 20 bytes, respectivamente. La clave debe ser elegida aleatoriamente y ser refrescada periódicamente.

3.3.3 Criptografía Asimétrica

La criptografía asimétrica es el método criptográfico que usa un par de claves para el envío de mensajes. Las dos claves pertenecen a la misma persona que ha enviado el mensaje. Una clave es pública y se puede entregar a cualquiera, la otra es privada y el propietario debe guardarla de modo que nadie más tenga acceso a ella. Además, los métodos criptográficos garantizan que esa pareja de claves sólo se puede generar una vez, de modo que se puede asumir que no es posible que dos personas hayan obtenido casualmente la misma pareja de claves.

Si el remitente usa la clave pública del destinatario para cifrar el mensaje, una vez cifrado, sólo la clave privada del destinatario podrá descifrar este mensaje, ya que es el único que la conoce. Por tanto se logra la **confidencialidad** del envío del mensaje, nadie salvo el destinatario puede descifrarlo.

Si el propietario del par de claves usa su clave privada para cifrar el mensaje, cualquiera puede descifrarlo utilizando su clave pública. En este caso se consigue, por tanto, la **identificación y autentificación** del remitente, puesto que sólo pudo haber sido él quien empleó su clave privada. Esta idea es el fundamento de la firma electrónica.

Ventajas:

- Los sistemas de cifrado de clave pública o sistemas de cifrado asimétricos se inventaron con el fin de evitar el problema del intercambio de claves de los sistemas de cifrado simétricos. Con las claves públicas no es necesario que el remitente y el destinatario se pongan de acuerdo en la clave a emplear. Todo lo que se requiere es que, antes de iniciar la comunicación secreta, el remitente consiga una copia de la clave pública del destinatario.
- Esa misma clave pública puede ser usada por cualquiera que desee comunicarse con su propietario, sólo se necesitarán, por tanto, n pares de claves por cada n personas que deseen comunicarse entre sí.

Inconvenientes:

- Para una misma longitud de clave y mensaje se necesita mayor tiempo de proceso que en el cifrado simétrico.
- Las claves deben ser de mayor tamaño que las simétricas.
- El mensaje cifrado ocupa más espacio que el original.

3.3.3.1 RSA

RSA (Rivest, Shamir y Adleman) es un sistema criptográfico de clave pública desarrollado en 1977. Es el primer y más utilizado algoritmo de este tipo y es válido tanto para cifrar como para firmar digitalmente. Se trata de un algoritmo de cifrado asimétrico basado en el problema

de la factorización entera, y aunque la descripción de este algoritmo fue propuesta en 1973 por Clifford Cocks, fue secreta hasta 1978 cuando se publicó RSA.

RSA es el algoritmo asimétrico de cifrado más usado, tanto en conexiones de Internet y protocolos seguros, como en cifrado de datos, por ejemplo en el sistema PGP. Las longitudes de clave usadas hoy en día varían desde los 512 hasta los 4096 bits. Aunque se suelen tomar de forma habitual claves de 1024 puesto que las 512 no se consideran suficientemente seguras. Este tamaño puede parecer pequeño, pero permite la generación de claves de longitudes de hasta 1233 cifras con 4096 bits.

La seguridad de este algoritmo radica en el problema de la factorización de números enteros. Los mensajes enviados se representan mediante números y el funcionamiento se basa en el producto, conocido, de dos números primos grandes elegidos al azar y mantenidos en secreto. No obstante, RSA no es infalible; la enorme complejidad computacional del problema de la factorización entera se debe a una limitación de los computadores actuales. Sin embargo, en el momento en el que se puedan construir computadores cuánticos (que trabajen con lógica ternaria en lugar de lógica binaria) suficientemente potentes, mediante la debida implementación del algoritmo de Shor este trabajo será trivial y permitiría resolver cirptosistemas basados en el problema de factorización entera en un tiempo polinomial.

3.3.4 Criptografía Híbrida

La criptografía híbrida es un método criptográfico que usa tanto un cifrado simétrico como un asimétrico. A pesar de ser computacionalmente mucho más complejos, son el estándar hoy en día. Estos sistemas duales aúnan las ventajas de ambos sistemas, pues un cifrado continuado en clave asimétrica requiere mucho esfuerzo computacional, y un sistema de clave simétrica no es seguro, pues necesita un canal seguro de traspaso de información, y el uso de claves únicas en cifrado simétrico garantiza la seguridad a la vez que se reduce sensiblemente el nivel de recursos necesarios. Ya que compartir una clave simétrica no es seguro, la clave usada es diferente para cada sesión.

La clave de sesión es cifrada con la clave pública, y el mensaje saliente es cifrado con la clave simétrica, todo combinado automáticamente en un sólo paquete. El destinatario usa su clave privada para descifrar la clave de sesión y acto seguido usa la clave de sesión para descifrar el mensaje.

Un sistema de cifrado híbrido no es más fuerte que el de cifrado asimétrico o el de cifrado simétrico de los que hace uso; independientemente de cuál sea más débil, el hecho de que en cada mensaje se cambie la clave hace que si un atacante pudiera descifrar una clave de sesión, sólo sería útil para poder leer un mensaje, el cifrado con esa clave de sesión. El atacante tendría que volver a empezar y descifrar otra clave de sesión para poder leer cualquier otro mensaje.

3.4 Firma Digital

- Una firma digital es un esquema matemático que sirve para demostrar la autenticidad de un mensaje digital, que puede ser por ejemplo un documento electrónico. Una firma digital da al destinatario seguridad de que el mensaje fue creado por el remitente, **autenticidad de origen**, y que no fue alterado durante la transmisión, **integridad**. Las firmas digitales se utilizan comúnmente para la distribución de software, transacciones financieras y en otras áreas donde es importante detectar la falsificación y la manipulación.
- La firma electrónica es una firma digital que se ha almacenado en un soporte hardware; mientras que la firma digital se puede almacenar tanto en soportes hardware como software. La firma electrónica reconocida tiene el mismo valor legal que la firma manuscrita.

La validez de una firma se ampara en el secreto del firmante, es decir, el conocimiento exclusivo de una clave utilizada para generar la firma. Para garantizar la seguridad de las firmas digitales es necesario a su vez que estas sean:

- Únicas: Las firmas deben poder ser generadas solamente por el firmante y por lo tanto infalsificable.
- Infalsificables: Para falsificar una firma digital el atacante tiene que resolver problemas matemáticos de una complejidad muy elevada, es decir, las firmas han de ser computacionalmente seguras.
- Verificables: Las firmas deben ser fácilmente verificables por los receptores de las mismas y, si es necesario, también por los jueces o autoridades competentes.
- Innegables: El firmante no debe ser capaz de negar su propia firma.
- Viables: Las firmas han de ser fáciles de generar por parte del firmante.

La firma digital de un mensaje permite al receptor asegurarse que el contenido del mismo no se ha cambiado accidental o deliberadamente. Una firma digital posibilita al receptor comprobar que un emisor ha originado un mensaje, pero no debe permitir construir el mensaje firmado y el receptor por su parte, debe tener acceso a la información pública del emisor para poder verificar o validar el mensaje. En caso de una disputa, el receptor puede suministrar al juez información no secreta, es decir, el mensaje firmado y la información pública disponible para determinar la autenticidad y origen del mensaje.

Las firmas electrónicas pueden clasificarse en dos categorías:

- Firmas universales, directas o convencionales: Son aquellas que pueden validarse por cualquiera que tenga acceso a parámetros de validación disponibles públicamente. Las firmas universales pueden generarse utilizando Criptografía de clave pública. El RSA es un método o técnica de clave pública eficiente para producir firmas digitales de este tipo,

que no precisan de la existencia de un árbitro seguro, además del emisor y receptor del mensaje.

- Firmas arbitrarias, notariadas o indirectas: Son aquéllas que necesitan de los servicios de un árbitro, notario o tercera parte segura que firme el mensaje para el emisor y lo valide para el receptor. Las firmas arbitrarias se pueden implementar por medio de cifradores simétricos como el AES, IDEA o 3DES. Se precisa el envío de un mensaje autenticado desde el emisor a un árbitro seguro y la transmisión de un mensaje autenticado separado desde el árbitro al receptor. Cada usuario comparte una clave secreta individual con el árbitro, pero el receptor y el emisor no conocen las claves secretas entre sí.

3.5 Certificado Digital

- Un certificado electrónico o digital es un documento digital mediante el cual un tercero confiable (una autoridad de certificación) garantiza la vinculación entre la identidad de un sujeto o entidad (por ejemplo: nombre, dirección y otros aspectos de identificación) y una clave pública.

Cualquier individuo o institución puede generar un certificado digital, pero si éste emisor no es reconocido por quienes interactúen con el propietario del certificado, el valor del mismo es prácticamente nulo. Por ello los emisores deben acreditarse.

- Acreditar: proceso por el cuál entidades reconocidas, generalmente públicas, otorgan validez a la institución certificadora, de forma que su firma pueda ser reconocida como fiable, transmitiendo esa fiabilidad a los certificados emitidos por la citada institución.

El certificado electrónico garantiza:

- La **autenticidad** de las personas y entidades que intervienen en el intercambio de información.
- **Confidencialidad,** que sólo el emisor y el receptor del mensaje puedan acceder a la información.
- La **integridad** de la información intercambiada, asegurando que no se produce ninguna manipulación.
- El **no repudio**, que garantiza al titular del certificado que nadie más que él puede generar una firma vinculada a su certificado y le imposibilita a negar su titularidad en los mensajes que haya firmado.

Un certificado electrónico sirve para:

- Autentificar la identidad del usuario, de forma electrónica, ante terceros.
- Firmar electrónicamente de forma que se garantice la integridad de los datos trasmitidos y su procedencia. Un documento firmado no puede ser manipulado, ya que la firma está asociada matemáticamente tanto al documento como al firmante

- Cifrar datos para que sólo el destinatario del documento pueda acceder a su contenido.

Algunos ejemplos de los servicios al ciudadano que las distintas Administraciones Públicas españolas están ofreciendo son:

- Presentación de recursos y reclamaciones.
- Cumplimentación de los datos del censo de población y viviendas.
- Presentación y liquidación de impuestos.
- Consulta e inscripción en el padrón municipal.
- Consulta de multas de circulación.
- Domiciliación bancaria de tributos municipales (IBI, IVTM, IAE...).
- Consulta y trámites para solicitud de subvenciones.
- Consulta de asignación de colegios electorales.
- Actuaciones comunicadas.
- Firma electrónica de documentos oficiales y expedición de copias compulsadas.

En España, actualmente los certificados electrónicos emitidos por entidades públicas son el DNIe (DNI electrónico) y el de la Fábrica Nacional de Moneda y Timbre (FNMT).

Un certificado emitido por una entidad de certificación autorizada, además de estar firmado digitalmente por ésta, debe contener, por lo menos, lo siguiente:

- Nombre, dirección y domicilio del suscriptor.
- Identificación del suscriptor nombrado en el certificado.
- El nombre, la dirección y el lugar donde realiza actividades la entidad de certificación.
- La clave pública del usuario.
- La metodología para verificar la firma digital del suscriptor impuesta en el mensaje de datos.
- El número de serie del certificado.
- Fecha de emisión y expiración del certificado.

3.5.1 PKI

- Infraestructura de clave pública (PKI): combinación de hardware y software utilizada cuando se despliega la criptografía de clave pública convencional en un entorno donde se precisan comunicaciones electrónicas seguras.
- Autoridad de certificación, certificadora o certificante (CA Certification Authority): entidad de confianza, responsable de emitir y revocar los certificados digitales, utilizados en la firma electrónica, para lo cual se emplea la criptografía de clave pública. Jurídicamente es un caso particular de Prestador de Servicios de Certificación.

Los componentes más habituales de una infraestructura de clave pública son:

- La autoridad de certificación (CA, Certificate Authority): es la encargada de emitir y revocar certificados. Es la entidad de confianza que da legitimidad a la relación de una clave pública con la identidad de un usuario o servicio.
- La autoridad de registro (RA, Registration Authority): es la responsable de verificar el enlace entre los certificados (concretamente, entre la clave pública del certificado) y la identidad de sus titulares.
- Los repositorios: son las estructuras encargadas de almacenar la información relativa a la PKI. Los dos repositorios más importantes son el repositorio de certificados y el repositorio de listas de revocación de certificados. En una lista de revocación de certificados (CRL, Certificate Revocation List) se incluyen todos aquellos certificados que por algún motivo han dejado de ser válidos antes de la fecha establecida dentro del mismo certificado.
- La autoridad de validación (VA, Validation Authority): es la encargada de comprobar la validez de los certificados digitales.
- La autoridad de sellado de tiempo (TSA, TimeStamp Authority): es la encargada de firmar documentos con la finalidad de probar que existían antes de un determinado instante de tiempo.
- Los usuarios y entidades finales son aquellos que poseen un par de claves (pública y privada) y un certificado asociado a su clave pública. Utilizan un conjunto de aplicaciones que hacen uso de la tecnología PKI (para validar firmas digitales, cifrar documentos para otros usuarios, etc.)

3.5.2 Estándar x.509

- En criptografía, x.509 es un estándar UIT-T para infraestructuras de claves públicas. x.509 especifica, entre otras cosas, formatos estándar para certificados de claves públicas y un algoritmo de validación de la ruta de certificación

La estructura de un certificado digital x.509 v3 es la siguiente:

- Certificado
  - Versión.
  - Número de serie.
  - ID del algoritmo.
  - Emisor.
  - Validez.
    - No antes de.
    - No después de.
  - Sujeto.
  - Información de clave pública del sujeto.
    - Algoritmo de clave pública.
    - Clave pública del sujeto.

- Identificador único de emisor (opcional).
- Identificador único de sujeto (opcional).
- Extensiones (opcional).
  - ...
- Algoritmo usado para firmar el certificado.
- Firma digital del certificado.

El DNI electrónico posee dos certificados x.509, estos proporcionan autenticación y firma.

4 DOCUMENTO DE ANÁLISIS

4.1 Introducción

4.1.1 Propósito

El sistema permitirá hacer copias de seguridad de las notas existentes en la base de datos del Sistema de gestión de cursos de código abierto Moodle mejorando la seguridad existente.

El sistema Moodle no es seguro, como ya se indicó en el capítulo 2 de este proyecto; nuestro sistema permitirá detectar y avisar de cada cambio relativo a las calificaciones a los profesores, gestores, o personal encargado pertinente. En caso de que el intruso no tuviera los permisos necesarios para realizar el cambio, se informaría de tal incidencia al Administrador del sistema. Añadido a esto, los encargados podrán seleccionar la nota real entre las dos que han causado el conflicto permitiendo, la actualización automática del sistema.

4.1.2 Ámbito

Como hemos indicado, el objetivo principal de nuestro sistema es la mejora de la seguridad asociada a las calificaciones almacenadas en Moodle. Para ello dispondremos de un servidor que realice peticiones a la base de datos de Moodle y recupere la información necesaria.

Cuando las notas recibidas no se encuentren en el sistema, se almacenarán en nuestro servidor de dos maneras, una explícita, para acelerar las búsquedas, y otra cifrada, para aumentar la seguridad. La información será solicitada cada 15 minutos de manera automática, minimizando el tiempo posible de actuación, acceso y modificación por parte del intruso malicioso.

Cuando las calificaciones se encuentren ya almacenadas en el sistema, se procederá a su comparación comprobando, de esta manera, si ha habido un cambio en las mismas. En caso de detectarse una inconsistencia en las notas, el sistema enviará un aviso al profesor, gestor, o personal encargado del curso que modificó esa nota la última vez, de acuerdo con los registros de Moodle, informándole de lo sucedido y pidiendo la confirmación del cambio. Forma parte de las atribuciones de estos encargados el comprobar este hecho; utilizando la interfaz gráfica del aula virtual y modificar la nota pertinente en ese sistema en el caso de que hubiera sido alterada sin su consentimiento.

Una vez se detecte por primera vez una inconsistencia en las notas, el sistema dejará de comprobar esa nota hasta que el usuario responsable avisado resuelva el conflicto, indicando al sistema cuál es el valor real de la nota.

Si a una hora determinada del día alguna de las notificaciones no hubiera sido atendida, el sistema lanzará un nuevo aviso al personal encargado. En él se volverá a solicitar que se indique que se ha comprobado la modificación de las calificaciones en Moodle y cuál es la nota. Una vez seleccionada la calificación, el sistema por si sólo se encargará de actualizar la nota existente, almacenar la incidencia en la tabla reservada para ello y almacenar la nota real cifrada.

Cada vez que se detecte un cambio en las notas, se procederá a guardar un registro de las incidencias en el sistema a modo de histórico, de tal forma que se pueda utilizar más adelante para obtener estadísticas, por ejemplo, de las veces que se han detectado esos errores, la frecuencia y los cursos más afectados por ellas.

Dado que nuestro sistema necesitará conocer que usuarios son los encargados de las asignaturas o cuáles tienen permiso para modificar una nota, será preciso almacenar datos personales de estos trabajadores. Así podremos garantizar que el mensaje de aviso, en caso de inconsistencia, llegará al personal pertinente, evitando molestar innecesariamente a personas no relacionadas con la modificación de esa calificación y pudiendo detectar cualquier acceso de usuarios que no tengan los permisos requeridos para modificar esa nota.

Para que el usuario pueda formar parte del sistema, debe existir en la Base de Datos de Moodle como usuario registrado y estar asociado al menos a un curso con los permisos de “Profesor” o “Profesor con Permiso de Edición” o a todo el sistema con permisos de “Gestor”. A los primeros se les permitirá modificar exclusivamente las notas relacionadas con los cursos a los que estén asociados, los segundos podrán modificar cualquier nota, independientemente del curso.

Si el usuario que modificó la nota no tuviera permisos de “Profesor”, “Profesor sin Permiso de Edición” o “Gestor”, automáticamente se le consideraría un intruso, por lo que se procedería a avisar al Administrador del sistema sobre la intromisión. Forma parte de las atribuciones del Administrador comprobar la incidencia, mantener la consistencia en ambas bases de datos, y en caso de que sea pertinente, avisar a los profesores responsables.

En ningún momento se modificarán valores de manera directa en la base da datos de Moodle para evitar inconsistencias entre la multitud de tablas relacionas que existen en ella; todo se realizará utilizando el interfaz gráfico de este sistema.

Si la aplicación fuera lo suficientemente útil, la documentación en inglés se publicaría en la página oficial de Moodle; contemplando esa posibilidad se utilizará PHP para el desarrollo del Trabajo Fin de Grado, lenguaje usado en la implementación de este CMS.

4.1.3 Definiciones, Acrónimos y Abreviaturas

El objetivo de este apartado es evitar cualquier confusión o ambigüedad al lector a lo largo de la lectura de este documento, proporcionando diversas acepciones relevantes, así como información necesaria para comprender el funcionamiento semántico del sistema:

- *Course Management System (CMS):* programa que permite crear una estructura de soporte (framework) para la creación y administración de contenidos, principalmente en páginas Web, por parte de los administradores, editores, participantes y demás roles.
  Consiste en una interfaz que controla una o varias bases de datos donde se aloja el contenido del sitio web. El sistema permite manejar de manera independiente el contenido y el diseño. De esta forma es posible manejar el contenido y darle en cualquier momento un diseño distinto al sitio web, sin tener que darle formato al contenido de nuevo, además de permitir la fácil y controlada publicación en el sitio a varios editores.
  - Sinónimos: Ninguno.
  - Homónimos: Ninguno.

- *Learning Management System (LMS):* sistema de software diseñado para facilitar a profesores la gestión de cursos virtuales para sus estudiantes, especialmente ayudándolos en la administración y desarrollo del curso. El sistema puede seguir a menudo el progreso de los principiantes; puede ser controlado por los profesores y los mismos estudiantes.
  Los componentes de estos sistemas incluyen generalmente las plantillas para elaboración de contenido, foros, charla, cuestionarios y ejercicios tipo múltiple-opción, verdadero/falso y respuestas de una palabra. Los profesores completan estas plantillas y después las publican para ser utilizados por los estudiantes.
  - Sinónimos: Virtual Learning Environment (VLE).
  - Homónimos: Ninguno.
- *Open Source*: término con el que se conoce al software distribuido y desarrollado libremente. El código abierto tiene un punto de vista más orientado a los beneficios prácticos de compartir el código que a las cuestiones éticas y morales, las cuales destacan en el llamado software libre.
  - Sinónimos: Ninguno.
  - Homónimos: Ninguno.

- *Free Software*: denominación del software que respeta la libertad de los usuarios sobre su producto adquirido y, por tanto, una vez obtenido puede ser usado, copiado, estudiado, modificado, y redistribuido libremente
  - Sinónimos: Ninguno.
  - Homónimos: Ninguno.

- *Moodle*: Sistema de Gestión de Cursos de Código Abierto (Open Source Course Management System, CMS), conocido también como Sistema de Gestión del Aprendizaje (Learning Management System, LMS) o como Entorno de Aprendizaje Virtual (Virtual Learning Environment, VLE).
  - Sinónimos: Ninguno.
  - Homónimos: Ninguno.

- *Cifrado*: procedimiento que, utilizando un algoritmo cifrado con cierta clave de cifrado, transforma un mensaje, sin atender a su estructura lingüística o significado, de tal forma que sea incomprensible o, al menos, difícil de comprender, para toda persona que no tenga la clave secreta de descifrado usada para poder interpretarlo.
  - Sinónimos: Ninguno.
  - Homónimos: Ninguno.

- *Certificado Digital*: documento digital mediante el cual un tercero confiable, una autoridad de certificación, garantiza la vinculación entre la identidad de un sujeto o entidad, por ejemplo: nombre, dirección y otros aspectos de identificación, y una clave pública. Este tipo de certificados se emplea para comprobar que una clave pública pertenece a un individuo o entidad.
  - Sinónimos: Certificado de Clave Pública, Certificado de Identidad.
  - Homónimos: Ninguno.
- *Cron:* administrador regular de procesos en segundo plano, demonio, que ejecuta procesos o guiones a intervalos regulares, por ejemplo, cada minuto, día, semana o mes. Los procesos que deben ejecutarse y la hora en la que deben hacerlo se especifican en el fichero "crontab".
  - Sinónimos: Ninguno.
  - Homónimos: Ninguno.

4.2 Requisitos funcionales

4.2.1 Requisitos de Gestión de Notas:

4.2.1.1 El sistema podrá conectarse de manera remota con la base de datos del sistema Moodle.
4.2.1.2 El sistema recogerá la información asociada necesaria sobre las notas almacenadas en Moodle para su tratamiento.
4.2.1.3 El sistema almacenará las notas nuevas de manera explícita.
4.2.1.4 El sistema almacenará las notas nuevas cifradas.
4.2.1.5 El sistema será capaz de procesar las notas y saber si han sido modificadas desde la última vez que se accedió a ellas o si permanecen intactas.
4.2.1.6 El sistema guardará un registro de incidencias que contendrá ambas notas, las fechas en las que se modificaron y alguna información identificativa extra.
4.2.1.7 El sistema será capaz de actualizar de manera automática las notas almacenadas en el sistema para solucionar las incidencias.
4.2.1.8 El sistema será capaz de actualizar las notas cifradas cuando se produzca un cambio autorizado en las notas almacenadas.

4.2.2 Requisitos de Gestión de Avisos:

4.2.2.1 El sistema será capaz de localizar al “Profesor”, “Profesor sin Permisos de Edición” o “Gestor” responsable del curso que modificó por última vez la nota.
4.2.2.2 El sistema será capaz de enviar un aviso al personal encargado al detectar una inconsistencia.
4.2.2.3 El sistema avisará de la modificación de la nota en cuanto ésta se detecte y no volverá a comprobar la nota hasta que el usuario responsable indique cuál de las de las dos calificaciones almacenadas es la válida.
4.2.2.4 El sistema mandará un aviso cada 24 horas a todos los usuarios responsables que no hayan solucionado el conflicto entre las notas para evitar que la calificación permanezca inconsistente de manera indefinida.
4.2.2.5 El sistema informará al usuario responsable de la actividad, curso y alumno afectados por el cambio de la nota.
4.2.2.6 El sistema comunicará al usuario responsable qué nota ha sido modificada, cuál es el valor anterior y posterior al cambio, acompañados de la fecha y hora de cada uno para facilitar la identificación de la inconsistencia.
4.2.2.7 Si el usuario que modifica la nota no tuviera los permisos necesarios, el sistema mandará un aviso al Administrador avisando de la situación.
4.2.2.8 El sistema facilitará al usuario responsable la posibilidad de arreglar la inconsistencia del sistema de manera automática.
4.2.2.9 Si el acceso a cualquiera de los dos sistemas dejara de ser posible, se

enviaría un aviso al Administrador.

4.2.3 Requisitos de Gestión de Usuarios:

4.2.3.1 El sistema podrá recoger la información necesaria sobre los usuarios con permisos de "Profesor", "Profesor sin Permisos de Edición" o "Gestor" del sistema Moodle y que estén encargado de uno o más cursos en el caso de los dos primeros, y de todo el sistema en el caso de los últimos.

4.2.3.2 El sistema almacenará la información relativa a este tipo de usuarios asociándolos a los cursos que puedan modificar.

4.2.3.3 El sistema será capaz de detectar a los intrusos que cambien las notas sin tener los permisos suficientes ya sean "Profesores" no asociados a ese curso, "Alumnos" o "Invitados".

4.2.3.4 Si la modificación de una nota estuviera asociada a un usuario no guardado en el sistema, éste se podría en contacto con Moodle para solicitar la información disponible de todos los usuarios con los roles "Profesor", "Profesor sin Permisos de Edición" o "Gestor", actualizando la información almacenada.

4.2.3.5 Si tras actualizar la información sobre los usuarios encargados la modificación de las notas siguiera asociada a un usuario sin permisos, el usuario asociado sería considerado un intruso y se solicitaría su información personal al sistema Moodle.

4.2.4 Requisitos de Gestión de la Aplicación:

4.2.4.1 El sistema se lanzará de manera automática, comprobando cada 15 minutos que las notas existentes en el sistema no han sido modificadas.

4.2.4.2 El sistema lanzará una consulta cada 24 horas que genere avisos para los casos de inconsistencia no resueltos.

4.3 Requisitos no funcionales:

4.3.1 La aplicación se ejecutará sobre un servidor con sistema operativo tipo Unix, sin interfaz gráfica, independiente del servidor sobre el que se ejecute Moodle, para evitar la sobrecarga de servidor principal y agilizar las operaciones.

4.3.2 Se utilizará el comando cron de Unix, para el lanzamiento automático de la aplicación cada 15 minutos.

4.3.3 Se elegirá una hora en la que la actividad del sistema sea menor para realizar el aviso masivo cada 24 horas utilizando el fichero "crontab" pata el lanzamiento.

4.3.4 Las notas de un curso seguirán solicitándose hasta 30 días después de su fecha de finalización; una vez que se alcance esa fecha las notas correspondientes a ese curso dejarán de comprobarse.

4.3.5 La aplicación será compatible con el Sistema de Gestión de Cursos de Código Abierto Moodle.

4.3.6 Los datos del usuario serán tratados confidencialmente y de acuerdo con lo establecido en la Ley Orgánica 15/1999 del 13 de Noviembre de Protección de Datos de Carácter Personal, (LOPD).

4.3.7 El sistema deberá almacenar todos los datos en una base de datos SQL estándar, que sea gestionada de manera óptima mediante un Sistema Gestor de Bases de Datos MySQL.

4.3.8 El formato de las fechas será TimeStamp en la base de datos y "Día de la semana, Mes, Día del mes, Hora y Año" para los avisos.

4.3.9 Se realizará periódicamente una copia de seguridad de los datos del sistema para su posterior recuperación en caso de error.

4.3.10 Las contraseñas de los servidores serán suficientemente robustas como para proteger la información de los intrusos maliciosos. Más de ocho caracteres alfanuméricos, con al menos un número, un carácter especial, una mayúscula y una minúscula, sin que formen una palabra reconocible.

4.3.11 Se aplicará cifrado HMAC para mayor seguridad en las notas.

4.3.12 Se utilizará un mecanismo de cifrado de las notas para garantizar la seguridad extra de las notas.

4.3.13 Se utilizara el lenguaje de programación PHP para la elaboración del sistema.

4.3.14 Se desarrollará un manual técnico de referencia para el buen uso de la aplicación, haciendo especial hincapié en los requisitos necesarios para la optimización del uso del sistema.

4.3.15 Se facilitará la ampliación y modificación de la funcionalidad del sistema a través de la difusión de su documentación en la página oficial de Moodle en inglés.

4.4 Requisitos de información:

4.4.1 Información requerida para las Notas:

4.4.1.1 Identificador de la nota almacenada, único en el sistema.
4.4.1.2 Nota final.
4.4.1.3 Fecha y Hora en la que se modificó.
4.4.1.4 Identificador del Usuario al que pertenece (alumno).
4.4.1.5 Identificador del Usuario que la modificó por última vez.
4.4.1.6 Identificador de la Actividad a la que está asociada.

4.4.2 Información requerida para las Actividades:

4.4.2.1 Identificador de Actividad.
4.4.2.2 Nombre de la Actividad.
4.4.2.3 Tipo de Actividad (módulo, bloque, manual, curso, categoría…).
4.4.2.4 Identificador del Curso al que pertenece.

4.4.3 Información requerida para los Usuarios:

4.4.3.1 Identificador de Usuario.
4.4.3.2 Nombre.
4.4.3.3 Apellidos.
4.4.3.4 Email.

4.4.4 Información requerida para los Cursos:

4.4.4.1 Identificador del Curso.
4.4.4.2 Nombre del Curso.

4.5 Descripción de los actores

El objetivo de este apartado es evitar cualquier confusión o ambigüedad con respecto a los actores que pueden utilizar el sistema, para ello se dará una breve descripción de los mismos y sus funciones:

- *Moodle:* sistema que facilita los datos a la aplicación.
- *Tiempo:* capaz de lanzar la aplicación tras un tiempo determinado.
- *UsuarioPrivilegiado:* Hay 4 tipos de usuarios privilegiados:
  - *Administrador:* No puede cambiar calificaciones. Se limita a atender dos tipos de avisos: cuando se detecta un acceso ilícito al sistema, es decir, la modificación de la nota fue realizada por un usuario sin los permisos necesarios para hacerlo, y cuando alguna de las bases de datos del sistema ha dejado de funcionar.
  - *Gestor:* Puede cambiar la nota de cualquier curso existente. Tiene permisos sobre todo el sistema. Se le enviará un aviso cuando sea responsable de la modificación de la nota.
  - *Profesor:* Puede cambiar la nota de cualquiera de los cursos que tiene asignado. Se le enviará un aviso cuando sea responsable de la modificación de la nota en uno de sus cursos.
  - *Profesor sin Permiso de Edición:* Igual al anterior.

4.6 Diagrama de casos de uso

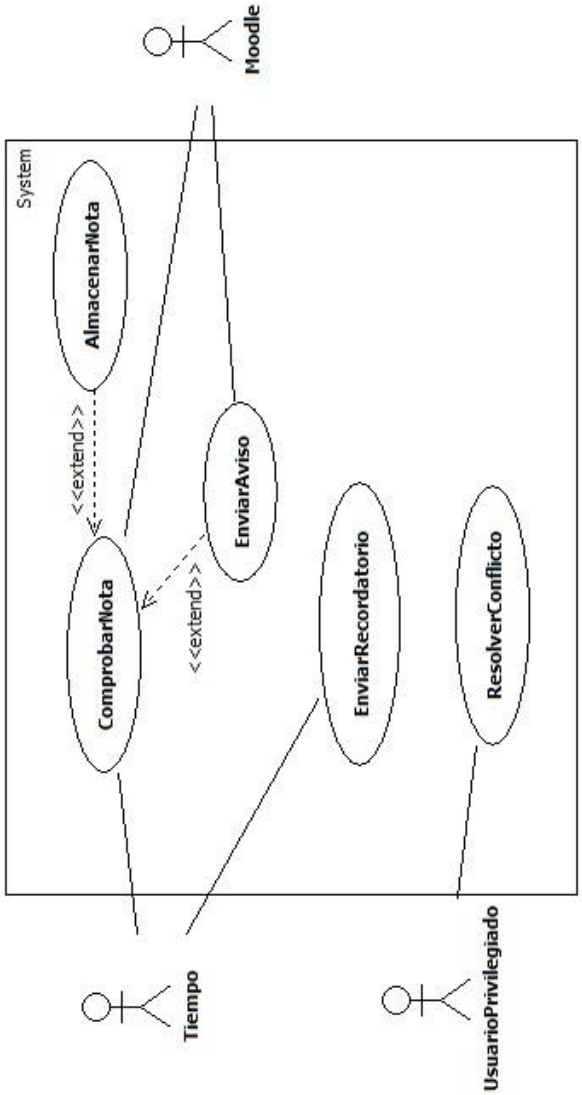


Diagrama de Casos de Uso de Análisis

4.7 Especificación de casos de uso

4.7.1 Caso de Uso: ComprobarNota.

Caso de Uso: ComprobarNota.
ID: 1
Breve Descripción: Tras un determinado periodo de tiempo el sistema solicita todas las calificaciones existentes en Moodle para comprobar que no han sido modificadas.
Actores Principales: Tiempo.
Actores Secundarios: Moodle.
Precondiciones: El sistema es capaz de conectarse a Moodle para conseguir la información necesaria sobre las notas.
Flujo Principal: 1 El caso de uso se inicia cuando, pasado un determinado periodo de tiempo, el sistema solicita las notas disponibles en Moodle. 2 Moodle devuelve las notas solicitadas. 3 Mientras haya notas por comparar el sistema comprueba si la nota ya está almacenada. 3.1 Si la nota existe: Compara el valor almacenado con el obtenido. 3.1.1 Si coinciden: Pasa a la nota siguiente. 3.1.2 Si no coinciden: Extend(EnviarAviso). 3.2 Si no existe: Extend(AlmacenarNota). 4 El caso de uso finaliza cuando se termina de comparar la última de las notas obtenidas.
Postcondiciones: Todas las notas han sido comprobadas.
Flujos Alternativos: Ninguno.

4.7.2 Caso de Uso: AlmacenarNota.

Caso de Uso: AlmacenarNota.
ID: 2
Breve Descripción: Una nota nueva se almacena en el sistema.
Actores Principales: Tiempo.
Actores Secundarios: Ninguno.
Precondiciones: La nota no existía con anterioridad en el sistema.
Flujo Principal: 1 El caso de uso se inicia cuando el sistema detecta que la nota que desea comparar no ha sido almacenada con anterioridad. 2 La información de la nota se almacena en el sistema con las medidas de seguridad necesarias. 3 El caso de uso finaliza cuando la nota ha sido correctamente almacenada.
Postcondiciones: La nota queda almacenada en el sistema con las medidas de seguridad necesarias.
Flujos Alternativos: Ninguno.

4.7.3 Caso de Uso: EnviarAviso.

Caso de Uso: EnviarAviso.
ID: 3
Breve Descripción: El sistema envía un aviso para informar de la inconsistencia detectada en una calificación.
Actores Principales: Tiempo.
Actores Secundarios: Moodle.
Precondiciones: La nota existía en el sistema y tenía un valor distinto al obtenido en Moodle.
Flujo Principal: 1 El caso de uso se inicia cuando el sistema detecta que la nota comparada existía previamente en el sistema y tenía un valor asociado distinto. 2 El sistema busca la información del usuario que modificó la calificación y genera un aviso. 3 El caso de uso finaliza cuando el aviso ha sido enviado.
Postcondiciones: Se ha generado y enviado un aviso a la persona que modificó la calificación.
Flujos Alternativos: SinPermisosNecesarios.

4.7.3.1 Flujo Alternativo: DatosDeUsuarioNoEncontrados

Flujo Alternativo: EnviarAviso:SinPermisosNecesarios.
ID: 3.1
Breve Descripción: El Sistema no tiene almacenados los datos del Usuario que modificó la calificación.
Flujo Alternativo: 1 El flujo alternativo comienza cuando el Sistema no encuentra los datos del usuario que modificó la calificación. 2 El Sistema solicita a Moodle la información de todos los usuarios que tienen permisos para modificar calificaciones. 3 Moodle devuelve la información solicitada. 4 Mientras haya usuarios que comparar, el sistema comprueba si el usuario ya está almacenado en el sistema: 4.1 Si existe: Pasa al usuario siguiente. 4.2 Si no existe: almacena la información. 5 El flujo alternativo finaliza cuando se termina de comparar el último de los usuarios obtenidos.

4.7.3.2 Flujo Alternativo: EnviarAviso:SinPermisosNecesarios.

Flujo Alternativo: EnviarAviso:SinPermisosNecesarios.
ID: 3.2
Breve Descripción: El Sistema detecta que el usuario que modificó la calificación no tenía los permisos necesarios para realizar tal acción.
Flujo Alternativo: 1 El flujo alternativo comienza cuando el Sistema detecta que el usuario que modificó la calificación no tenía los permisos para realizar tal acción. 2 El Sistema solicita a Moodle la información sobre el usuario que modificó la calificación. 3 Moodle devuelve la información solicitada. 4 El Sistema genera un aviso sobre el problema detectado y se lo envía al Usuario privilegiado encargado, en este caso el Administrador del sistema. 5 El flujo alternativo finaliza cuando el aviso ha sido enviado.

4.7.4 Caso de Uso: EnviarRecordatorio.

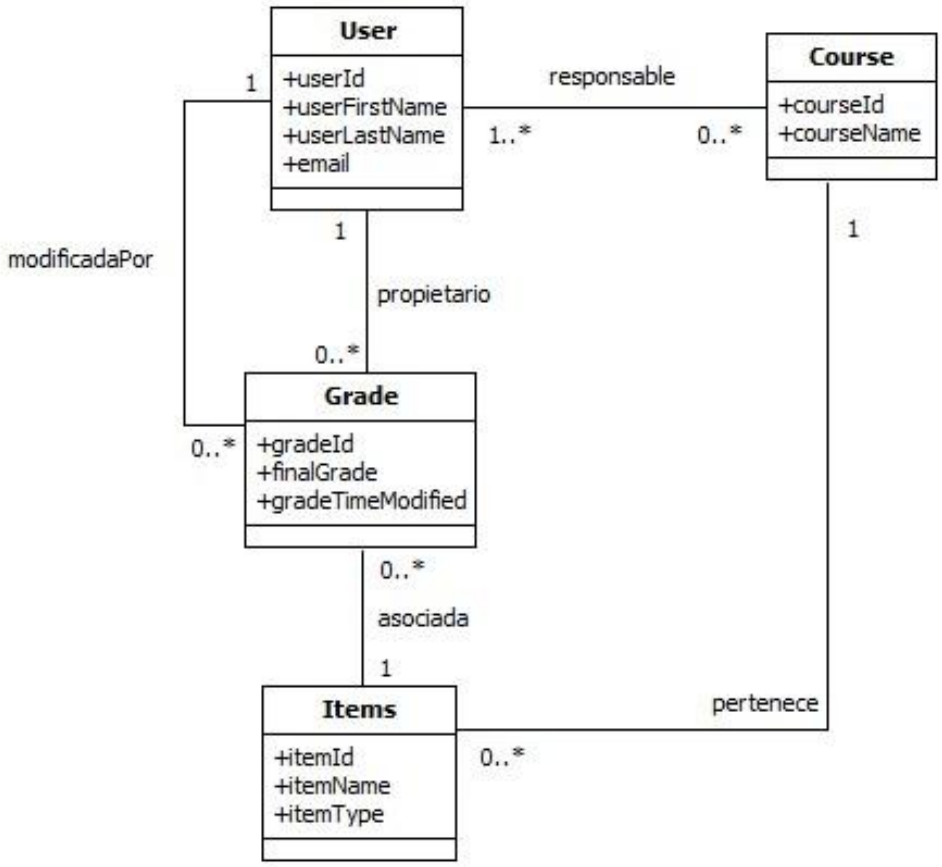
Caso de Uso: EnviarRecordatorio.
ID: 4
Breve Descripción: El sistema enviará un aviso recordatorio indicando que se ha detectado un conflicto en las calificaciones que aún no ha sido resuelto.
Actores Principales: Tiempo.
Actores Secundarios: Ninguno.
Precondiciones: Al menos una nota generó un aviso anterior por un conflicto entre el valor almacenado y el obtenido y éste no fue atendido a una hora determinada del día.
Flujo Principal: 1 El caso de uso se inicia cuando el sistema detecta que un aviso generado al detectarse un conflicto en las notas no ha sido atendido a una hora determinada del día. 2 El sistema genera un aviso recordatorio y lo envía al Usuario Privilegiado encargado de resolver la incidencia. 3 El caso de uso finaliza cuando el aviso recordatorio ha sido enviado.
Postcondiciones: Se ha enviado un recordatorio a la persona que modificó la calificación o al Administrador del sistema en caso de ser una modificación ilícita.
Flujos Alternativos: Ninguno.

4.7.5 Caso de Uso: ResolverConflicto.

Caso de Uso: ResolverConflicto.
ID: 5
Breve Descripción: El Usuario resuelve un conflicto en las notas.
Actores Principales: Usuario Privilegiado.
Actores Secundarios: Ninguno.
Precondiciones: El Usuario ha recibido un aviso por una nota con un conflicto entre el valor almacenado y el obtenido.
Flujo Principal: 1 El caso de uso se inicia cuando el Usuario Privilegiado, tras recibir un aviso sobre una posible inconsistencia en sus notas, desea resolver el conflicto detectado. 2 El Usuario Privilegiado indica cual es la calificación correcta entre las dos disponibles e informa al sistema. 3 El sistema recibe la información correcta, la almacena e informa al usuario. 4 El caso de uso finaliza cuando el Usuario Privilegiado ha sido correctamente informado.
Postcondiciones: Se ha resuelto el conflicto en las notas.
Flujos Alternativos: Ninguno.

4.8 Realización de casos de uso:

4.8.1 Modelo de Dominio



Modelo de Dominio

4.8.2 Diagramas de Secuencia de Sistema

4.8.2.1 Caso de Uso 1: ComprobarNota

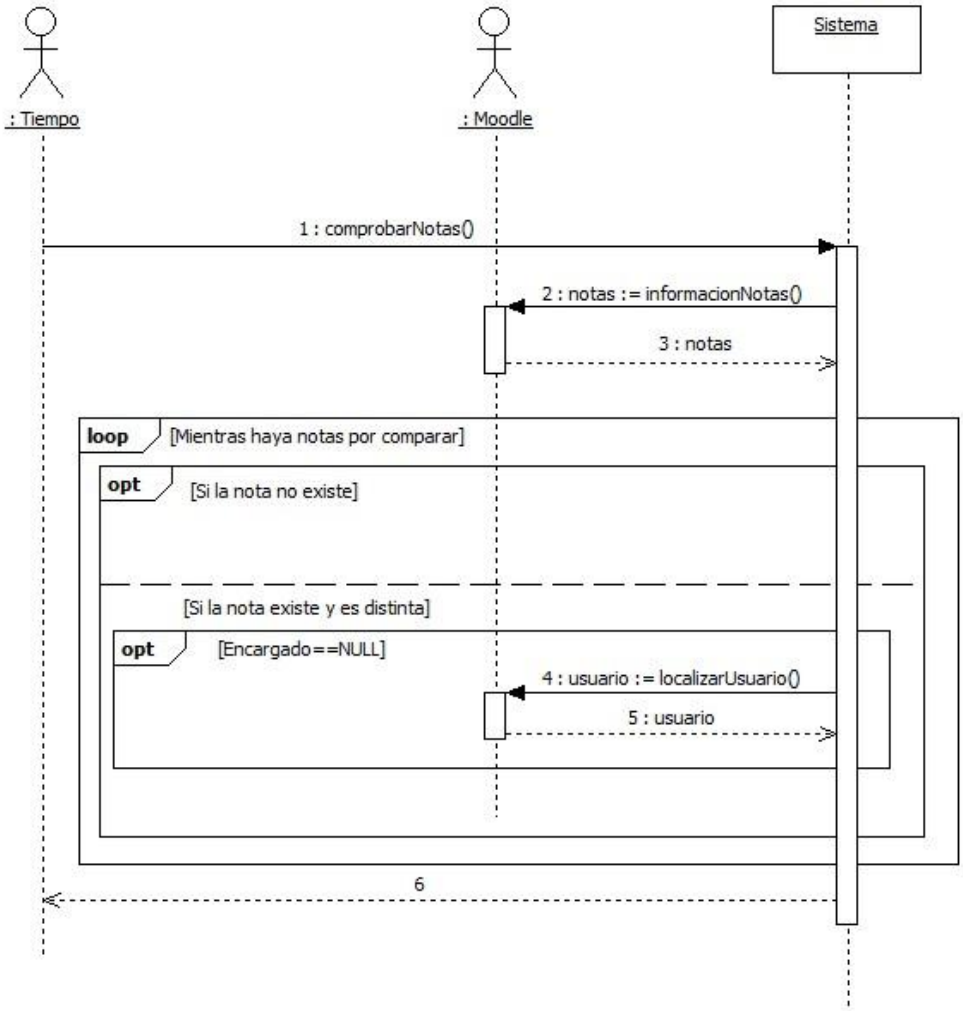


Diagrama de Secuencia 1

4.8.2.2 Caso de Uso 2: EnviarRecordatorio

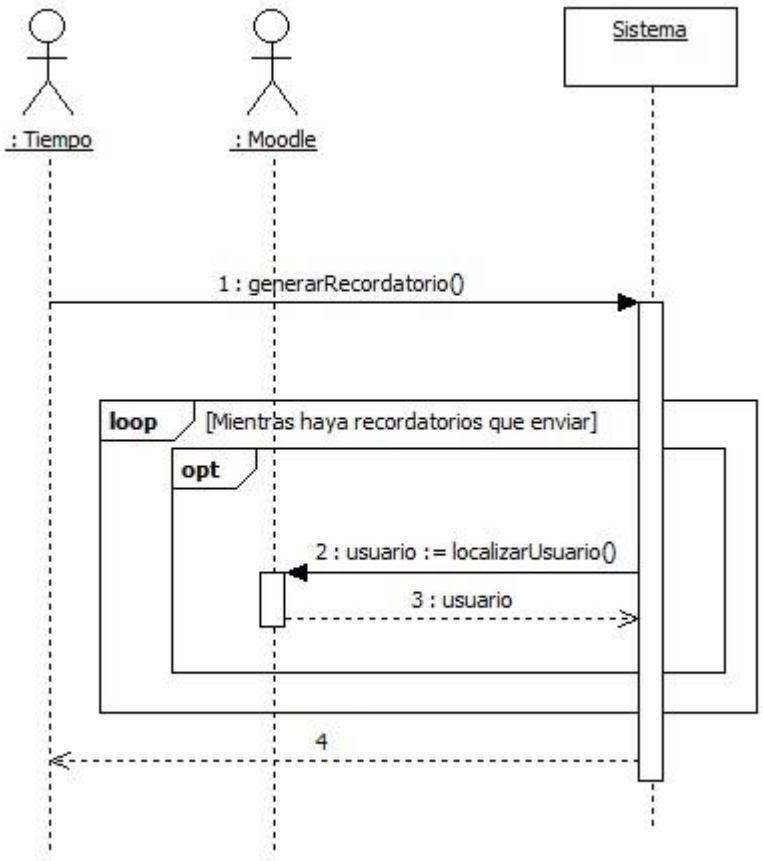


Diagrama de Secuencia 2

4.8.2.3 Caso de Uso 3: ResolverConflicto

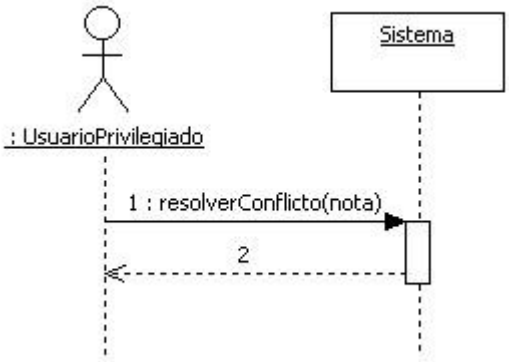


Diagrama de Secuencia 3

5 DOCUMENTO DE DISEÑO

5.1 Introducción.

En este documento descubriremos cada una de las fases de diseño iniciales de una aplicación software enunciadas por Sommerville en su libro "Ingeniería del Software"; es decir, el diseño de la arquitectura del sistema, una primera especificación abstracta del mismo, el diseño de todas las interfaces que se usarán y la subdivisión en componentes. Cada una de estas partes será documentada y todas las decisiones tomadas serán comentadas así como el espacio de diseño en el que nos encontramos a cada momento.

5.1.1 Propósito del Sistema

El propósito de este documento es la elaboración del diseño de la aplicación "Gestión de Calificaciones" en el Sistema de Gestión de Cursos de Código Abierto Moodle, a partir de los requisitos recogidos en el documento de análisis de este mismo proyecto.

5.1.2 Definiciones, Acrónimos y abreviaturas

El objetivo de este apartado es evitar cualquier confusión o ambigüedad al lector a lo largo de la lectura de este documento proporcionando diversas acepciones relevantes:

- *WWW:* La World Wide Web es un sistema de distribución de información basado en hipertexto o hipermedios enlazados y accesibles a través de Internet.
- *HTTP:* Es el protocolo de transferencia de hipertexto. Es uno de los protocolos más usados en cada transacción de la World Wide Web.
- *HTML:* HyperText Markup Language (Lenguaje de Marcado de Hipertexto) es el lenguaje de marcado predominante para la elaboración de páginas web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes.
- *Servidor:* Computadora que, formando parte de una red, provee servicios a otras computadoras denominadas clientes. Además, puede ser una computadora en la que se ejecute un programa que realice alguna tarea en beneficio de otras aplicaciones llamadas clientes, tanto si se trata de un ordenador central (mainframe), un miniordenador, una computadora personal, una PDA o un sistema embebido.
- *SQL:* Structured Query Language es un lenguaje declarativo de acceso

a bases de datos relacionales que permite especificar diversos tipos de operaciones en éstas.

- *PHP:* Lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas. Se usa principalmente para la interpretación del lado del servidor, aunque puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica usando las bibliotecas Qt o GTK+.

5.2 Diseño de la arquitectura del sistema.

La propuesta arquitectónica que hemos considerado en esta fase queda reflejada en el siguiente diagrama de despliegue:

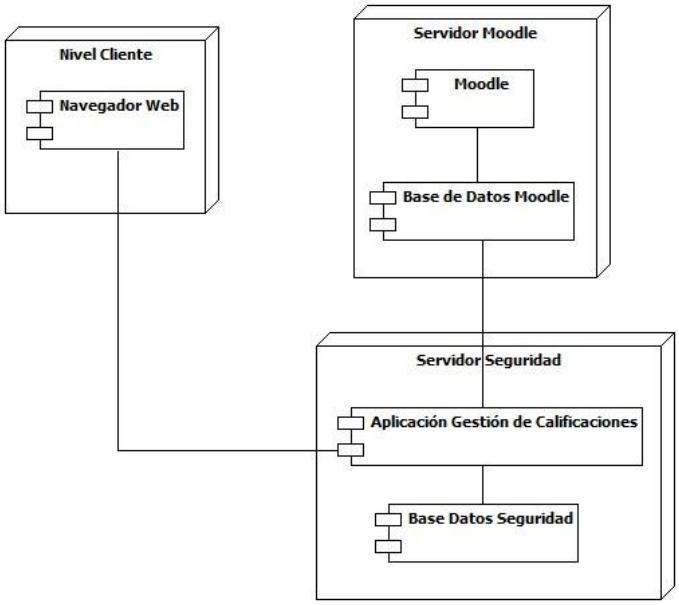


Diagrama de despliegue del sistema

Como se puede apreciar en el diagrama de despliegue, nuestra arquitectura está constituida por una máquina cliente y dos servidores. Sobre uno de ellos se ejecutará la herramienta Moodle y el otro albergará nuestro sistema. Solamente se adoptará la arquitectura Cliente-Servidor para resolver los conflictos detectados en el valor de las calificaciones.

Dado que la funcionalidad principal de nuestro sistema será independiente de los usuarios humanos, no siendo necesario un control continuo de las operaciones; el "Nivel Cliente" estará limitado a un navegador Web, utilizado únicamente para resolver la inconsistencia en la base de datos de seguridad.

El "Servidor Moodle" contendrá la herramienta Moodle junto con su base de datos asociada; nuestro sistema se nutrirá de la información almacenada en ella. El "Servidor Seguridad" contendrá nuestra aplicación, acompañada de una base de datos en la que se almacenarán las copias cifradas de seguridad de las calificaciones.

Necesitamos que nuestro sistema sea distribuido, entendiendo como sistema distribuido aquel que se basa en la comunicación de diferentes nodos, los cuales realizan procesamiento de la información paralelamente. Esta necesidad radica en que el sistema será utilizado por distintos usuarios, para resolver los conflictos encontrados en las notas y ninguno de ellos tendrá los permisos necesarios para acceder directamente al servidor sobre el que corre la aplicación y, de esta forma, actualizar la base de datos de seguridad manualmente. La distribución es, por tanto, necesaria, puesto que se debe permitir que cada usuario desarrolle su actividad, cuando sea requerida, desde un ordenador personal.

El uso de dos servidores distintos, uno exclusivo para la aplicación y otro para Moodle, se debe a los requisitos iniciales del Trabajo de Fin de Grado. La aplicación debe ser capaz de correr en una máquina distinta a la utilizada por Moodle, para evitar sobrecargas de trabajo en dicho servidor. A pesar de ello, nuestro sistema puede perfectamente ejecutarse sobre la misma máquina, aunque la mejor opción para garantizar una mayor seguridad y rendimiento es instalar dicha aplicación en una máquina separada.

Una de las mayores desventajas que podemos encontrar en este modelo es una alta congestión del tráfico. Para evitarlo se ha reducido el número de accesos de nuestro sistema a la base de datos de Moodle para solicitar información. Cada quince minutos se ejecutará una consulta que pida toda la información existente sobre calificaciones junto con la información asociada a ellas. El sistema utilizará la información almacenada en la base de datos de seguridad sobre los usuarios para enviar los avisos y, sólo si fuera necesario, solicitaría de nuevo datos sobre ellos para actualizar dicha información. En caso extremo, el sistema realizará una tercera consulta, solicitando los datos de un usuario que no posea los permisos necesarios para modificar una nota y, sin embargo, lo haya hecho. Ésta última consulta será más rápida, puesto que involucra únicamente una tabla.

Por último, podemos decir que el tener en la base de datos de seguridad toda la información del sistema de manera centralizada nos permite realizar copias de seguridad de dichos datos periódicamente, de una manera sencilla y rápida.

Como conclusión y teniendo en cuenta la disponibilidad de máquinas existente en la empresa para la que fue diseñado e implementado este proyecto, el sistema será desarrollado para funcionar sobre un servidor con sistema operativo Linux y hará consultas a la base de datos de Moodle instalado en otro.

5.3 Diseño de Algoritmos

5.3.1. Introducción

La información necesaria para el funcionamiento de nuestra aplicación se encuentra almacenada en la base de datos de Moodle. La conexión se realizará únicamente por tres motivos:

- Cada quince minutos para solicitar información sobre las calificaciones.
- Al detectarse una inconsistencia en las calificaciones y no tener datos almacenados sobre el usuario que la modificó:
  - Se solicitarán los datos sobre todos los "Gestores", "Profesores" y "Profesores sin Permiso de Edición" asociados a todo el sistema o a algún curso.
  - Si tras la primera búsqueda de usuarios, el que modificó la nota no tiene permisos de edición para el curso, se le considerará un intruso y se solicitará información sobre él a la base de datos de Moodle para poder informar al Administrador.

En ningún momento se modificarán valores de manera directa sobre la base da datos de Moodle, para evitar inconsistencias en la multitud de tablas relacionadas que existen, para cualquier modificación que fuera necesaria, se utilizará la interfaz gráfica. Formará parte de las atribuciones de los profesores o gestores el mantener la consistencia en la plataforma.

Para mayor información sobre las tablas que se mencionan a continuación véase el Capítulo 2 de esta memoria.

5.3.2. Selección de Datos

Previo al diseño de algoritmos se debe tener en cuenta a quién va a ir dirigida la información que estamos solicitando:

- Los identificadores numéricos de cursos, actividades y alumnos no tendrán sentido para los profesores encargados de los cursos. La mayor parte de ellos no poseerá los conocimientos suficientes como para entenderlos, ni sabrán acceder a la base de datos para consultar a quien se refieren. Por tanto, será necesario conocer los nombres de cursos, alumnos y actividades, para informar a los profesores cuando se detecte una inconsistencia en las notas. De esta manera se podrá comprobar la incidencia utilizando el interfaz gráfico de Moodle.
- Por otra parte, para el Administrador de la plataforma Moodle, ciertos identificadores, como el de los usuarios intrusos, podrá tener sentido, por lo que esta información se adjuntará a los nombre de cursos, alumnos y actividades.

Se solicita toda la información en una sola llamada para evitar el cuello de botella que supone la conexión innecesaria con la base de datos de Moodle. Otra

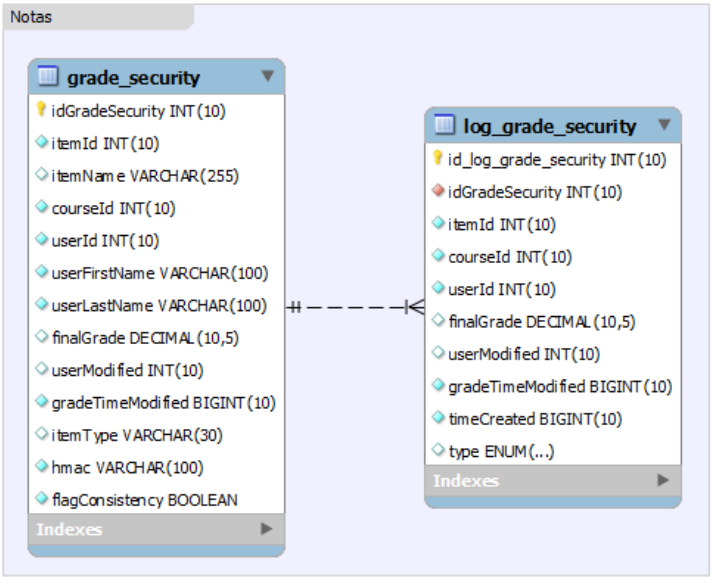
información relevante será el nombre completo de profesores y administradores, para poder dirigirse a ellos en los avisos, utilizando su nombre y apellidos.

Las fechas en la base de datos de Moodle se almacenan como la secuencia de caracteres Timestamp que contiene la fecha y la hora en la que ocurrió el evento. Para que esta fecha sea comprensible tanto para Administradores como para Profesores en los avisos que se generen, se realizará una transformación al formato "d M Y D, H:i:s e", por ejemplo, "15 May 2012 Tue, 10:20:09 Europe/Berlin". Siendo los campos:

- d: Día del mes. Desde 01 hasta 31.
- M: Representación textual corta de un mes, tres letras. Desde Jan hasta Dec.
- Y: Representación numérica del año, 4 dígitos. Ejemplos: 1999, 2012…
- D: Representación textual del día, tres letras. Desde Mon hasta Sun.
- H: Formato de 24 horas de una hora. Desde 00 hasta 23.
- i: Minutos. Desde 00 hasta 59.
- s: Segundos. Desde 00 hasta 59.
- e: Abreviatura de la zona horaria. Ejemplos: UTC, GMT, Atlantic/Azores…

5.3.3. Calificaciones

De toda la información almacenada en la base de datos de Moodle, la información relevante sobre calificaciones queda almacenada en la base de datos de seguridad en las siguientes tablas:



Tablas de Calificaciones en la base de datos de Seguridad

5.3.3.1 Tabla GRADE_SECURITY

Esta tabla relaciona las calificaciones con sus usuarios en una actividad y un curso concretos. Para que se pueda ejecutar la aplicación cada 15 minutos, toda la información se almacenará en la misma tabla, debido a que los "joins" entre tablas ralentizarían mucho el proceso.

Campo	Descripción
idGradeSecurity	Identificador único de la copia de seguridad de la nota en el sistema. (PK)
itemId	Identificador de la actividad a la que está asociada la calificación.
itemName	Nombre de la actividad.
courseId	Identificador del curso al que está asociada la actividad calificada.
userId	Identificador del usuario (alumno) al que pertenece la nota.
userFirstName	Nombre del Usuario.
userLastName	Apellido/s del Usuario.
finalGrade	Calificación.
userModified	Identificador del usuario que modificó la nota.
gradeTimeModified	Fecha en la que se modificó por última vez la nota.
itemType	Tipo de actividad. Nos interesan aquellas que tienen valor 'course' en este campo, puesto que corresponden a la nota global final del curso.
hmac	Almacena la información relevante con un cifrado que combina función hash, cifrado SHA-256, y clave secreta.
flagConsistency	Flag utilizado para indicar si se ha producido un cambio en la nota: 0= La calificación no ha sido modificada. 1= La calificación ha sido modificada y está en espera de la resolución del conflicto. 2= Se ha detectado un acceso ilícito a la base de datos de seguridad, se avisa al administrador cada quince minutos de la incidencia. 3= Se ha detectado un acceso ilícito a la base de datos de seguridad, se ha detenido el flujo de avisos mientras se resuelve la inidencia.

5.3.3.2 Tabla LOG_GRADE_SECURITY

Esta tabla mantiene el registro de los conflictos entre las calificaciones almacenadas y las calificaciones recibidas de Moodle.

Campo	Descripción
Id_log_grade_security	Identificador único de un conflicto entre calificaciones. (PK)
idGradeSecurity	Identificador de la copia de seguridad de la nota a la que hace referencia. (PK)
itemId	Identificador de la actividad a la que está asociada la calificación.
courseId	Identificador del curso al que está asociada la actividad calificada.
userId	Identificador del usuario (alumno) al que pertenece la nota.
finalGrade	Calificación.
userModified	Identificador del usuario que modificó la nota.
gradeTimeModified	Fecha en la que se modificó por última vez la nota.
timeCreated	Fecha en la que se registró el conflicto entre las calificaciones.
type	Tipo de inserción: 'Inconsistencia' o 'Resolución de Inconsistencia'.

5.3.3.3 Tablas de Moodle Implicadas

Las tablas de Moodle utilizadas para obtener dicha información son:

- "grade_items", contiene todas aquellas actividades que pueden tener una calificación asociada. Nuestra consulta sólo devolverá las actividades que tengan una calificación asociada. Los campos con datos relevantes son:
  - "id": Identificador único de la actividad en la base de datos.
  - "courseid": Identificador del curso al que pertenece la actividad calificable.
  - "itemname": Nombre de la actividad.
  - "itemtype": Tipo de actividad. Relevante cuando se quieren ejecutar consultas que devuelvan solamente las calificaciones globales del curso, valor "course".

- "grade_grades", contiene las notas numéricas. Los campos con datos relevantes son:
    - "itemid": Identificador de la actividad a la que pertenece la nota.
    - "userid": Identificador del alumno al que pertenece la nota.
    - "usermodified": Identificador del usuario que modificó la nota por última vez.
    - "finalgrade": Nota final almacenada asociada a la actividad para ese alumno.
    - "timemodified": Fecha de la última modificación de la nota.
- "user", contiene información sobre los usuarios existentes en la plataforma. Los campos con datos relevantes son:
    - "id": Identificador único del usuario en la base de datos.
    - "firstname": Nombre real del usuario.
    - "lastname": Apellido(s) reales del usuario.

Finalmente, para optimizar el algoritmo, se busca una forma de evitar que se sigan solicitando las calificaciones de un curso cuando hayan pasado 30 días, en nuestro caso, desde su finalización. Una vez transcurrido este tiempo, a falta de resolución de conflictos si los hubiera, se consideraría que las notas no van a volver a ser modificadas y que la nota real almacenada en nuestra base de datos cifrada es la auténtica. La tabla de Moodle utilizada para obtener esta información es:

- "course_completion_criteria", contiene los criterios de finalización asociados a un curso. Los campos con datos relevantes son:
    - "course": Identificador del curso al que se asocia el criterio de finalización.
    - "criteriatype": Tipo de criterio de finalización. Nos interesan aquellos cuyo tipo sea date, valor numérico 2.
    - "timeend": Contiene la fecha de finalización del curso, si criteriatype=2.

5.3.3.4 Consulta

El código de la consulta, para recuperar la información necesaria, es por tanto:

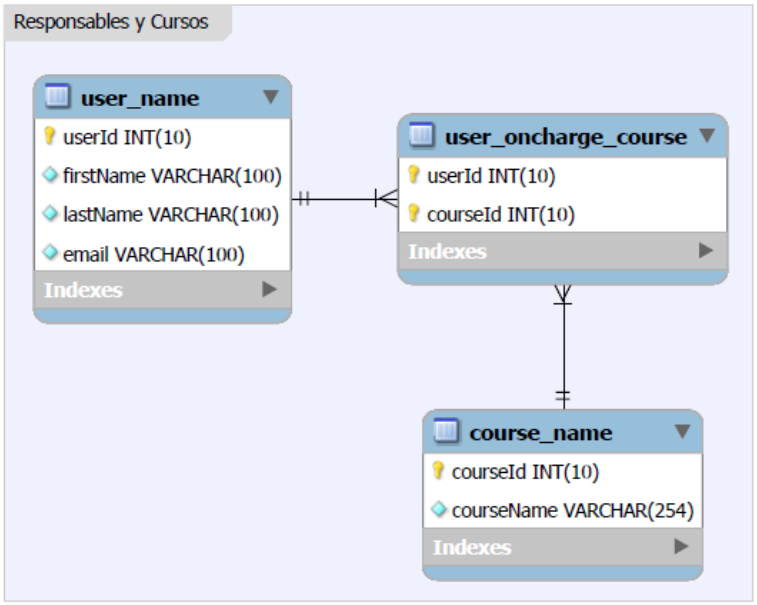
```
SELECT
        item.id,
        item.itemname,
        item.courseid,
        grade.userid,
        user.firstname,
        user.lastname,
        grade.finalgrade,
        grade.timemodified,
        item.itemtype,
        grade.usermodified
FROM
        moodle.mdl_grade_items item,
        moodle.mdl_grade_grades grade,
        moodle.mdl_user user
WHERE
        item.id = grade.itemid and
        user.id = grade.userid and
        courseid not in (
            SELECT
                course
            FROM
               moodlelocal1.mdl_course_completion_criteria
               CC
           WHERE
                (CC.criteriatype = 2 and
                 UNIX_TIMESTAMP
            (DATE_SUB(sysdate(),INTERVAL
                                30 DAY)) >= CC.timeend));
```

5.3.4. Usuarios

Como se comentó en el Capítulo 2, en Moodle existen ocho roles por defecto; de esos ocho sólo estamos interesados en aquellos que pueden modificar una nota sin ser considerados intrusos:

- "Gestor", queda asociado al contexto "sistema" (10) y puede modificar todas las notas existentes en el mismo.
- "Creador de curso", queda inscrito con el rol "Profesor" una vez que crea el curso.
- "Profesor" y "Profesor sin Permisos de Edición", queda asociado al contexto "curso" (50) y puede modificar todas las notas existentes en el mismo.

Una vez elegidos los roles, la información relevante sobre esos usuarios y los contextos sobre los que tienen permiso queda almacenada en las siguientes tablas:



Tablas de Usuarios-Cursos en la base de datos de Seguridad

5.3.4.1 Tabla USER_NAME

Esta tabla contiene la información sobre los usuarios que tienen permiso para modificar calificaciones en algún contexto determinado. Dado que los avisos se envían por correo electrónico, es necesario conocer tanto el email del usuario como su nombre completo para poder dirigirse a él.

Campo	Descripción
userId	Identificador único del Usuario. (PK)
firstName	Nombre del Usuario.
lastName	Apellido/s del Usuario.
email	Correo electrónico del Usuario.

5.3.4.2 Tabla COURSE_NAME

Esta tabla contiene información sobre los contextos, cursos o todo el sistema en nuestro caso, sobre los que se pueden tener permisos.

Campo	Descripción
courserId	Identificador único del Curso o el sistema. (PK)
courseName	Nombre del Curso.

5.3.4.3 Tabla USER_ONCHARGE_COURSE

Esta tabla relaciona a los usuarios con los cursos sobre los que tienen permisos.

Campo	Descripción
userId	Identificador único del Usuario. (PK)
courserId	Identificador único del Curso o el sistema. (PK)

5.3.4.4 Tablas de Moodle Implicadas

Las tablas de Moodle utilizadas para obtener dicha información son:

- "course", contiene información sobre los distintos cursos existentes en Moodle. Los campos con datos relevantes son:
  - "id": Identificador único del curso.
  - "fullname": Nombre completo del curso.
- "user", contiene información sobre los usuarios existentes en la plataforma. Los campos con datos relevantes son:
  - "id": Identificador único del usuario en la base de datos.
  - "firstname": Nombre real del usuario.
  - "lastname": Apellido(s) reales del usuario.
  - "email": Email del usuario al que se envía el aviso, en caso de incidencia en las calificaciones.
- "rol_assignments", asigna a un usuario un determinado rol en un contexto; es la tabla de las matriculaciones de los usuarios a los distintos contextos. Los campos con datos relevantes son:
  - "roleid": Identificador del rol con el que el usuario ha sido matriculado.
  - "contextid": Identificador del contexto en el que el usuario ha sido matriculado con ese rol.
  - "userid": Identificador del usuario que ha sido matriculado con ese rol en ese contexto.
- "rol", contiene los roles existentes en Moodle. Los campos con datos relevantes son:
  - "id": Identificador único del rol en el sistema.
  - "shortname": Nombre corto usado para referirse al rol. Permite identificar aquellos roles que tienen permisos para editar calificaciones.
- "context", contiene los diferentes contextos existentes en Moodle. Los campos con datos relevantes son:
  - "contextlevel": Nivel del contexto, código numérico que lo identifica. Nos interesan los contextos con nivel 10, "todo el sistema", y 50 "cursos".
  - "instanceid": Identificador del curso si el nivel es 50 o del sistema si el nivel es 10.

Cuando se genere un conflicto en las calificaciones, la aplicación buscará, en su propia base de datos, la información sobre el usuario que modificó las notas, campo "usermodified" de la tabla "grade_grades"; si no lo encontrara, lanzaría dos consultas solicitando los datos sobre los "Profesores", "Profesores sin permisos de Edición" y "Gestores" y los cursos sobre los que tienen permisos.

5.3.4.5 Consultas

El código de la consulta para obtener los profesores y los profesores sin permisos de edición es por tanto:

```
SELECT U.id userId,
       firstname firstName,
       lastname lastName,
       email,
       instanceid courseId,
       CO.fullname courseName
FROM
       moodle.mdl_user U,
       moodle.mdl_role_assignments R,
       moodle.mdl_context C,
       moodle.mdl_course CO
WHERE
       U.id=R.userid and
       R.contextid=C.id and instanceid=CO.id and
       (R.roleid=3 or R.roleid=4);
```

El código de la consulta para obtener los gestores es por tanto:

```
SELECT distinct
       U.id userId,
       firstname firstName,
       lastname lastName,
       email
FROM
       moodle.mdl_user U,
       moodle.mdl_role_assignments R,
       moodle.mdl_context C
WHERE
       U.id=R.userid and
       R.contextid=C.id and
       C.contextlevel=10 and
       (R.roleid=1);
```

Si, tras ejecutar estas dos búsquedas, el identificador de usuario existente en el campo "usermodified" de la tabla "grade_grades" siguiera sin existir asociado al curso en el que se modificó la calificación en la base de datos de la aplicación, el usuario será automáticamente tratado como un intruso.

Para optimizar el uso de este algoritmo se recomienda precargar los usuarios, ejecutando los scripts apropiados, una vez creados los cursos y asignados los usuarios responsables.

5.3.5. Intrusos

La interfaz gráfica de Moodle impide que un usuario sin los permisos necesarios sea capaz de cambiar de manera legal una calificación de la tabla "grade_grades". Si un usuario es detectado cambiando una nota, ya sea un profesor en cualquier curso sobre el que no tenga permisos o un usuario con el rol estudiante, será automáticamente tratado como intruso.

Cuando la aplicación detecte un conflicto en las notas y no encuentre el usuario responsable de ese cambio solicitará la información sobre "Profesores", "Profesores sin permisos de edición" y "Gestores" a la base de datos de Moodle. Si, tras esa primera petición, sigue sin localizar al usuario responsable del cambio, la aplicación buscará los datos del intruso en Moodle. No se guardará información extra del intruso más allá de su identificador en el log del conflicto entre notas.

Cuando se detecte un intruso, el aviso irá dirigido al Administrador del sistema, en vez de al profesor o profesores del mismo, que puede ser externo al sistema Moodle.

5.3.5.1 Tablas de Moodle Implicadas

La tabla de Moodle utilizada para obtener dicha información es:

- "user", contiene información sobre los usuarios existentes en la plataforma. Los campos con datos relevantes son:
  - "id": Identificador único del usuario en la base de datos.
  - "firstname": Nombre real del usuario.
  - "lastname": Apellido(s) reales del usuario.
  - "email": Email del Usuario al que se envía el aviso en caso de incidencia en las calificaciones.

5.3.5.2 Consulta

El código de la consulta para obtener información sobre el intruso es por tanto:

```
SELECT
      U.id userId,
      firstname firstName,
      lastname lastName,
      email
FROM
      moodlelocal1.mdl_user U
WHERE
      U.id=$row['usermodified'];
```

Siendo "`$row['usermodified']`" el identificador del usuario que fue detectado cambiando las calificaciones sin permiso.

5.3.6. Generación de Avisos

Cuando se detecte un cambio en las notas se generará un aviso:

- Dirigido al usuario que modificó la calificación, cuando éste posea los permisos necesarios para realizarlo.
- Dirigido al Administrador del sistema, cuando la modificación la realice un usuario sin permisos.

El primer aviso, generado en el momento en el que se detecta el conflicto en las notas, contiene la siguiente información:

- Usuario encargado del curso que modificó la calificación.
- Curso en el que se produjo la modificación.
- Actividad asociada, si procede.
- Alumno afectado por la modificación.
- Nota y Fecha inicialmente almacenadas.
- Nota y Fecha tras la modificación.
- Solicitud de identificación de la nota real entre las dos disponibles.
- Si el aviso estuviera dirigido al Administrador del sistema:
    - Información relativa al intruso que modificó las calificaciones sin los permisos necesarios.

Una vez al día, el sistema localizará los avisos que no han sido atendidos y generará un aviso recordatorio al profesor encargado o al administrador; junto a los campos enunciados en el primer aviso, se añadirá:

- Hora en la que fue detectado el conflicto.

5.4 Seguridad de la Información

Nuestro sistema ha sido diseñado para aumentar la seguridad de las calificaciones en Moodle. Para ello, no es necesario únicamente hacer una copia de seguridad de las notas en texto en claro y comprobar periódicamente que la información almacenada no ha sido alterada. Si, simplemente, se implementaran estas medidas, nuestro sistema seguiría siendo vulnerable al mismo tipo de ataques que la plataforma Moodle, es decir, las inyecciones de código SQL. Es por ello que, en la base de datos de seguridad asociada a nuestro sistema, se almacena en un campo la misma información cifrada.

Cuando se detecta una calificación que no existía previamente en la base de datos de nuestro sistema, éste selecciona parte de la información asociada junto con la nota y le aplica un algoritmo de cifrado HMAC. Si se detecta un conflicto entre la información cifrada almacenada y las calificaciones obtenidas de Moodle tras aplicarles HMAC, se realiza una segunda comprobación, la cual detectará si la integridad de nuestra base de datos de seguridad se ha visto comprometida. Si, tras esta segunda comprobación, todo está correcto, el sistema procederá a generar un aviso que informe al usuario responsable de la inconsistencia detectada.

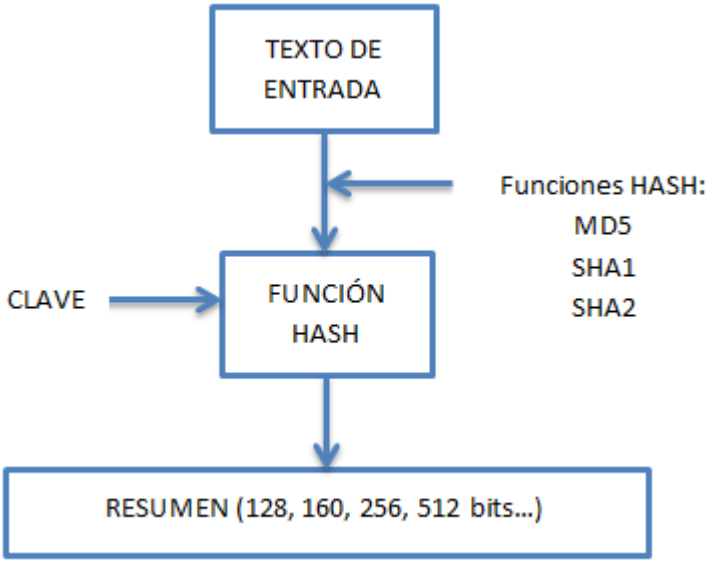
Cuando el usuario privilegiado resuelva el conflicto, el sistema recalculará la información cifrada y la almacenará actualizada en la base de datos, si fuera necesario.

5.4.1. HMAC

La función matemática que define HMAC es:

$$HMAC(K,m) = H((K \oplus opad) \,||\, H((K \oplus ipad) \,||\, m)).$$

- H(): función hash.
- K: clave secreta. Se le añaden ceros a la derecha, en caso de que su tamaño sea menor que el tamaño de bloque.
- m: texto en claro que va a ser cifrado.
- ||: denota concatenación.
- $\oplus$: operación XOR.
- opad: relleno exterior (constante “5c” repetida B veces).
- ipad: relleno interior (constante “36” repetida B veces).
- B: tamaño del bloque.



Cifrado HMAC

5.4.2. Generación de Avisos

Cuando se detecta una inconsistencia entre el valor almacenado en el campo cifrado con HMAC y la información obtenida de Moodle y posteriormente cifrada, el sistema procede a realizar una segunda comprobación de seguridad. Ésta permite detectar inyecciones de código SQL en la base de datos de seguridad implementada. La no correspondencia entre la información en claro y la cifrada almacenada será tratada como un fallo grave de seguridad.

Si esta situación se produce, se enviará un aviso al administrador del sistema, cada quince minutos, alertándolo del problema. El mensaje de aviso generado contendrá el identificador de la calificación que generó el problema, para que el Administrador pueda resolverlo.

Además de este identificador, al Administrador se le facilitarán dos opciones:

- Detener el flujo de correos, interrumpiendo la llegada de correos cada 15 minutos.
- Inconsistencia Resuelta. Una vez la inconsistencia haya sido resuelta, se le indicará a la base de datos que puede volver a comprobar la consistencia de dicha calificación con Moodle.

5.5 Diagrama de casos de uso de diseño

En el diagrama de casos de uso encontramos como novedades, respecto a su equivalente en análisis, que el actor "Tiempo" ha pasado a ser el actor "Cron", ya que será el actor encargado de lanzar la aplicación transcurrido un determinado periodo. El diagrama de casos de uso desde la perspectiva de diseño es el siguiente:

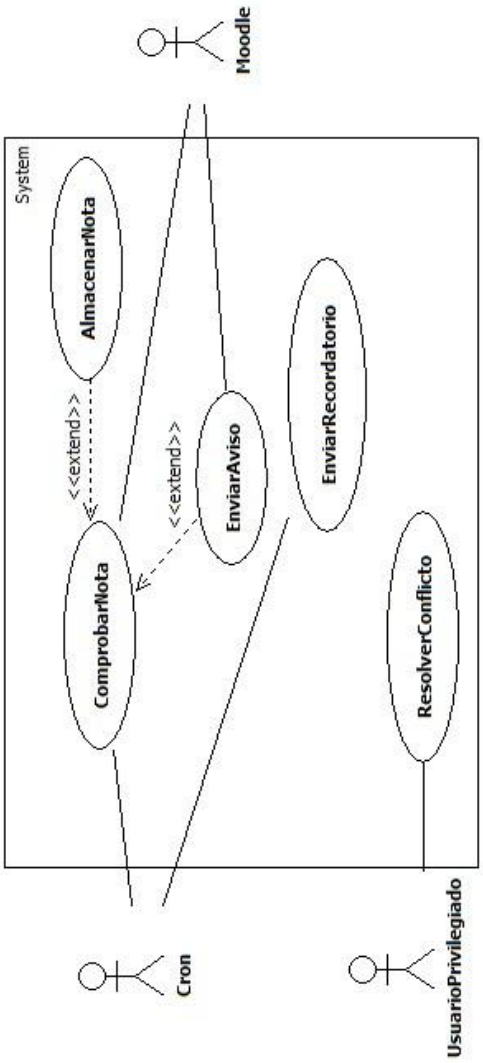


Diagrama de Casos de Uso de Diseño

5.6 Especificación de casos de uso de diseño

5.6.1. Caso de Uso: ComprobarNota

Caso de Uso: ComprobarNota.
ID: 1
Breve Descripción: El caso de uso empieza cuando, pasado un determinado periodo de tiempo, el sistema solicita todas las calificaciones existentes en el sistema.
Actores Principales: Cron.
Actores Secundarios: Moodle.
Precondiciones: El sistema es capaz de conectarse a Moodle para conseguir la información necesaria sobre las notas.
Flujo Principal: 1 El caso de uso se inicia cuando, pasado un determinado periodo de tiempo, cron inicia la aplicación que comprueba que las notas de Moodle no han sido modificadas. 2 El sistema se conecta a la base de datos “Moodle” y solicita a través de una consulta las notas almacenadas en ella. 3 Moodle devuelve la información solicitada a la aplicación, aquella perteneciente a las tablas “grade_items”, “grade_grades”, “user” y “course_completion_criteria”. 4 Mientras haya notas por comparar, el sistema comprueba si la nota ya se encontraba almacenada en la base de datos local “gradessecurity”. 4.1 Si una nota tiene el mismo identificador de curso, ítem y usuario, entonces existe en el sistema: Compara el valor gradeFinal obtenido en la consulta con el almacenado en la base de datos “gradessecurity”. 4.1.1 Si coinciden: Pasa a la siguiente nota a comparar. 4.1.2 Si no coinciden: Extend(EnviarAviso). 4.2 Si no se encuentra una coincidencia: Extend(AlmacenarNota). 5 El caso de uso finaliza cuando ya se han comparado todas las notas obtenidas en la ejecución de la consulta.
Postcondiciones: Se ha comprobado el valor de todas las notas obtenidas en la consulta de Moodle.
Flujos Alternativos: Ninguno.

5.6.2. Caso de Uso: AlmacenarNota.

Caso de Uso: AlmacenarNota.
ID: 2
Breve Descripción: Una nota nueva se almacena en el sistema.
Actores Principales: Cron.
Actores Secundarios: Ninguno.
Precondiciones: La nota obtenida en la consulta a la base de datos “Moodle” no existía con anterioridad en el sistema.
Flujo Principal: 1 El caso de uso se inicia cuando el sistema busca en la base de datos local “gradessecurity” una nota con el mismo identificador de actividad, curso y usuario que los obtenidos en la consulta a la base de datos de “moodle” y no encuentra ninguna coincidencia. 2 El sistema cifra cierta información de la nota, la resume utilizando una función hash y almacena dicha información junto con la obtenida en la consulta con esa nueva información en la base de datos local “gradessecurity”. 3 El caso de uso finaliza cuando la nota ha sido correctamente almacenada en la base de datos local.
Postcondiciones: La nueva información sobre la calificación queda almacenada en la base de datos local “gradessecurity”.
Flujos Alternativos: Ninguno.

5.6.3. Caso de Uso: EnviarAviso.

Caso de Uso: EnviarAviso.
ID: 3
Breve Descripción: El sistema enviará un aviso al usuario responsable para informar de la inconsistencia en la calificación.
Actores Principales: Cron.
Actores Secundarios: Moodle.
Precondiciones: La nota obtenida en la base de datos “moodle” existe en la base de datos local del sistema “gradessecurity” con el valor del campo “finalgrade” distinto.
Flujo Principal: 1 El caso de uso se inicia cuando el sistema busca en la base de datos local “gradessecurity” una nota con el mismo identificador de actividad, curso y usuario que los obtenidos en la consulta a la base de datos de “moodle” encuentra una coincidencia y al comparar el valor del campo “finalgrade” detecta que el valor es distinto. 2 El sistema busca la información del usuario que modificó la calificación en la base de datos local “gradessecurity” y genera un aviso con la siguiente información: 2.1 Usuario encargado del curso que modificó la calificación. 2.2 Curso en el que se produjo la modificación. 2.3 Actividad asociada, si procede. 2.4 Alumno afectado por la modificación. 2.5 Nota y Fecha almacenadas en local. 2.6 Nota y Fecha obtenidas en la consulta a Moodle. 2.7 Solicitud de identificación de la nota real entre las dos disponibles. 3 El caso de uso finaliza cuando el aviso ha sido enviado.
Postcondiciones: Se ha generado y enviado un aviso a la persona que modificó la calificación.
Flujos Alternativos: DatosDeUsuarioNoEncontrados, SinPermisosNecesarios.

5.6.3.1 Flujo Alternativo: EnviarAviso:DatosDeUsuarioNoEncontrados

Flujo Alternativo: EnviarAviso:SinPermisosNecesarios.
ID: 3.1
Breve Descripción: El Sistema no encuentra los datos del Usuario que modificó la calificación en la base de datos local “gradessecurity”.
Flujo Alternativo: 1 El flujo alternativo comienza cuando el Sistema detecta que el usuario que modificó la calificación no tenía los permisos para realizar tal acción. 2 El Sistema se conecta a la base de datos de Moodle y solicita la información sobre el usuario que modificó la calificación con una consulta. 3 Moodle devuelve la información solicitada a la aplicación, aquella perteneciente a las tablas “user”, “role_assigments”, “role”, “context” y “course”. 4 Mientras haya usuarios por comparar, el sistema comprueba si la información sobre el usuario ya se encontraba almacenada en la base de datos local “gradessecurity”: 4.1 Si existe: Pasa al siguiente usuario. 4.2 Si no coinciden: almacena la información en dicha base. 5 El flujo alternativo finaliza cuando ya se han comparado todos los usuarios obtenidos en la ejecución de la consulta.

5.6.3.2 Flujo Alternativo: EnviarAviso:SinPermisosNecesarios.

Flujo Alternativo: EnviarAviso:SinPermisosNecesarios.
ID: 3.2
Breve Descripción: El Sistema detecta que el usuario que modificó la calificación no tenía los permisos necesarios para realizar tal acción.
Flujo Alternativo: 1 El flujo alternativo comienza cuando el Sistema detecta que el usuario que modificó la calificación no tenía los permisos para realizar tal acción. 2 El Sistema se conecta a la base de datos de Moodle y solicita la información sobre el usuario que modificó la calificación con una consulta. 3 Moodle devuelve la información solicitada a la aplicación, aquella perteneciente a la tabla “user”. 4 El Sistema genera un aviso que enviará a un Usuario Privilegiado; en este caso el Administrador del sistema, con la siguiente información: 4.1 Información relativa al intruso que modificó las calificaciones sin los permisos necesarios. 4.2 Curso en el que se produjo la modificación. 4.3 Actividad asociada, si procede. 4.4 Alumno afectado por la modificación. 4.5 Nota y Fecha almacenadas en local. 4.6 Nota y Fecha obtenidas en la consulta a Moodle. 4.7 Solicitud de identificación de la nota real entre las dos disponibles. 5 El flujo alternativo finaliza cuando el aviso ha sido enviado.

5.6.4. Caso de Uso: EnviarRecordatorio.

Caso de Uso: EnviarRecordatorio.
ID: 4
Breve Descripción: El sistema enviará un aviso recordatorio, indicando que se ha detectado un conflicto en las calificaciones que aún no ha sido resuelto.
Actores Principales: Cron.
Actores Secundarios: Ninguno.
Precondiciones: Al menos una nota generó un aviso anterior, por un conflicto entre el valor almacenado en la base de datos local “gradessecurity” y el obtenido en la consulta a la base de datos “moodle”, y éste no fue atendido a una hora determinada del día.
Flujo Principal: 1 El caso de uso se inicia cuando el sistema detecta que al menos una calificación generó un aviso anterior, por un conflicto entre el valor almacenado en la base de datos local “gradessecurity” y el obtenido en la consulta a la base de datos “moodle”, y éste no fue atendido a una hora determinada del día. 2 Mientras haya inconsistencias sin resolver, el sistema busca la información del usuario que modificó la calificación en la base de datos local “gradessecurity” y genera un aviso recordatorio con la siguiente información: 2.1 Dependiendo de quién modificara la calificación: 2.1.1 Si el usuario tenía permisos para realizar tal acción: Usuario encargado del curso que modificó la calificación. 2.1.2 Si el usuario no tenía permisos para realizar tal acción: Información relativa al intruso que modificó las calificaciones sin los permisos necesarios. 2.2 Curso en el que se produjo la modificación. 2.3 Actividad asociada, si procede. 2.4 Alumno afectado por la modificación. 2.5 Nota y Fecha almacenadas en local. 2.6 Nota y Fecha obtenidas en la consulta a Moodle. 2.7 Solicitud de identificación de la nota real entre las dos disponibles. 3 El caso de uso finaliza cuando el aviso recordatorio ha sido enviado.
Postcondiciones: Se ha generado y enviado un aviso recordatorio a la persona que modificó la calificación o al Administrador del sistema, en caso de ser una modificación ilícita.
Flujos Alternativos: Ninguno.

5.6.5. Caso de Uso: ResolverConflicto.

Caso de Uso: ResolverConflicto.
ID: 5
Breve Descripción: El Usuario resuelve un conflicto en las notas.
Actores Principales: Usuario Privilegiado.
Actores Secundarios: Ninguno.
Precondiciones: El Usuario ha recibido un aviso que le indica se ha detectado una nota con un conflicto entre el valor almacenado en la base de datos local "gradessecurity" y el obtenido en la consulta a la base de datos de Moodle.
Flujo Principal: 1 El caso de uso se inicia cuando el Usuario Privilegiado, tras recibir un aviso sobre una inconsistencia en las calificaciones, desea resolver el conflicto detectado. El aviso posee dos enlaces, cada uno de ellos asociado a una de las posibles calificaciones. 2 El Usuario Privilegiado hace clic sobre el enlace correspondiente a la calificación correcta almacenada en el sistema. 3 El sistema recibe la información correcta indicada por el usuario, la almacena en la base de datos "gradessecurity" e informa al usuario por pantalla. 4 El caso de uso finaliza cuando el Usuario Privilegiado ha sido correctamente informado.
Postcondiciones: Se ha resuelto el conflicto en las notas.
Flujos Alternativos: Ninguno.

5.7 Realización de casos de uso de diseño

5.7.1. Diagramas de Secuencia/Interacción

5.7.1.1 Diagrama de Secuencia: ComprobarNota

Debido a su envergadura, la versión completa del diagrama de secuencia de diseño correspondiente al caso de uso "ComprobarNota" y a los dos <<extend>> relacionados con él, "Almacenar Nota" y "Enviar Aviso", se encontrará recogida en el disco compacto que acompaña al proyecto. En este documento se mostrará el diagrama en dos partes, separando los accesos a la base de datos de Moodle, externa a nuestro sistema de la que simplemente obtiene información, del resto del caso de uso.

a) Comprobar Nota:

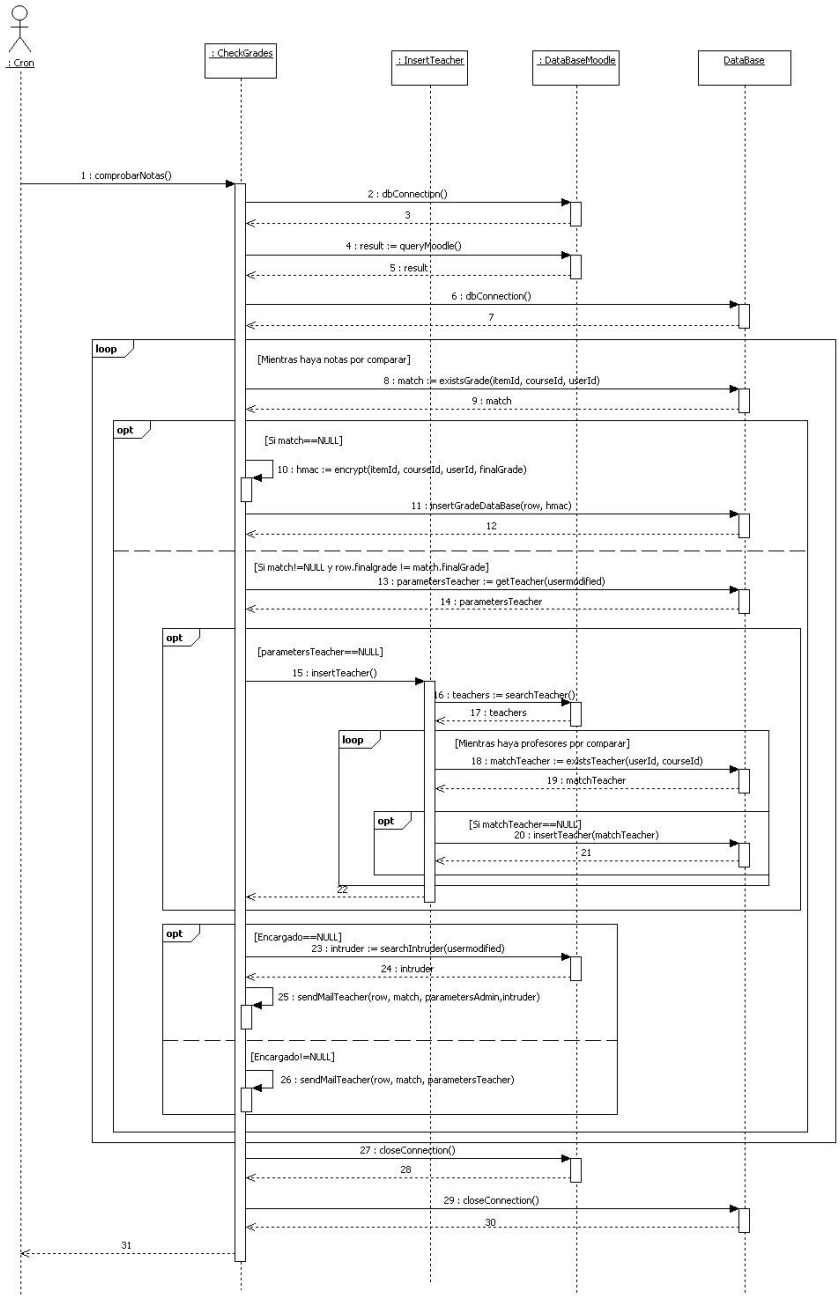


Diagrama de Secuencia de Diseño 1

b) Llamadas a la base de datos de Moodle:

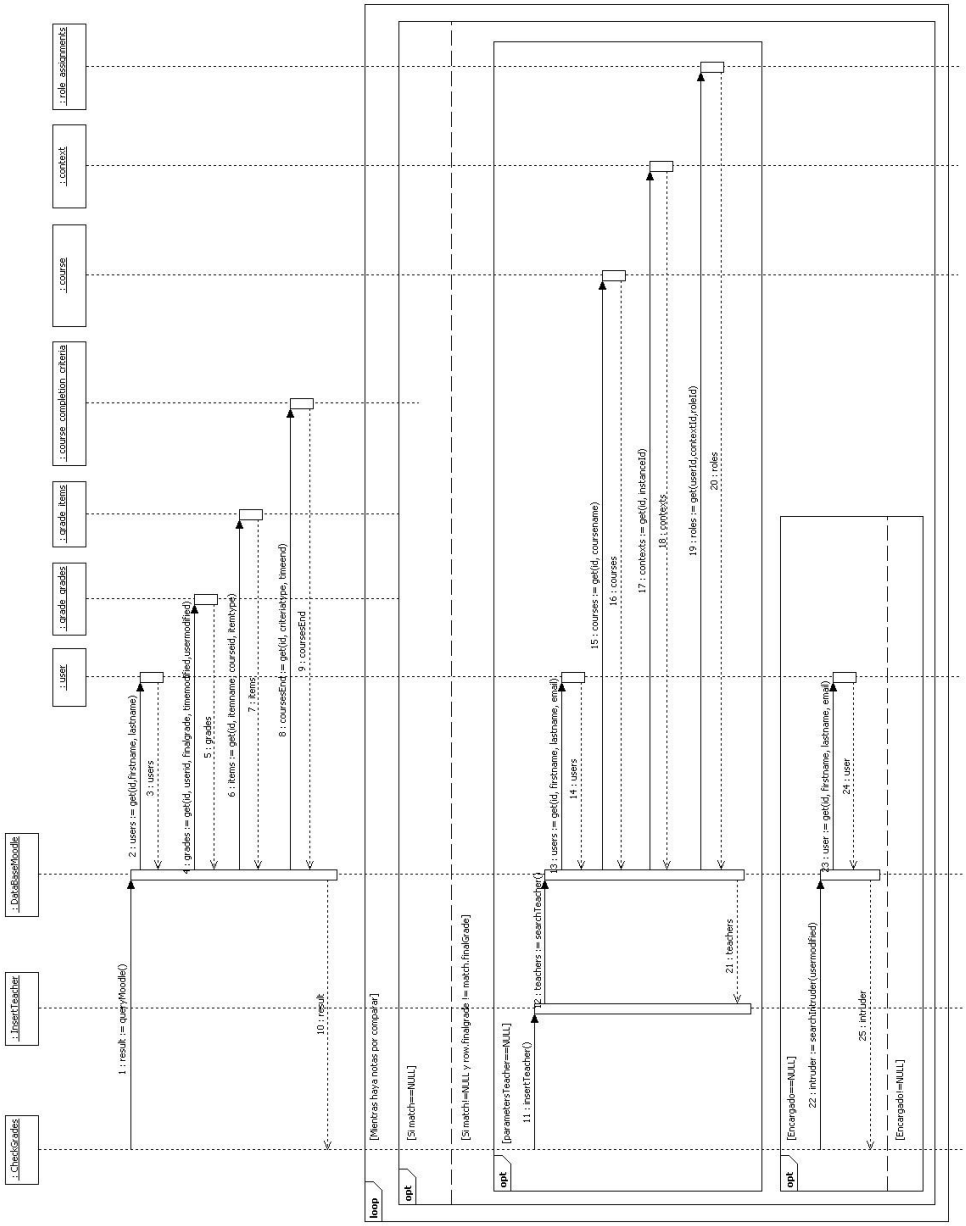


Diagrama de Secuencia de Diseño 1 – Llamadas a Moodle

5.7.1.2 Diagrama de Secuencia: EnviarRecordatorio

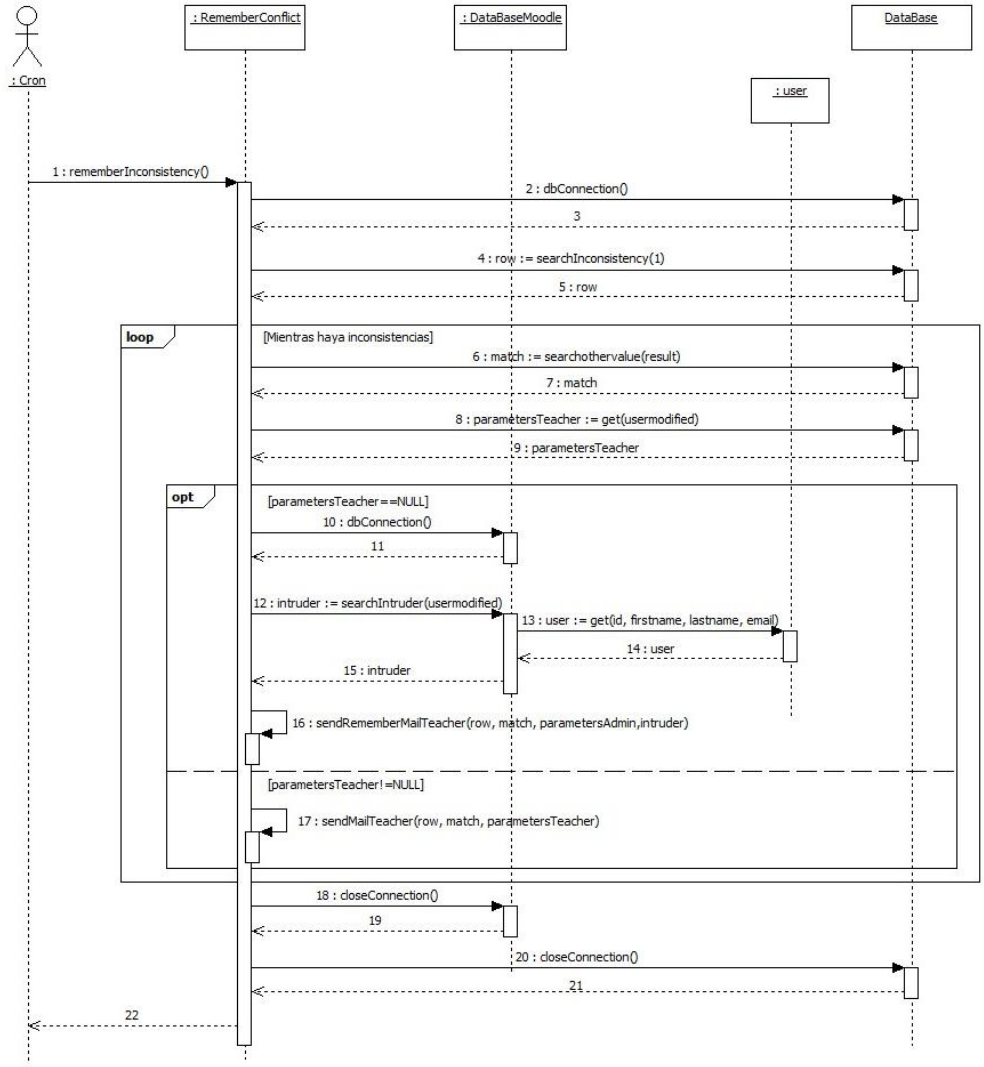


Diagrama de Secuencia de Diseño 2

5.7.1.3 Diagrama de Secuencia: ResolverConflicto

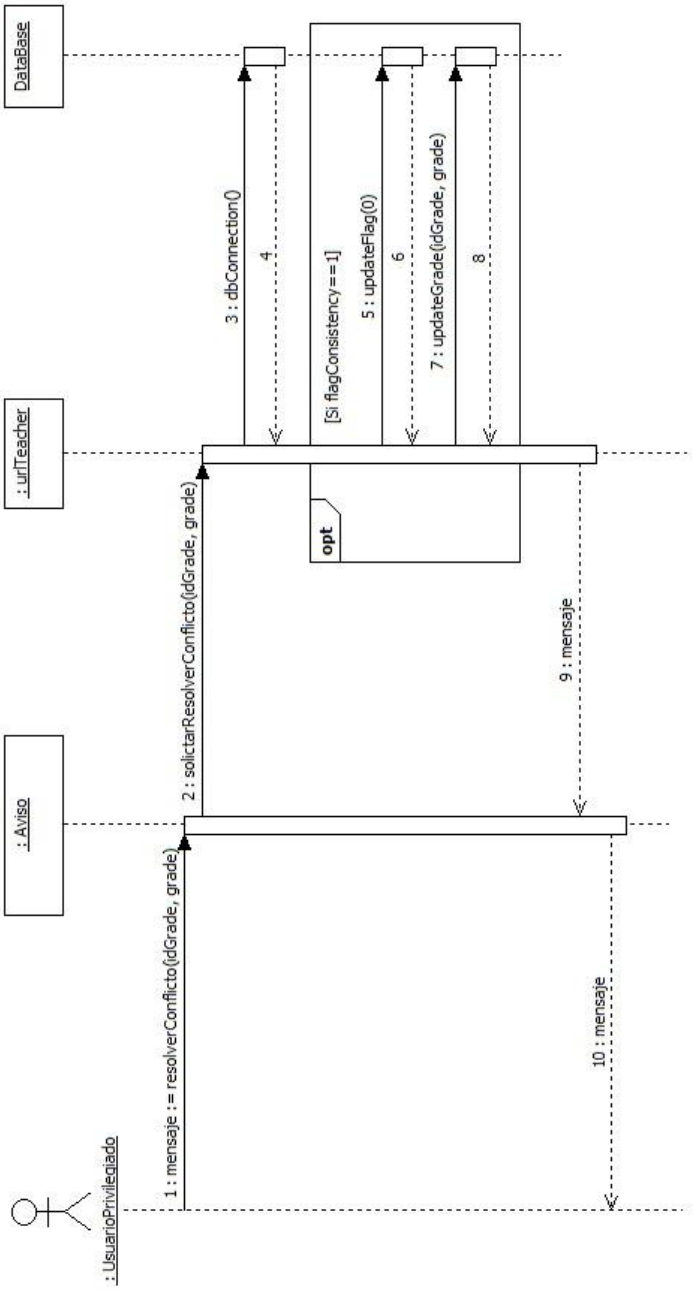


Diagrama de Secuencia de Diseño 3

5.8 Diagrama de clases de diseño.

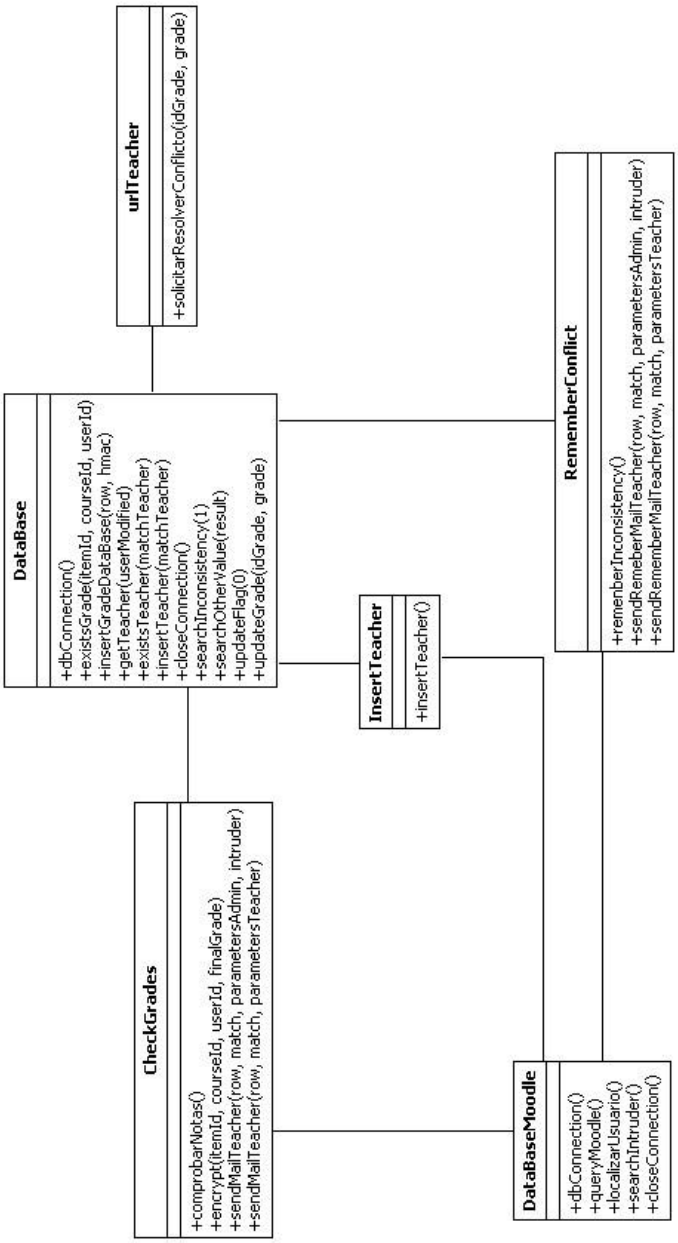


Diagrama de Clases de Diseño

6 DOCUMENTO DE IMPLEMENTACIÓN

6.1 Introducción.

En este documento de implementación se comentarán brevemente las tecnologías y los lenguajes de programación utilizados a lo largo del desarrollo de este proyecto. Prescindiremos de la inclusión de fragmentos de código ya que resultan poco informativos de cara al lector, aunque si éste está interesado en algún aspecto concreto de la codificación, le remitimos al dispositivo de almacenamiento suministrado junto con la documentación donde están incluidos todos los ficheros de código fuente utilizados en el desarrollo del proyecto.

6.2 Tecnologías Utilizadas

6.2.1. Moodle

Moodle es un paquete de software para la creación de cursos y sitios Web basados en Internet. Es un proyecto en desarrollo diseñado para dar soporte a un marco de educación social constructivista. Una herramienta que permite que una organización, por ejemplo una universidad, gestione de una manera eficaz recursos educativos proporcionados por un profesorado para la docencia de los alumnos del curso.

Se distribuye gratuitamente como software libre, bajo la Licencia Pública GNU. Esto implica que existe una gran comunidad de usuarios que desarrollan aplicaciones para la plataforma y nutren a ésta de nuevas funcionalidades que se pueden obtener en el propio sitio web de la plataforma.

Moodle puede funcionar en cualquier servidor en el que pueda correr PHP y soporta varios tipos de bases de datos, en especial MySQL.

La página web [moodle.org] proporciona un punto central de información, discusión y colaboración entre los usuarios de Moodle, incluyendo administradores de sistemas, profesores, investigadores, diseñadores de sistemas de formación y, por supuesto, desarrolladores de todo el mundo. Al igual que Moodle, su web está continuamente evolucionando para ajustarse a las necesidades de la comunidad.

6.2.2. Apache

El servidor Apache es un servidor web HTTP de código abierto, para plataformas Unix, Microsoft Windows y Macintosh entre otras, que implementa el protocolo HTTP/1.12 y la noción de sitio virtual.

Apache es utilizado para tareas donde el contenido necesita ser puesto a disposición de otros usuarios de forma segura y confiable. Los programadores se valen de una versión local de Apache con el fin de previsualizar y probar código mientras éste es desarrollado.

6.2.3. NetBeans

El IDE NetBeans es un entorno de desarrollo integrado, una herramienta para programadores pensada para escribir, compilar, depurar y ejecutar programas. El NetBeans Enterprise Pack soporta el desarrollo de Aplicaciones con Java EE5, incluyendo herramientas de desarrollo visuales de SOA, herramientas de esquemas XML, orientación a servicios Web y modelado UML. El NetBeans C/C++ Pack soporta proyectos de C/C++, mientras el PHP Pack, soporta PHP 5.

La plataforma NetBeans permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados módulos. Un módulo es un archivo Java que contiene clases escritas para interactuar con las APIs de NetBeans y un archivo especial, manifest file, que lo identifica como módulo. Las aplicaciones construidas a partir de módulos pueden ser extendidas agregándoles nuevos módulos. Debido a que los módulos pueden ser desarrollados independientemente, las aplicaciones basadas en la plataforma NetBeans pueden ser extendidas fácilmente por otros desarrolladores de software.

6.2.4. XAMPP

XAMPP es un servidor independiente de plataforma, que consiste principalmente en una base de datos MySQL, el servidor web Apache y los intérpretes para lenguajes de script PHP y Perl. El nombre proviene del acrónimo de X (para cualquiera de los sistemas operativos), Apache, MySQL, PHP y Perl.

El programa está liberado bajo la licencia GNU y actúa como un servidor Web libre, fácil de utilizar y capaz de interpretar páginas dinámicas. Actualmente XAMPP está disponible para Microsoft Windows, GNU/Linux, Solaris y MacOS X.

6.2.5. MySQLWorkBench

MySQL WorkBench es una herramienta visual de diseño de bases de datos que integra desarrollo software, administración, diseño, creación y mantenimiento de las mismas. Es el sucesor de DBDesigner 4 de fabFORCE.net y remplaza el anterior conjunto de software, MySQL GUI Tools Bundle.

MySQL WorkBench posee una potente interfaz gráfica que permite:

- Administrar bases de datos MySQL, centralizando las operaciones relativas a creación y administración de esquemas, tablas, campos y otros objetos SQL. Además, admite la creación de cuentas de usuarios para el acceso y la revisión de archivos vitales de la plataforma de base de datos, como revisión de logs del servidor de bases de datos.
- Generar consultas y código SQL, la herramienta incorpora un módulo para el desarrollo de código SQL de una forma más visual y fácil.
- Diseñar Modelos de Datos, MySQL WorkBench permite obtener código SQL a través del diseño de diagramas entidad-relación y, asimismo, gracias a la ingeniería inversa, conseguir el diagrama ER a partir de la relación de tablas en SQL.

6.2.6. MySQL

MySQL es un sistema de gestión de bases de datos relacional, multihilo y multiusuario. Se ofrece bajo una licencia GNU GLP.

Existen varias APIs que permiten, a aplicaciones escritas en diversos lenguajes de programación, acceder a las bases de datos MySQL, incluyendo C, C++, C#, Pascal, Delphi, Eiffel, Smalltalk, Java, Lisp, Perl, PHP, Python, Ruby, Gambas, entre otros; cada uno de estos utiliza una API específica. También existe una interfaz ODBC, llamado MyODBC que permite a cualquier lenguaje de programación que soporte ODBC comunicarse con las bases de datos MySQL.

6.3 Lenguajes utilizados

6.3.1. SQL

El lenguaje de consulta estructurado es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en éstas. Una de sus características es el manejo del álgebra y el cálculo relacional permitiendo efectuar consultas con el fin de recuperar información de interés de una base de datos, así como también hacer cambios sobre ella.

6.3.2. PHP

PHP es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas Web dinámicas. Se usa principalmente para la interpretación del lado del servidor, server-side scripting, aunque actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica usando las bibliotecas Qt o GTK+.

Cuando el cliente hace una petición al servidor para que le envíe una página Web, el servidor ejecuta el intérprete de PHP. Éste procesa el script solicitado que generará el contenido de manera dinámica, por ejemplo obteniendo información de una base de datos. El resultado es enviado por el intérprete al servidor, quien a su vez se lo envía al cliente. Mediante extensiones es también posible la generación de archivos PDF, Flash, así como imágenes en diferentes formatos.

Algunas ventajas de PHP son:

- Multiplataforma.
- Completamente orientado a la Web.
- Facilidad de conexión con la mayoría de motores de base de datos actuales.
- Capacidad de expandir su funcionalidad mediante módulos (exts).
- Amplia documentación disponible.
- Permite aplicar técnicas de orientación al objeto.
- Biblioteca nativa de funciones sumamente amplia e incluida.
- Débilmente tipado.
- Manejo de excepciones.
- Da libertad al programador para aplicar diferentes técnicas de programación y desarrollo.

6.3.3. CRON

Cron es un administrador regular de procesos en segundo plano, daemon, que ejecuta procesos o guiones a intervalos regulares. Los procesos que deben ejecutarse y la hora en la que deben hacerlo se especifican en el fichero /etc/crontab. Cada línea de un archivo crontab representa un trabajo y está compuesto por una expresión CRON, seguida por un comando Shell Script para ejecutarse. Para agregar, quitar o modificar tareas, hay que editar el archivo crontab. Esto se hace con la orden crontab -e.

Los comandos en cron tienen la siguiente estructura:

```
.---------------- minuto (0 - 59)
| .------------- hora (0 - 23)
| | .---------- día del mes (1 - 31)
| | | .------- mes (1 - 12) ó jan,feb,mar,apr ... (los meses
| | | | en inglés)
| | | | .---- día de la semana (0 - 6) (Domingo=0 ó 7) ó
| | | | | sun,mon,tue,wed,thu,fri,sat (los días en inglés)
| | | | |
| | | | |
* * * * * ruta del script a ejecutar
```

6.3.4. UML

Lenguaje Unificado de Modelado; es el lenguaje de modelado de sistemas de software estándar; está respaldado por el OMG, Object Management Group. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un modelo del sistema, incluyendo aspectos conceptuales tales como procesos de negocio, funciones del sistema y aspectos concretos como expresiones de lenguajes de programación y esquemas de bases de datos.

En el desarrollo de software se pueden aplicar gran variedad de metodologías, como por ejemplo el Proceso Unificado Racional o RUP UML no especifica qué metodología o proceso se debe seguir.

6.3.5. HTML

HTML, siglas de HyperText Markup Language, es el lenguaje de marcado predominante para la elaboración de páginas Web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes. El HTML se escribe en forma de «etiquetas», rodeadas por corchetes angulares (<,>).

7 PRUEBAS

7.1 Introducción:

Para la realización de los casos de prueba se han seguido los criterios de validación y verificación del software del libro "Ingeniería de Software" de Ian Sommerville.

Hemos llevado a cabo las siguientes pruebas:

7.2 Pruebas punto a punto:

Antes de la implementación completa del sistema, hemos realizado inspecciones periódicas de los distintos módulos que componen la aplicación. El objetivo de estas pruebas es la detección de defectos previos a la prueba del sistema en entorno real. De esta forma hemos depurado determinados errores del código y otros que podían impedir el funcionamiento completo del sistema (bucles infinitos o conexiones erróneas a la base de datos) o llevarlo a un estado inestable (inserciones fallidas o con información equivocada en la base de datos). Para garantizar que se chequean todas las clases usamos:

7.2.1. Pruebas de Casos de Uso:

Son aquellas que prueban individualmente los casos de uso para asegurar que se cumple correctamente la funcionalidad. Para ello, hemos probado la secuencia de operaciones asociadas además de las consultas a ambas bases de datos.

De la misma forma, hemos realizado análisis de flujo, comprobando las entradas y salidas de funciones para garantizar la obtención de salidas deseadas. Sirven igualmente para detectar llamadas erróneas.

A continuación se detallan las pruebas más significativas de entre las realizadas:

7.2.1.1 Caso de Uso 1: ComprobarNota

Acción	Entrada	Acción Siguiente	Resultado Esperado
Solicitar Nota		Consulta a la Base de Datos de Moodle.	Información relativa a las notas si existe.
Conectar Base de Datos	Conexión Errónea	Envío de aviso.	Aviso generado y enviado.
	Ninguna Nota.		Exit.
	Nota Nueva.	Cifrar y almacenar en la base de datos de seguridad.	Información en claro y cifrada almacenada.
	Nota existente.	Comprueba el valor.	Determina si ha sido

			modificada.
Nota existente	Nota sin cambio.	Comprueba siguiente nota.	Siguiente nota.
Nota existente	Nota con cambio.	Busca Usuario con Permisos.	Información Usuario.
Usuario existente	En base de datos de seguridad.	Envío de aviso.	Aviso generado y enviado.
Usuario desconocido		Consulta a la Base de Datos de Moodle.	Información relativa a los usuarios.
Usuario Existente	Tras búsqueda en Moodle.	Envío de aviso.	Aviso generado y enviado.
Usuario desconocido.		Consulta a la Base de Datos de Moodle.	Información relativa al intruso.
Aviso al Administrador	Intruso detectado.	Envío de aviso.	Aviso generado y enviado.
Nota existente	Inconsistencia entre la nota cifrada y en claro.	Envío de aviso al Administrador.	Aviso de error grave generado y enviado
Cambio de estado en Flag	0 → 1 0 → 2	Acciones correspondientes en cada uno de los estados.	
Inserción en el Log	Nota conflictiva con sus parámetros asociados.	Almacena dicha información en la base de datos de seguridad.	Información almacenada.

7.2.1.2 Caso de Uso 2: EnviarRecordatorio

Acción	Entrada	Acción Siguiente	Resultado Esperado
Buscar Avisos sin atender	Flag = 1	Consulta a la Base de Datos de seguridad.	Información relativa a los avisos sin atender.
Conectar Base de Datos	Conexión Erronea	Envío de aviso.	Aviso generado y enviado.
	Ningún Aviso.		Exit.
Usuario existente	En base de datos de seguridad.	Envío de aviso.	Aviso generado y enviado.
Usuario desconocido.		Consulta a la Base de Datos de Moodle.	Información relativa al intruso.

Aviso al Administrador	Intruso detectado.	Envío de aviso.	Aviso generado y enviado.

7.2.1.3 Caso de Uso 3: ResolverConflicto

Acción	Entrada	Acción Siguiente	Resultado Esperado
Aviso generado		Muestra toda la información necesaria sobre el conflicto	Aviso completo y correcto
Resolver Inconsistencia	Nota antigua.	Cambio en el Flag de estado, inserción en el Log.	Flag de 1 → 0 Información almacenada.
Resolver Inconsistencia	Nota nueva.	Inserción en claro y cirada, inserción en el Log, cambio en el Flag.	Flag de 1 → 0 Información almacenada.
Resolver Inconsistencia	Detener aviso de incidencia grave.	Detener envío de correos de aviso para dicha incidencia, cambio en el Flag.	Flag de 2 → 3
Resolver Inconsistencia	Inconsistencia grave resulta.	Volver a comprobar dicha nota con el valor correcto.	Flag de 3 → 0 ó Flag de 2 → 0

7.2.2. Pruebas de Trayectoria:

Cubren todos los escenarios posibles de un caso de uso y todas y cada una de las bifurcaciones en cada toma de decisión por separado, garantizando que todo se ejecuta al menos una vez.

Hemos realizado estas pruebas, prestando especial atención a las funciones del sistema que hacen consultas a bases de datos y a las combinaciones de funciones que comparten parámetros entre ellas.

7.2.3. Pruebas de Caja Negra:

Estas pruebas nos informan si la salida es correcta para todas las entradas introducidas sin tener en cuenta el código. Hemos probado tanto las funciones de la aplicación como las consultas a las bases de datos, con datos correctos y completos y con datos insuficientes; por ejemplo, calificaciones sin actividad y actividades sin calificación.

7.2.4. Pruebas de Caja Blanca:

Se basan en la comprobación de la estructura e implementación del código. A parte de la comprobación de código que hemos realizado, hicimos un análisis de utilización de datos y variables garantizando que todas son usadas, están bien inicializadas, no son redundantes y su nombre no produce conflictos con otras variables de la misma aplicación.

Nuestro entorno de desarrollo, NetBeans, nos ayuda con este tipo de pruebas, avisándonos de posibles errores en el código y de las variables no inicializadas o no usadas más adelante.

Los dos siguientes tipos de prueba no se han podido realizar en entornos locales y probablemente no se obtendrán resultados significativos que permitan comprobar el funcionamiento del sistema, incluso en un entorno real, hasta que no se tengan suficientes calificaciones, usuarios y actividades, es decir, suficiente carga de trabajo.

7.3 Pruebas de esfuerzo:

Forzando incrementalmente el sistema hasta que deje de funcionar, ya sea con un comportamiento extraño, erróneo o que el sistema se ralentice exageradamente.

7.4 Pruebas estadísticas:

Las pruebas estadísticas contemplan los siguientes aspectos:

- Para distintas clases de entradas, con un conjunto de datos de prueba, se cuentan las caídas y se calcula la fiabilidad estadísticamente.
- Problemas: coste e incertidumbre de que esas entradas seleccionadas sean las adecuadas para calcular esa fiabilidad.
- Las pruebas se detienen cuando se alcanza la fiabilidad requerida.

8 CONCLUSIONES

Como primera conclusión de nuestro proyecto podemos decir que los objetivos han sido alcanzados con éxito.

Se ha diseñado e implementado una aplicación capaz de controlar las notas almacenadas en Moodle para cualquier tipo de titulación y asignatura, tratando de evitar que pase desapercibida para la persona responsable del curso, una modificación ilícita de las calificaciones, ya sea a través de la interfaz gráfica de la herramienta o directamente en el servidor que contiene la base de datos. Además, se han implementado medidas de seguridad extra en las calificaciones almacenadas, concretamente el cifrado HMAC (SHA-256 con clave).

La aplicación nos permite:

- Almacenar las notas finales de actividades y cursos de manera segura sin que el profesor tenga que comprobar por sí mismo que no ha habido una modificación ilegal de las notas almacenadas.
- Avisar al profesor, gestor o personal encargado, en caso de que se produzca un cambio en las notas, ya sea autorizado o no, permitiendo revelar los cambios no deseados, además de mantener un log de las incidencias detectadas en el sistema.
- El almacenamiento de las notas cifradas durante varios años para futuras reclamaciones tal y como estipula la ley.
- Detectar modificaciones ilegales en la base de datos que contiene las notas cifradas.

De la misma manera se ha cumplido con los objetivos complementarios: creación de un manual de usuario de requisitos previos en Moodle, entre los que se incluyen, la creación de usuarios (globales al sistema, profesores y alumnos asignados a cursos concretos), cursos y notas, junto con la explicación detallada de las tablas utilizadas en el diseño de los algoritmos de acceso a la plataforma para obtener la información necesaria.

En conclusión, se puede dar por finalizado el desarrollo del trabajo fin de grado expuesto en esta documentación. La realización del presente proyecto ha constituido una experiencia muy positiva que ha complementado la formación recibida a lo largo de la carrera aplicada a un contexto cotidiano y práctico.

Para desarrollar este proyecto ha sido necesario el estudio de herramientas, lenguajes de programación, seguridad de la información, tablas y relaciones de la base de datos de Moodle y algoritmos que nos han aportado nuevos conocimientos y ampliado los ya existentes.

Creemos que la aplicación será útil y podrá ayudar a los responsables de los cursos de Moodle a mantener la seguridad en sus calificaciones. Por lo que, tras realizar pruebas durante

unos meses en entorno real, lo más probable es que se pueda compartir la aplicación implementada a lo largo de este proyecto, junto con la parte de documentación relevante para su comprensión en la página oficial [www.moodle.org]. PHP es el lenguaje utilizado en el desarrollo de Moodle, por ello fue elegido como lenguaje de implementación; de esta manera, cualquier desarrollador de la plataforma podrá comprender el código e incluso aportar ideas para la mejora del mismo.

9 LÍNEAS FUTURAS DE TRABAJO

La aplicación práctica del trabajo de fin de grado radica en el aviso a los profesores cuando se detecta un cambio en las calificaciones. Nuestro objetivo principal era elaborar un sistema capaz de realizar todo ese trabajo de forma automática y simple, con una breve aportación del profesor que resuelva el conflicto. Tras la implementación y prueba de la aplicación en entorno real hemos encontrado las siguientes líneas futuras de ampliación:

- Mejora de la interfaz gráfica del sistema.
    - Los avisos, enviados por correo electrónico, podrán llevar adjunta la firma identificativa de la universidad, empresa o institución en la que se instale la aplicación.
    - Asimismo, los mensajes de retroalimentación de los profesores podrán adaptarse a la interfaz gráfica de la página web o la plataforma Moodle de la que obtengan información.
- Actualización automática de la base de datos de Moodle, al igual que se hace con la que contiene la copia de seguridad cifrada de las calificaciones. Debido a la complejidad de dicha base ha sido imposible la actualización automática, dejando como atribución del personal responsable la actualización de la información de la misma.
- Traducir la documentación asociada al proyecto al inglés, para poder compartirla con el resto de desarrolladores Moodle en la página Web. El código ha sido implementado en inglés, con las variables en dicho idioma, para facilitar esta tarea.
- Adaptación de la consulta de solicitud de calificaciones al resto de los criterios de finalización de cursos, enunciados en el Capítulo 2 de esta memoria, además del de fecha de finalización ya implementado.

10 SEGUIMIENTO DEL PROYECTO

10.1 Historial de revisiones de la documentación:

Fecha	Versión	Descripción
25/02/12	<0.1>	Especificación de Objetivos y Requisitos.
12/03/12	<0.2>	Documento de Análisis.
24/04/12	<0.3>	Descripción Base de Datos de Moodle.
27/04/12	<0.4>	Introducción a la Seguridad de la Información.
07/05/12	<0.5>	Documento de Diseño.
14/06/12	<0.6>	Documento de Implementación y Pruebas.
03/07/12	<0.7>	Anexos.
09/07/12	<1.0>	Revisión Global de la Documentación y conclusiones.

10.2 Calendario

En la siguiente tabla comparamos las sucesivas estimaciones temporales sobre la duración del proyecto y la duración real aproximada en horas.

	Estimación Inicial	Realidad
Objetivos y Restricciones	17 horas	16 horas
Análisis	30 horas	36 horas
Diseño	40 horas	37 horas
Estudio de la Plataforma y Seguridad	150 horas	170 horas
Documentación uso de la Plataforma	33 horas	28 horas
Implementación	125 horas	129 horas
Pruebas	35 horas	40 horas
Maquetación	20 horas	19 horas
Total	450 horas	475 horas

11 ANEXOS

11.1 ANEXO I: Contenido del CD-ROM

El CD-ROM que se adjunta con la memoria contiene los siguientes directorios:

- Código Fuente: en esta carpeta se incluye la aplicación completa, tanto de la comprobación de las calificaciones, como la generación de avisos de recordatorio, obtención de datos de Profesores y Gestores del sistema y la resolución de conflictos de la base de datos de seguridad.
- Memoria: En este directorio se encuentra la versión de esta memoria en formato PDF.
- Diagramas: Localización de todos los diagramas realizados a lo largo del proceso de desarrollo en un fichero de StarUML. (.uml)

11.2 ANEXO II: Manual Básico de Moodle.

11.2.1. Introducción.

En este apartado detallaremos brevemente los requisitos previos en Moodle para hacer funcionar nuestra aplicación de forma óptima. Para instalar la plataforma en el sistema operativo Windows o en Linux, ya sea en un servidor o en un equipo local se recomienda consultar los manuales en línea, la documentación en español se encuentra en [http://docs.moodle.org/all/es/Instalaci%C3%B3n_de_moodle].

Para cualquier tipo de información extra sobre documentación se puede consultar [http://moodle.org/support/].

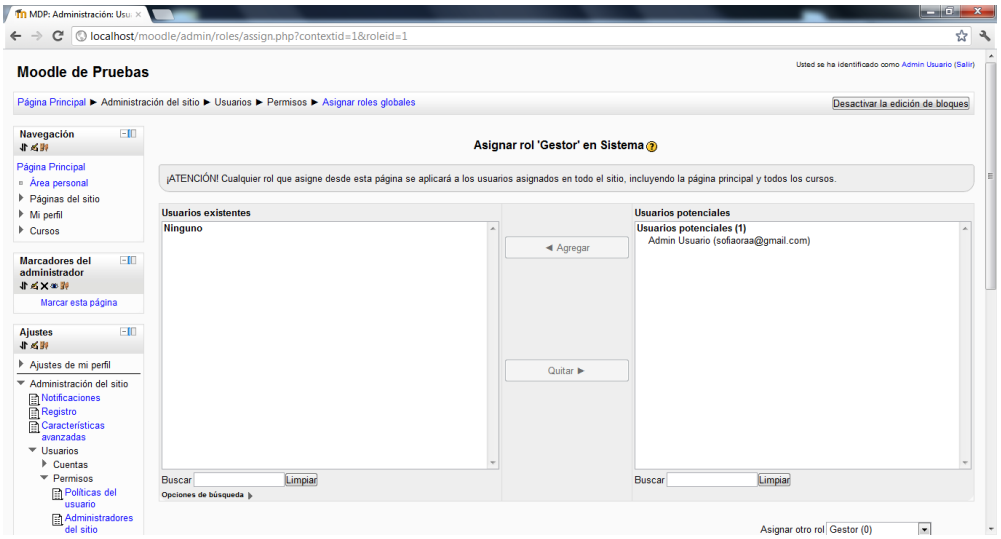
11.2.2. Creación de Usuarios Privilegiados

Todas las operaciones que puede realizar un administrador se encuentran recogidas en el panel de administración [http://docs.moodle.org/all/es/Bloque_de_administración_del_sitio]. Entre ellas se encuentra la creación de usuarios.

Para crear manualmente un usuario el Administrador debe hacer clic sobre la opción “Usuarios” del menú “Administración del Sitio”, esto desplegará la opción “Cuentas” y al elegirla entre otras opciones se encontrará la de “Agregar Usuario” que nos permite realizar dicha actividad. Tras rellenar los campos requeridos en el formulario que aparece y presionar el botón “Actualizar información personal” se creará la cuenta de usuario correspondiente.

Panel de Administración del Sitio

Todos los administradores, así como los usuarios que deban tener permisos sobre todo el sistema, deberán obligatoriamente, para no ser tratados como intrusos, estar asociados al rol Gestor. Para ello, se hace clic en "Permisos" que desplegará la opción "Asignar roles globales". Esta opción muestra la pantalla "Asignar roles en sistema" en la que podremos elegir asignar usuarios a los roles "Gestor" o "Creador del Curso". Haciendo clic en "Gestor" podremos seleccionar los usuarios que tendrán esos permisos.

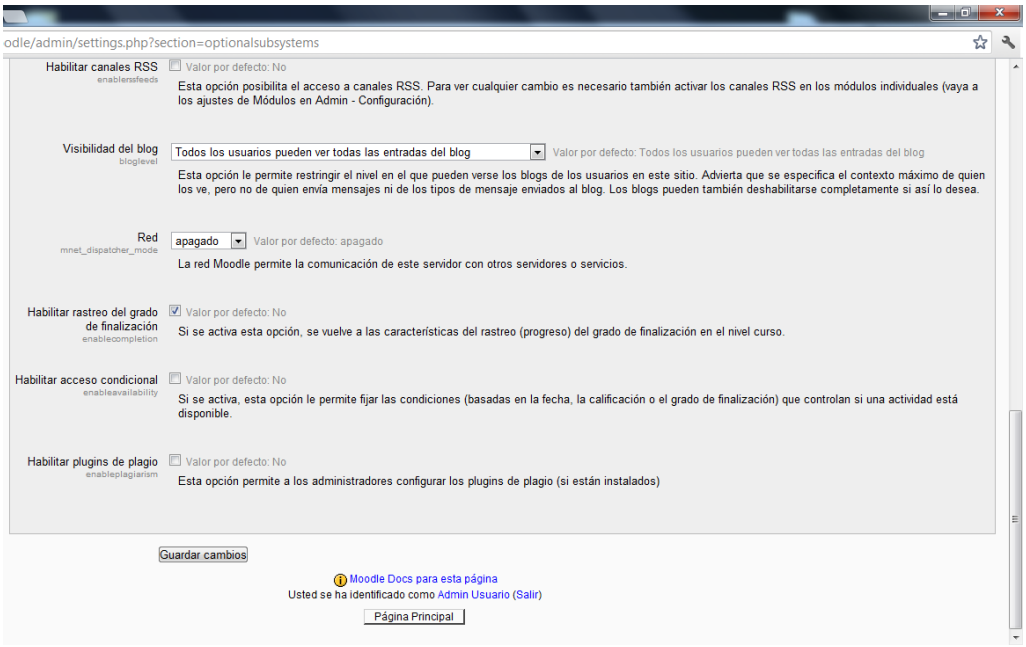


Asignar rol "Gestor" en Sistema

11.2.3. Activar Rastreo de Finalización

Una de las condiciones necesarias para la optimización del algoritmo de búsqueda de calificaciones es la selección de una fecha de fin de curso. Para poder agregar una fecha de fin a un curso, previamente el Administrador del sistema debe activar el "Rastreo de Finalización" en todo el sistema.

El administrador debe hacer clic sobre la opción "Características Avanzadas" del menú "Administración del Sitio". Una vez se desplieguen todas las características posibles se debe buscar aquella en la que ponga "Habilitar rastreo del grado de finalización" y habilitarla haciendo clic sobre el checkbox correspondiente. Finalmente, presionar en "Guardar cambios" para que el cambio sea permanente.



Rastreo de Finalización en todo el Sistema

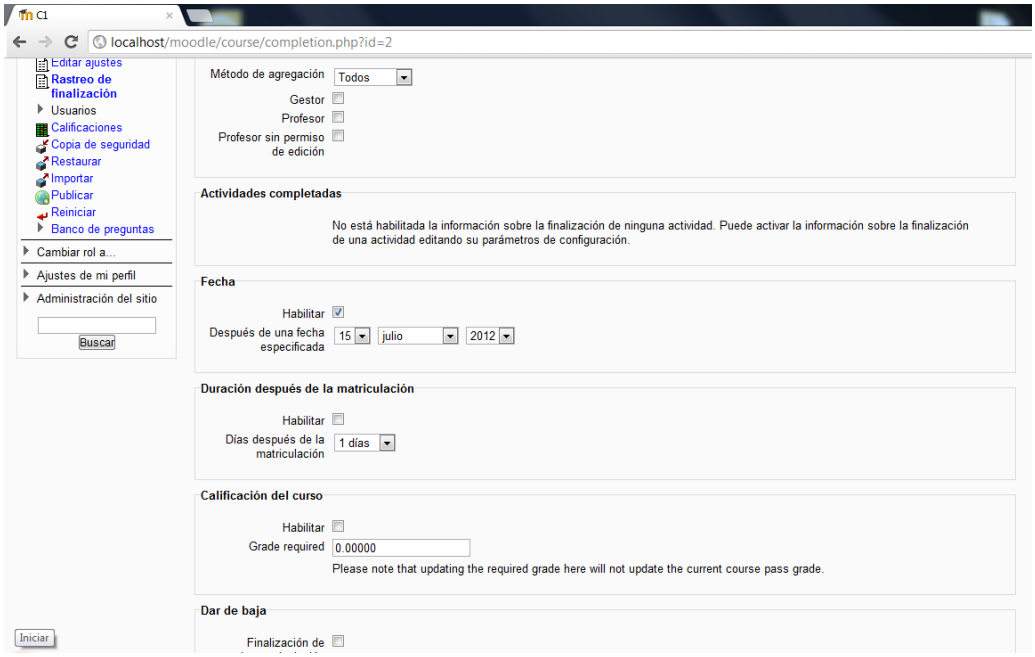
Una vez activado el rastreo de finalización del sitio será necesario seguir los pasos indicados en el apartado “11.2.4. Creación de Cursos” para habilitar el rastreo de finalización en los mismos.

11.2.4. Creación de Cursos

Los usuarios que tienen permisos para crear un curso son: profesor con condición de creador de cursos, administrador del sistema y gestores. Para crear un curso se debe hacer clic en la opción “Cursos” del menú “Administración del sitio” en la página principal de la plataforma. Tras ello se selecciona "Agregar/editar cursos" y una vez aparezca la opción "Agregar un nuevo curso". Bastará con rellenar los diferentes campos solicitados en el formulario desplegado y finalmente, bastará con pulsar en “Guardar cambios”, de esta manera aparecerá el nuevo curso creado en la página del sitio.

Es necesario para la optimización del algoritmo, que en el bloque “Progreso del estudiante”, de la pantalla “Crear Curso”, se elija la opción “Activado, control por medio de los ajustes de finalización y de actividad” en “Rastreo de finalización”, además de activar la opción, haciendo clic en el checkbox asociado: “El rastreo de la finalización comienza en la matriculación”.

Tras crear el curso, una vez se muestre la página principal del mismo, se deberá seleccionar la opción "Rastreo de finalización", disponible en el menú "Administración del curso" en el panel a la izquierda, lo cual nos mostrará la pantalla "Editar ajustes de grado de finalización del curso". En esta pantalla se debe habilitar la opción de rastreo del bloque "Fecha" y señalar la fecha de finalización del curso en el campo correspondiente "Después de una fecha especificada". Finalmente hacer clic en "Guardas cambios".



Rastreo de Finalización del Curso

11.2.5. Creación de Actividades

Para agregar o editar actividades se debe activar la edición del curso, para ello, basta con hacer clic en el botón "Activar edición" situado en la parte superior derecha de la página principal.

Existen varios tipos de actividades de aprendizaje interactivo que se pueden incluir en un curso. Los trabajos del alumnado pueden ser enviados y calificados por los profesores mediante los módulos de Tareas o Talleres. Se puede calificar de forma automática mediante los Cuestionarios o añadir ejercicios "Hot Potatoes". Las comunicaciones se pueden realizar en los Chat y en los Foros para debates y las Consultas para obtener sus opciones preferidas. Los alumnos pueden trabajar de forma colaborativa mediante los Wikis. También pueden utilizar los Blogs.

De la misma manera, el contenido se puede presentar y gestionar usando las actividades de Lecciones y SCORM. Las palabras claves del curso se pueden agregar en los Glosarios y, opcionalmente, también podrán hacerlo sus estudiantes. Las Encuestas y las Bases de Datos son actividades de gran ayuda en cualquier curso. Por último, se pueden añadir módulos no estándar que no forman parte de la versión oficial de Moodle.

11.3 ANEXO III: Ventajas de la plataforma Moodle

A continuación, se citan varias ventajas que tiene la utilización de Moodle como plataforma para la formación.

- Código abierto y gratuito: de esta manera se obtiene software a coste cero y con la posibilidad de poder adaptarlo a las necesidades de cualquier empresa.
- Comunidad de usuarios: Gracias a la extensa red de usuarios de Moodle, tanto a nivel de desarrollador, administrador, como usuario común, nuevas funcionalidades se crean continuamente, existe un gran banco de testeo, una gran comunidad de desarrollo, gente a la que consultar y con la que compartir conocimiento…
- Esta comunidad no permanece estática, de hecho la base de usuarios registrados incluye más 58 millones, distribuidos en más de 67.000 sitios validados en todo el mundo y con la plataforma traducida a 70 idiomas en 217 países.
- Alta disponibilidad: El LMS debe ser lo suficientemente robusto como para satisfacer las diversas necesidades de miles de estudiantes, administradores, creadores de contenidos y profesores simultáneamente. Moodle presenta una interfaz basada en WEB de alta disponibilidad, permitiendo a los aprendices, tutores y administradores iniciar sesión de manera permanente y ejecutar sus tareas diarias.
- Escalabilidad: La infraestructura debe poder ampliarse o escalar para resolver el futuro crecimiento, tanto en términos de volumen de contenidos educativos como del número de estudiantes. Existe un acuerdo en la comunidad Moodle respecto a que la mejor opción es un servidor web basado en Linux que ejecute Apache, junto con PHP y un acelerador PHP; por otra parte, también hay acuerdo en que el servidor web y el servidor de bases de datos deberían residir en máquinas separadas. Moodle facilita responder a futuras demandas de alumnos y cursos, adaptando las tecnologías bajo las que se ejecuta. Esto sería posible incluso en un entorno vivo, a fin de mejorar el servicio sin interrupciones importantes.
- Facilidad de uso: Apoyar un conjunto de servicios automatizados y personalizados, tales como aprender a ritmo individual y perspectivas específicas de aprendizaje, el acceso, la entrega y la presentación de materiales deben ser fáciles de utilizar, como navegar por la Web. Su interfaz es muy intuitivo, facilita la creación de cursos y tareas, incluso para personas fuera del sector de las TIC.
- Interoperabilidad: Para admitir contenido de diferentes fuentes, y soluciones de equipos de cómputo o programas de diversos proveedores, el LMS debería intercambiar información utilizando estándares abiertos de la industria para implementaciones WEB.

- Estabilidad: La infraestructura del LMS puede soportar de manera confiable y efectiva una implementación productiva a gran escala las 24 horas del día, los 7 días de la semana. Este tema está relacionado con lo comentado en los apartados de Alta disponibilidad y Escalabilidad.
- Seguridad: Al igual que sucede con cualquier solución colaborativa, el LMS puede limitar y controlar selectivamente el acceso de la diversa comunidad de usuarios a los contenidos en línea, recursos y funciones del servidor tanto interna como externamente.

12 BIBLIOGRAFÍA

A continuación mostramos la bibliografía, así como documentación técnica o artículos relacionados que nos han sido de cierta utilidad en la realización de este documento.

[1] Areitio, J. (2008), *Seguridad de la Información. Redes, informática y sistemas de información*, Paraninfo.

[2] Arlow, J. (2006), *UML 2*, Anaya Multimedia, Madrid.

[3] Bruegge, B. & Dutoit, A. (2000), Object-Oriented Software Engineering. Conquering Complex and Changing Systemes, Prentice Hall.

[4] Larman, C. (2005), *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and the Unified Process*, Prentice-Hall.

[5] Sommerville, I. (2002), *Ingeniería de Software*, 6ª ed. Addison- Wesley.

[6] Steven, P. & Pooley, R. (2007), *Utilización de UML en Ingeniería del Software con objetos y componentes*, 2ªed. Addison-Wesley.

[7] Página Oficial de Moodle.
Disponible en:
http://moodle.org/
[Última consulta: 15 Julio 2012].

[8] The Keyed-Hash Message Authentication Code.
Disponible en:
http://csrc.nist.gov/publications/fips/fips198/fips-198a.pdf
[Última consulta: 15 Julio 2012]

[9] Unified Process for EDUcation [UPEDU].
Disponible en:
http://www.upedu.org/
[Última consulta: 15 Julio 2012].

[10] Universidad de Valladolid.
Disponible en:
http://www.uva.es/
[Última consulta: 15 Julio 2012]

[11] Real Academia Española [RAE]
Disponible en:
http://www.rae.es
[Última consulta: 15 Julio 2012]

[12] Wikipedia, La enciclopedia Libre
Disponible en:
http://es.wikipedia.org
[Última consulta: 15 Julio 2012]

[13] MySQL: The world's most popular open source database
Disponible en:
http://www.mysql.com/
[Última consulta: 15 Julio 2012]

[14] NetBeans Download (Windows y Linux)
Disponible en:
http://netbeans.org/downloads/index.html
[Última consulta: 15 Julio 2012]

[15] StarUML Download
Disponible en:
http://staruml.sourceforge.net/en/download.php
[Última consulta: 15 Julio 2012]

[16] Java SE Development Kit
Disponible en:
http://www.oracle.com/technetwork/java/javase/downloads/java-se-jdk-7-download-432154.html
[Última consulta: 15 Julio 2012]

[17] Xampp Download (Windows y Linux)
Disponible en:
http://www.apachefriends.org/es/xampp.html
[Última consulta: 15 Julio 2012]