

**EPS**

Escola Politècnica
Superior

Projecte/Treball Fi de Carrera

Estudi: Enginyeria Informàtica. Pla 1997

Títol: Mòdul per gestionar el model de usuari i permisos, el sistema de arxius i el de notícies en la plataforma CoSpace

Document: Memòria

Alumne: Rachid Mahmoudi

Director/Tutor: Silvia Margarita, Baldiris Navarro i Teodor Maria Jové Lagunas

Departament: Arquitectura i Tecnologia de Computadors

Àrea: Arquitectura y Tecnología de Computadores

Convocatòria (mes/any): Septiembre 2014

Agradecimientos

En primer lugar quiero agradecer a mis tutores, Silvia Baldiris y Teodor Jové, la confianza en mí desde el inicio de este trabajo. Me han ayudado muchísimo a lo largo de todo este recorrido con sus ideas, comentarios y opiniones. Y en el Departamento de Arquitectura y Tecnología de Computadores a mi director Ramon Fabregat Gesa a permitirme realizarlo. Y como no, no puedo olvidarme también de los momentos de risas tan necesarios. ¡Gracias por todo Silvia y Teodor Jové!

Gracias también a mis compañeros Jorge Luis Bacca Acosta y Cecilia Avila Garzon. Desde que escogí este Proyecto Fin de Carrera se convirtieron en mis segundos tutores. Siempre han estado disponible para cualquier cosa que he necesitado.

No puedo olvidarme del laboratorio donde se ha desarrollado este PFC: BCDS. En él he conocido a grandes compañeros y amigos, gracias Hendrys Fabián Tobar. Muchas Gracias a todos ;).

No, no me he olvidado de una gran persona Llorenç Burgas Nadal. Este trabajo es tuyo también pues me habéis escuchado y animado en infinidad de ocasiones a lo largo de todos estos años.

Desde que empecé la carrera tuve la suerte de rodearme de buena gente que con el paso del tiempo se convirtieron en grandes amigos. Con ellos he pasado muchísimos momentos buenos tanto fuera (fiestas, cumpleaños, viajes...) como dentro de la EPS (clases, prácticas, estudios, exámenes...) imposibles de olvidar. Ahora empieza una nueva etapa y aunque de otra forma, sé que cuento con vosotros del mismo modo.

Por último, pero no menos importante, le agradezco a mi familia su apoyo incondicional. Especialmente a mis padres, los cuales han dado siempre todo por mí, en cada paso que he dado y en éste especialmente, con el único objetivo de verme feliz. Sin vosotros nada sería posible.

Contenido

Capítulo 1:	Introducción.....	1
1.1-	Contexto.....	1
1.2-	Motivaciones personales.....	2
1.3-	Propósitos y objetivos.....	2
1.4-	Tecnologías vinculadas al proyecto	3
1.5-	Estructuración de la documentación	3
Capítulo 2:	Estudio de la viabilidad	5
2.1-	Herramientas de desarrollo.....	5
2.2-	Recursos Humanos	5
2.3-	Evaluación previa de costes y medios.....	5
2.2.1-	Estudio de viabilidad técnica.....	5
2.2.2-	Estudio de viabilidad económica.....	6
2.2.2.1-	Costes de recursos humanos	6
Capítulo 3:	Metodología	9
3.1-	Metodología.....	9
Capítulo 4:	Planificación	13
4.1-	Tiempo estimado	13
4.2-	Resultados estimados	15
Capítulo 5:	Marco de trabajo y conceptos previos.....	17
Capítulo 6:	Requisitos del sistema	19
6.1-	Necesidades tecnológicas	19
6.2-	Descripción del sistema	19
6.3-	Técnicas utilizadas para la identificación de requerimientos	20
6.4-	Requerimientos funcionales.....	20
6.4.	Requisitos no funcionales	21
6.5-	Restricciones.....	22
6.5.1-	Políticas.....	22
6.5.2-	Interfaces con otras aplicaciones.....	23
6.5.3-	Funciones de control.....	23
6.5.4-	Lenguaje de programación.....	23
6.5.5-	Protocolos de comunicación	23
6.5.6-	Requisitos de Fiabilidad.....	23
6.5.7-	Consideraciones acerca de la seguridad	23
Capítulo 7:	Estudios y decisiones.....	25
7.1-	Justificación del entorno de desarrollo	25

Capítulo 8:	Análisis y diseño del sistema	33
8.1-	Diseño general	33
8.2-	Diagrama de casos de uso	33
8.2.1-	Identificación de los actores	33
8.2.1.1-	Casos de uso del usuario no registrado	34
8.2.1.2-	Casos de uso del Colaborador	35
8.2.1.3-	Casos de uso del colaborador administrador de foros	35
8.2.1.4-	Casos de uso del profesor	36
8.2.1.5-	Casos de uso del Administrador	37
8.2.2-	Diagrama de caso de uso general	38
8.3-	Diagrama de actividad	40
8.4-	Diagrama de Entidad Relación	44
8.5-	Diagrama de clases	44
8.6-	Arquitectura Física	46
8.6.1-	Servidor Web	46
8.7-	Modelo de programación	46
8.7-1.	Capa de presentación	46
8.7-2.	Capa de Negocio	47
8.7-3.	Capa de Persistencia	47
Capítulo 9:	Implementación y pruebas	49
9.1-	Clases	49
9.1.1-	Modelo usuario	49
9.1.2-	Noticia	51
9.1.3-	Conjunto de permisos	51
9.1.4-	EsAdmin	55
9.1.5-	Archivo	56
9.1.6-	Etiqueta	57
9.1.7-	Carpeta	57
9.2-	Diseño de la base de datos	58
9.3-	Pruebas de funcionamiento	59
9.4-	Acceso a la aplicación	64
9.5-	Problemas encontrados durante la realización del proyecto	64
Capítulo 10:	Resultados	65
10.1-	Modelo de usuario	65
10.2-	Sistema de gestión de archivos	69
10.3-	Noticias	69

10.4-	Permisos.....	70
Capítulo 11:	Conclusiones.....	73
Capítulo 12:	Trabajo futuro.....	75
Capítulo 13:	Bibliografía.....	77
Capítulo 14:	Anexos.....	79
Capítulo 15:	Manual de usuario	81
15.1-	Registro e inicio de sesión	82
15.2-	Modelo de usuario.....	85
15.2-1.	Escenario de desempeño	87
15.2-2.	Perfil general.....	89
15.3-	Noticias	91
15.4-	Archivos	94
15.5-	Gestión de permisos.....	98

Capítulo 1: Introducción

CoSpace es una plataforma web diseñada para proporcionar un espacio virtual de interacción y colaboración entre formadores en comunidades virtuales. Se originó como resultado de las necesidades adicionales que surgieron cuando los profesores que trabajan en temas de educación en contextos de diversidad (en las áreas de lenguaje, de matemáticas y de ciencias) requirieron de una herramienta para favorecer o apoyar la interacción y la colaboración entre ellos para compartir, ideas, experiencias, objetos virtuales de aprendizaje, entre otros elementos de información, desde una perspectiva actual del uso de la tecnología desde una visión más cercana al usuario final. Este paradigma promueve la idea que las aplicaciones deben estar concebidas para usuarios poco expertos, permitiendo a éstos un mayor control explícito en la modificación y creación de artefactos software de forma autosuficiente, y sin presuponer conocimientos en programación tradicionalmente reservados a expertos en computación, y teniendo en cuenta los principios de usabilidad y accesibilidad.

El alcance de este proyecto está definido por la necesidad de gestión por parte de los profesores, de la información de su perfil personal, permitiendo a los usuarios la creación, edición, consulta y eliminación de tal información, así mismo de la necesidad de compartir archivos, publicar noticias entre comunidades de práctica¹, y finalmente de la necesidad de gestionar los permisos de usuario para que estos gestionen módulos de la plataforma.

En esta memoria se detallan los motivos por los que se decidió realizar el presente proyecto y los objetivos que se pretenden alcanzar con la realización del mismo. Se continúa comentando la estructura del documento y se finaliza mostrando un listado de acrónimos, que aparecerán a lo largo del texto con sus correspondientes términos y definiciones que se consideran necesarios para la correcta comprensión del contenido del documento.

1.1- Contexto

Este proyecto nace como una de las partes de proyecto europeo ALTER-NATIVA. ALTER-NATIVA fue un proyecto desarrollado por doce universidades, cuatro europeas y 8 latinoamericanas, y que fue coordinado técnicamente por el grupo de investigación BCDS² situado en el parque científico de la universidad de Girona.

¹ Las **comunidades de práctica** son grupos sociales constituidos con el fin de desarrollar un conocimiento especializado, compartiendo aprendizajes basados en la reflexión compartida sobre experiencias prácticas. Etienne Wenger ha estudiado las Comunidades de práctica y las ha definido como un “grupo de personas que comparten un interés, un conjunto de problemas, o una pasión sobre un tema, y quienes profundizan su conocimiento y experiencia en el área a través de una interacción continua que fortalece sus relaciones”

² El grupo **BCDS** es activo en dos áreas principales de investigación: Control de Red de Banda Ancha y de Gestión y de Sistemas Adaptativos hipermedia. El grupo ha realizado distintos proyectos competitivos con otras universidades y empresas como proveedores de servicios de Internet e instituciones de salud pública. Estos proyectos, junto con la investigación básica realizada en el grupo, han dado lugar a muchas publicaciones y

Para el desarrollo de la aplicación se han realizado diversas reuniones con los directores de proyecto y de las universidades, y se ha estudiado cómo se estaban realizando las tareas para después poder informatizarlas adecuadamente. La mayoría de las decisiones de interfaz y funcionales de la aplicación fueron tomadas junto con los directores, que además ha proporcionado toda la documentación necesaria sobre la dinámica del proyecto.

1.2- Motivaciones personales

Uno de los motivos personales para el desarrollo de este proyecto fue la oportunidad de conocer los lenguajes de programación de páginas web en los que está desarrollado CoSpace, ya que muchas de las ofertas de trabajo, lo piden, y en la carrera sólo hemos hecho una asignatura en quinto que se llama "Alternativas de diseño de aplicaciones". Desde el principio busqué un proyecto relacionado con esta temática, porque quería profundizar más, y con un proyecto de este tipo podía profundizar en esta rama de la informática.

1.3- Propósitos y objetivos

Los objetivos de este PFC son:

- 1- Definir un Modelo de usuario de los individuos que interactúan en CoSpace, formado por un perfil general, un perfil especializado en educación, y un perfil de accesibilidad a tecnología de interacción y comunicación.
- 2- Diseñar el sistema de gestión de archivos de CoSpace que proporcione áreas de almacenamiento compartidas con control de versiones y acceso a los materiales, grupos y subgrupos para el desarrollo de las tareas asociadas a las comunicaciones dentro de la plataforma y que contemple la política de derechos del sistema.
- 3- Diseñar un sistema de notificaciones para divulgar información de interés para los miembros de las comunidades dentro de la plataforma que tenga en cuenta los permisos de cada comunidad.
- 4- Diseñar los mecanismos que permitan adaptar y modificar el modelo de usuario, los permisos de cada dominio y asociar usuarios a dichos dominios.
- 5- Introducir la gestión de permisos/privilegios en el modelo de usuario para que pueda ser utilizado en el sistema de gestión de archivos y en el sistema de noticias.
- 6- Diseñar la gestión de permisos/privilegios en la plataforma, para que pueda ser utilizada en el sistema de gestión de archivos, sistema de noticias y foros.

1.4- Tecnologías vinculadas al proyecto

Este proyecto se desarrollará con las tecnologías JavaScript, Node.js, Redis, Jade y MongoDB.

Podemos organizar el trabajo siguiendo una estructura modular, lo que implica:

- Documentar el código existente de la web CoSpace
- Aprender y manejar los lenguajes de programación de páginas web
- Diseñar e implementar el modelo de usuario
- Test del funcionamiento del modelo de usuario
- Diseñar e implementar sistema de gestión de archivos (SGA)
- Realizar el test del funcionamiento del SGA
- Diseñar e implementar sistema de noticias
- Realizar el test del funcionamiento de sistema de noticias
- Diseñar e implementar sistema de privilegios para el modelo de usuario, SGA y sistema de noticias
- Realizar el test del funcionamiento de Sistema de privilegios
- Documentar el trabajo realizado

1.5- Estructuración de la documentación

Esta memoria está estructurada en catorce capítulos fundamentales que recogen toda la documentación y explicaciones necesarias para la completa comprensión del proyecto.

1. Introducción, contexto, motivación, propósitos y objetivos del proyecto. En este capítulo se explicará el porqué del desarrollo de este proyecto, cuáles son los objetivos propuestos y como se ha organizado el desarrollo.
2. Estudio de la viabilidad. En este capítulo se justifican los parámetros que hacen posible el desarrollo del proyecto.
3. Metodología. Este capítulo contiene la explicación del proceso metodológico seguido para el desarrollo del proyecto.
4. Planificación. En esta etapa se define la estrategia seguida para alcanzar los objetivos planteados.
5. Marco de trabajo y conceptos previos. En este capítulo se describen los diversos aspectos relacionados con el desarrollo general del proyecto, que ayudarán a entender mejor los siguientes capítulos. También se tratarán las principales acciones desarrolladas durante las primeras etapas de la realización del proyecto. Se incluirán los pasos de estudio y aprendizaje de conceptos que se hayan utilizado para el desarrollo.
6. Requisitos del sistema. En este capítulo se definen los requerimientos del software, los cuales recogen, a grandes rasgos, los objetivos de la aplicación junto con las funcionalidades que se quieren obtener. Este documento permite entender los elementos que rodean al sistema informático que se intenta construir.

7. Estudios y decisiones. Esta sección contiene una descripción de las herramientas utilizadas, con sus características y el uso que se les ha dado.
8. Análisis y diseño del sistema. Este apartado proporciona una comprensión precisa de las necesidades del sistema, es decir, se encarga del análisis del problema a resolver. Usando la ingeniería de software, esta sección traduce los requerimientos descritos en capítulos anteriores a un lenguaje formal. Esta parte del diseño permite aumentar el nivel de especificación, y realizar un esquema de implementación del sistema mediante diversas herramientas de programación orientada a objetos, explicando las clases y los métodos que forman el proyecto.
9. Implementación. En este capítulo se dan a conocer cómo se ha implementado la web, las clases y los métodos que resultan más significativos para la comprensión del funcionamiento de la web.
10. Pruebas. En este capítulo se muestran pruebas de ejecución de la web.
11. Conclusiones. En este apartado se expondrán las conclusiones obtenidas una vez finalizado el proyecto.
12. Trabajo futuro. En esta sección se expone todo lo que se podría mejorar en la web, o ampliarla de forma interesante.
13. Bibliografía. Este capítulo contiene las referencias utilizadas para el desarrollo del proyecto.
14. Anexos. Esta sección contiene las definiciones de los tecnicismos más frecuentemente utilizados, así como explicaciones y experimentos no indispensables para la comprensión global del proyecto.
15. Manual de usuario. Esta sección incluye las especificaciones del funcionamiento del producto.

Capítulo 2: Estudio de la viabilidad

En este capítulo describimos los diferentes tipos de viabilidad que se analizaron antes de iniciar el desarrollo de este PFC.

2.1- Herramientas de desarrollo

Los lenguajes de programación utilizados han sido los lenguajes HTML, JavaScript, nodeJs, jade y MongoDB. Estas herramientas son open sources, lo cual facilitó en gran medida el desarrollo del proyecto.

Este PFC se ha desarrollado bajo el entorno Unix, utilizando herramientas de desarrollo multiplataforma. Además estas herramientas están disponibles en plataformas Windows y Mac OS X.

2.2- Recursos Humanos

En este proyecto ha sido necesaria la participación de un desarrollador de software capaz de proyectar las ideas iniciales en código. También es necesario un Analista de software. Ambos roles, tanto el de programador como el de analista de software los ha asumido el autor del proyecto.

Otros roles importantes han sido los de dirección, que han orientado el desarrollo de la aplicación y también los usuarios finales que han aportado su visión de la funcionalidad requerida para la plataforma CoSpace.

2.3- Evaluación previa de costes y medios

Son dos estudios de viabilidad necesarios para el desarrollo del sistema: la viabilidad técnica, y la viabilidad económica.

2.2.1- Estudio de viabilidad técnica

El estudio de la viabilidad tecnológica comienza con una definición técnica del sistema propuesto, dando respuesta a las siguientes preguntas: ¿Qué tecnología se necesitarán para lograr la funcionalidad de la aplicación? ¿Qué nuevos materiales, métodos o procesos se requieren? Como afectarán al coste estos elementos de tecnología?

Por suerte, ya se disponen de los dispositivos hardware necesario para poder realizar el proyecto, es decir, un ordenador capaz de realizar toda la programación, la interpretación del código y la visualización de los resultados.

2.2.2- Estudio de viabilidad económica

La valoración del análisis económico se separará en dos partes: costes de recursos humanos y costes de hardware.

2.2.2.1- Costes de recursos humanos

Para calcular los costes de los recursos humanos se ha asociado un perfil o rol de trabajador por cada uno de las tareas (el coste de director multiplicado por el total de directores, en este caso son 2)

Coste analista: 20 € / hora

Coste programador: 15 € / hora

Coste Director: 50 € / hora

Tarea	Perfil	Horas	Coste
Documentar Código	Programador	80	1200
Estudio de lenguajes de programación webs	Programador	80	1200
Diseño Modelo de Usuario	Analista	15	300
Implementación código	Programador	100	1500
Implementación interface	Programador	40	600
Avaluación Directores	Director	20	2000
Diseño Sistema de gestión de archivos	Analista	15	300
Implementación de gestión de archivos	Programador	100	1500
Implementación interface	Programador	40	600
Avaluación Directores	Director	20	2000
Diseño de sistema de noticias	Analista	4	80
Implementación de sistema de noticias	Programador	40	600
Implementación interface	Programador	15	225
Avaluación Directores	Director	12	1200
Diseño de sistema de privilegios/permisos	Analista	20	400
Implementación de sistema de privilegios	Programador	100	1500
Implementación interface	Programador	40	600
Avaluación Directores	Director	24	2400
Pruebas y optimización	Programador	20	300
Memoria	Analista	100	2000
Total		809	12905

2.2.2.2- Costes de Hardware

No he tenido ningún gasto por hardware, ya que desde el principio tenía todos los componentes necesarios, y en el transcurso del proyecto no se ha necesitado ningún hardware nuevo. Aun así, se ha contabilizado la amortización del hardware:

$$\text{Amortización} = \frac{\text{Coste hardware}}{\text{Tiempo de amortización}} * \text{Tiempo de proyecto}$$

Se ha definido que el tiempo de amortización son 3 años, el tiempo del proyecto 2 año y el coste del hardware, un computador portátil, es de 499 €. Esto da como resultado un gasto de amortización de 332 €.

Al no tener que pagar licencia por software, y teniendo en cuenta la amortización, el coste total del proyecto, incluyendo los costes de programación, ha sido de 20.837€.

Capítulo 3: Metodología

3.1- Metodología

Para la realización de este proyecto se ha seguido una metodología de trabajo como si indica en los pasos siguientes:

1. Documentar el código inicial de la aplicación.
2. Aprender el lenguaje de programación y las herramientas escogidas.
3. Estructurar el trabajo en partes según las funciones que se deban realizar.
4. Análisis de requisitos. Se estudian las necesidades de los usuarios, se decide qué debe hacer la aplicación informática para satisfacerlas en todo o en parte.
5. Diseño detallado. En esta quinta parte de la actividad de diseño se fijan las funciones de cada módulo, con el detalle de su interfaz. Se genera el código de declaración (o especificación) de cada módulo.
6. Codificación. Se desarrolla el código de cada módulo.
7. Al finalizar cada parte, hacer comprobaciones para confirmar que el funcionamiento es correcto al finalizar cada parte, es decir:
 - Si al hacer las comprobaciones el resultado no es el deseado, se volverá al punto 6 para realizar los cambios oportunos en la última parte desarrollada o en las anteriores, si es necesario.
 - Si al hacer las comprobaciones el resultado es el deseado, se desarrollará la siguiente parte volviendo al punto 6. Una vez se hayan finalizado todas las partes con sus respectivas comprobaciones, se iniciará el punto 8.
8. Integrar todas las partes desarrolladas y comprobar que el funcionamiento es correcto:
 - Si al hacer las comprobaciones el resultado no es el deseado, se volverá al punto 6 para realizar los cambios oportunos en la última parte desarrollada o en las anteriores, si es necesario.
 - Si al realizar las comprobaciones el resultado es el esperado, se iniciará el punto 9.
9. Pruebas de validación. Como paso final de la integración se realizan nuevas pruebas de la aplicación en su conjunto. En este caso el objetivo es comprobar que el producto desarrollado cumple con lo establecido en el documento de requisitos, y satisface por tanto las necesidades de los usuarios en la medida prevista.
10. Presentar la documentación.

Tal y como se puede ver, la metodología del proyecto permite dividir el proyecto en módulos y organizar en el tiempo el desarrollo, el tiempo de verificación y de corrección. Tanto durante el tiempo de desarrollo como de verificación se hizo un seguimiento mediante tutorías semanales o bisemanales dependiendo de la etapa, ya que en los inicios la manera de avanzar es mucho más lenta que al final y no siempre es necesario hacer tutorías semanalmente. Cuando se termina un módulo, siempre que se pueda, se finaliza totalmente de manera que no se vuelve a tocar, así

garantizamos que los errores que puedan surgir en el módulo actual son únicamente de éste y no de ninguno de los anteriores, o al menos que afecten lo menos posible.

La técnica que se ha seguido para la creación de esta aplicación es la de diseño descendente. Por eso hemos utilizado como lenguaje de especificación UML (Lenguaje de Modelado Unificado o Unified Modeling Language), que es un lenguaje de modelado estándar para la ingeniería de software. El UML se utiliza para definir un sistema, detallar sus elementos, documentarlo y construirlo. Para lograr esto, UML dispone de numerosos tipos de diagramas que muestran diversos aspectos de los elementos representados.

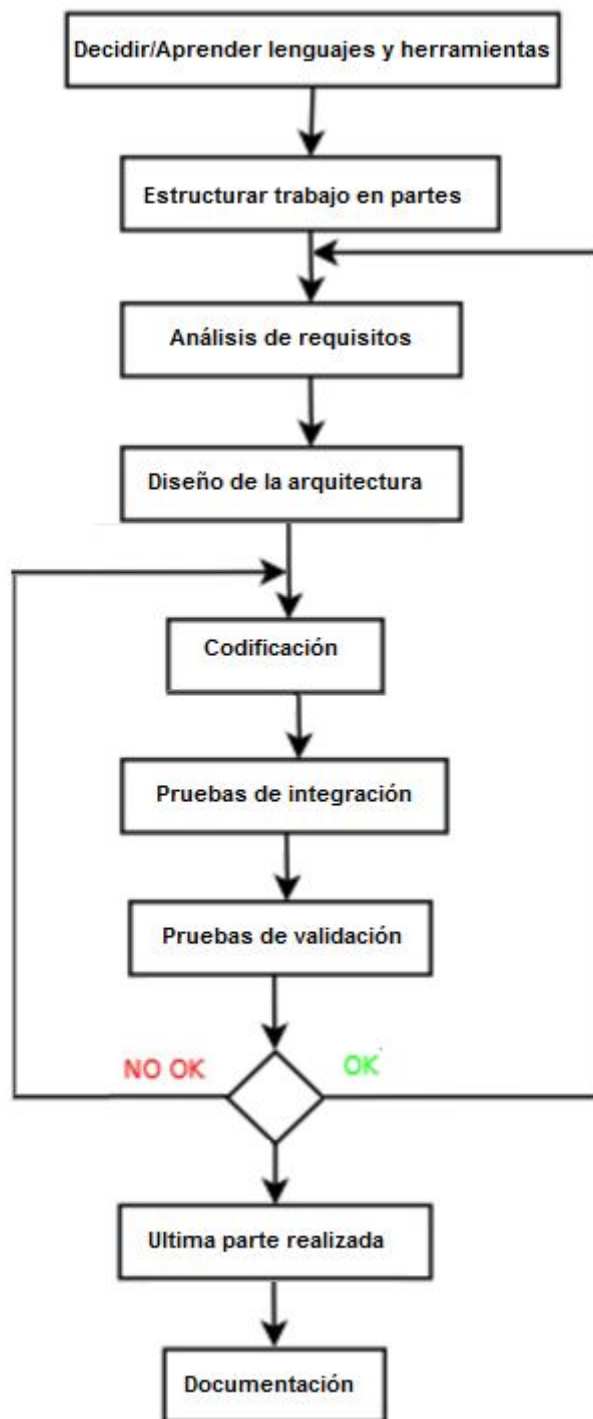


Figura 1: Diagrama de flujo de la metodología

Capítulo 4: Planificación

Este proyecto se inició al 01 de agosto del 2012 y se ha terminado a final de agosto de 2014.

En la primera etapa, fase de aprendizaje de las herramientas que se utilizaron, se documentó el código existente de la web, ya que no era fácil entender un código no documentado. La siguiente etapa era la toma de contacto con las herramientas necesarias para poder desarrollar el proyecto. Una vez aclarados los conceptos básicos, el siguiente paso fue introducirse en el mundo de los lenguajes de páginas web, lectura de manuales, lectura de código, ejecución de ejemplos, etc.

Una vez asimilados todos estos conceptos, el siguiente paso consistió en comenzar a implementar los algoritmos correspondientes a la primera tarea.

El objetivo de la primera fase consistió en recoger toda la información relacionada con el perfil general, de educación, y la accesibilidad de la tecnología de interacción y comunicación.

La segunda fase del proyecto, después de tener el modelo de usuario definido, fue diseñar un sistema de gestión de archivos, que proporcionará a los usuarios acceso y control de archivos y de directorios.

La tercera fase del proyecto, fue diseñar un sistema de noticias, que diese soporte para informar y comunicar a los usuarios que pertenecen a una comunidad donde se publicarán las noticias.

La cuarta tarea consistió en diseñar un sistema de gestión de permisos/privilegios. El objetivo de esta tarea consistió en que los usuarios con más privilegios puedan controlar qué tipo de acceso otorgan a otros usuarios, como dar o quitar los permisos: crear, leer, o borrar.

La siguiente fase fue hacer las interfaces de usuarios y la corrección de errores.

Por último, cabe la verificación y pruebas finales de los algoritmos implementados. En esta fase se realizaron comprobaciones de que los algoritmos hiciesen realmente lo que deben hacer en situaciones generales, y corregir los posibles errores que detectasen.

Finalmente, los dos últimos meses han sido dedicados a corregir errores que han venido surgiendo, mejorar el código ya realizar y organizar la memoria de este proyecto.

4.1- Tiempo estimado

Inicialmente se planeó que el proyecto durará 2 años, y la distribución de tareas que se eligió en un primer momento fue la que se puede observar en la figura 3:

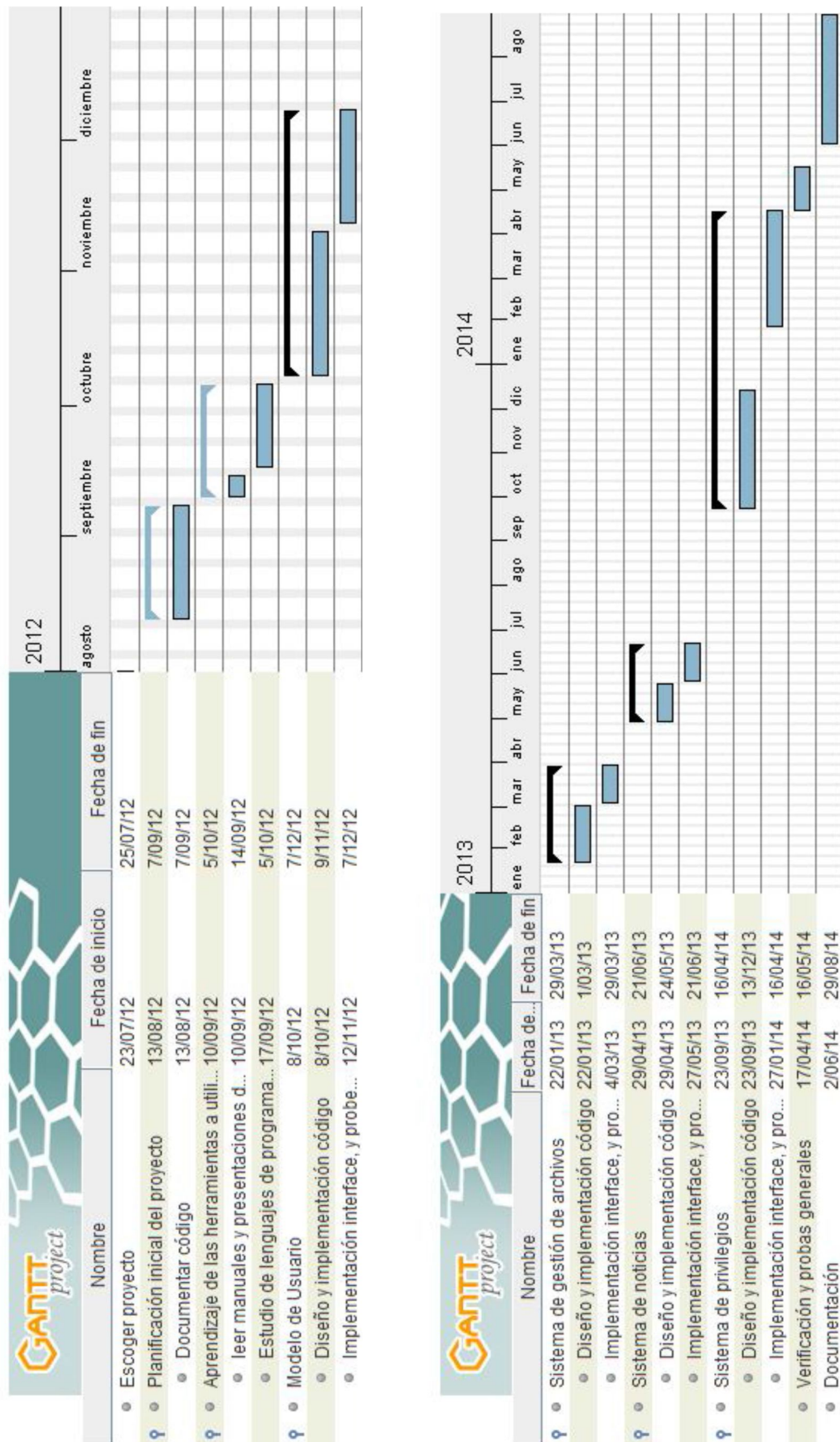


Figura 2: Diagrama de Gantt

4.2- Resultados estimados

Se espera añadir implementaciones totalmente funcionales, que pueda ser reutilizada en su totalidad, cumpliendo todas las expectativas y objetivos planteados en los apartados anteriores. Así, el usuario podrá acceder, crear, modificar, y eliminar noticias y foros según sus privilegios.

Capítulo 5: Marco de trabajo y conceptos previos

En este apartado se detallaran los conceptos necesarios para poder entender los algoritmos desarrollados y las herramientas utilizadas para poder llevar a cabo con éxito este proyecto.

Metodología

Para la confección de un software educativo en términos generales se deben seguir los siguientes pasos:

1. Definición del proyecto
2. Confección del modelo de desarrollo
3. Construcción de la web

La definición del proyecto debe incluir los objetivos y contenidos docentes de la web (qué y por qué), caracterización de los usuarios (quién) ambiente de uso (dónde y cuándo) recursos de desarrollo (cómo) y recursos de uso (con qué).

La confección del modelo de desarrollo debe incluir un conjunto de principios que inspiran y enmarcan el diseño e implementación del software, así como una secuencia de actividades o pasos que comprenden el modelo y diseño de la aplicación.

El modelo de desarrollo del proyecto debe tener presente considerar los siguientes principios subyacentes:

- Perceptivo: referente a lo que el usuario podrá percibir al usar la web, la aplicación tiene que ser sugerente; atractiva, que uno se sienta motivado a utilizarlo, con una apariencia llamativa y conocida, que invite su uso y explotación, que sea relevante.
- Metodológico: relacionado con los principios que sustentan el diseño de la forma de uso de la aplicación. Debe ser colaborativo para trabajo grupal, además debe ser complementario, el sistema complementa los conocimientos del tema. Este principio apoya la concepción de la computadora como una herramienta al servicio del profesor.
- Funcional: qué podrá hacer el usuario con la aplicación. El usuario tiene el control del mismo la mayor parte del tiempo; así como entregar un resultado. Este es un aspecto de retroalimentación que se basa en el reforzamiento de los logros del usuario, lo cual es importante desde el punto de vista psicológico.

La construcción de la web que debe incluir los siguientes aspectos:

1. Ya instalado el servicio World Wide Web, y las páginas confeccionadas crear un sitio Web donde se colocarán todas ellas y configurar dicho servicio para poder acceder al sitio.
2. Designar una persona responsable para atender los servicios, actualizarlos e interactuar con las personas que acceden a él y lo utilicen.

Capítulo 6: Requisitos del sistema

Los requisitos son el conjunto de funcionalidades que una aplicación debe cumplir, en este capítulo serán descritos los requerimientos. Los cuales explican a grandes trazos los objetos de la aplicación.

6.1- Necesidades tecnológicas

Requisitos desde el punto de vista de un usuario final:

- Disponer de un ordenador con:
 - o Sistema operativo a elegir según las preferencias del usuario.
 - o Un navegador a elegir según las preferencias del usuario.

Requisitos desde el punto de vista de un desarrollador:

- Disponer de un ordenador con:
 - o Sistema operativo a elegir según las preferencias del usuario.
 - o Capacidad suficiente para almacenar y compilar todas las librerías necesarias. Puede llegar a necesitarse según la modalidad de complicación y los ejemplos que se deseen hasta un 1Gb.

6.2- Descripción del sistema

El sistema contará con una aplicación cliente que se comunicará a través de la red con una serie de aplicaciones (servidores) que proporcionan a éstas una serie de servicios (acceso a los documentos compartidos, etc.).

Por parte de la aplicación cliente se tienen:

1. Conjunto de usuarios que pueden acceder a ella.
2. Conjunto de roles que pueden acceder a una aplicación.
3. Conjunto de componentes con los que un usuario (bajo un determinado rol) puede actuar.

Las relaciones que se establecen entre los elementos anteriores son las siguientes:

1. Un determinado usuario puede desempeñar más de un rol en el sistema, pero sólo puede desempeñar un único rol en un momento dado.
2. Se pueden especificar restricciones de acceso a un determinado componente para un determinado usuario, con un esquema: escritura/lectura (el usuario puede ver el componente e interactuar con él), sólo lectura (el usuario sólo podrá ver el componente) y acceso prohibido (el componente se esconde).
3. El mismo comportamiento del punto anterior de los usuarios con respecto a los componentes se da para los roles.

6.3- Técnicas utilizadas para la identificación de requerimientos

Para la identificación de los requerimientos que se presentan en este documento se han aplicado las técnicas siguientes:

- Entrevistas y reuniones
- Talleres estructurados
- Revisión de documentación técnica
- Desarrollo de prototipos

6.4- Requerimientos funcionales

En este apartado se describirán los servicios que ofrecerá esta aplicación, clasificadas según el tipo de usuario que accede al servicio.

▪ **Usuarios no registrados**

La única función disponible para los usuarios no registrados es la de autenticarse para acceder a la parte privada de la aplicación.

Apartado	Descripción
Título	Autenticación
Propósito	Acceder a la intranet de la aplicación
Entrada	Nombre de usuario y contraseña
Proceso	Se comprueba que el usuario existe
Salida	Confirmación de acceso a la intranet o aviso de que el nombre de usuario o contraseña no es/son válidos.

▪ **Usuarios registrados**

Los usuarios registrados en general sólo pueden cerrar su sesión para abandonar la intranet y volver de nuevo a la parte pública de la aplicación.

Apartado	Descripción
Título	Cerrar sesión
Propósito	Salir de la intranet de la aplicación
Entrada	-
Proceso	Se redirige a la página de inicio
Salida	-

▪ ***Usuario colaborador***

Sin embargo, si el usuario registrado es un usuario *Colaborador* puede consultar su modelo de usuario y cambiar su contraseña. Si pertenece a una comunidad, puede ver las noticias, los archivos y los foros de esa comunidad.

▪ ***Usuario colaborador Admin de un foro***

Si el usuario registrado es del tipo *Colaborador*, y administra Foros, tiene funciones como ver, crear, modificar y eliminar los foros de la comunidad a donde pertenece.

▪ ***Usuario administrador de la comunidad***

Si el usuario registrado es del tipo *administrador* de la comunidad, entonces a nivel de esa comunidad puede:

- Listar los usuarios
- Agregar, quitar usuarios a esa comunidad
- Cambiar permisos a los usuarios de la misma
- Cambiar nombre de la comunidad
- Eliminar comunidad
- Ver/crear foros
- Cambiar permisos a los usuarios respecto al foro
- Crear/eliminar carpetas
- Crear/eliminar ficheros
- Modificar perfil general
- Ver/crear noticias
- Eliminar noticias

▪ ***Usuario Administrador General***

El usuario registrado de tipo administrador dispone, a más de las funciones que tiene un usuario administrador de una comunidad, de las funciones para dar de alta y de baja a los colaboradores de toda la plataforma, de las comunidades, de las actividades, de las tareas, aprobar usuarios registrados, administrar (listar, aprobar, crear, modificar, eliminar) categorías, agregar usuarios a una comunidad, cambiar permisos a usuarios de una comunidad.

6.4. Requisitos no funcionales

Los requerimientos no funcionales, en cambio, describen aspectos sobre cómo debe ser el programa y no sobre lo que debe hacer. Debe cumplir lo siguiente:

- Usabilidad: El sistema estará dotado de una interfaz altamente usable, de manera que su aprendizaje y manejo resulte fácil para cualquier tipo de usuario.
 - o El sistema mostrará una interfaz con menús, ventanas, mensajes informativos y otras opciones familiares para el usuario.

- Disponibilidad: El programa debe poder usarse en cualquier lugar y en cualquier momento.
 - o El único requerimiento es disponer de internet debido a la conexión con el servidor en donde se encuentra la información.
 - o Se tiene que garantizar una alta disponibilidad de la aplicación, de manera que el servicio esté siempre disponible.
- Fiabilidad: El sistema a desarrollar ha de ser robusto. Intentando, en la medida de lo posible, que la tasa de fallos que presente el programa durante el uso por parte del usuario sea el menor posible.
- Mantenibilidad: Se deberán realizar revisiones periódicas del sistema, de manera que la aplicación se encuentre actualizada constantemente para evitar posibles fallos en los equipos o en la información y para garantizar un correcto funcionamiento del sistema.
 - o Para cumplir este requisito se deberán llevar a cabo tareas de tipo correctivo (modificaciones que se hacen en el sistema tras detectar defectos, ambigüedades o errores), preventivo (modificar el sistema con los cambios necesarios para mantener la eficiencia y fiabilidad del software), adaptativo (modificaciones en el sistema para acomodarlo a cambios físicos del entorno) y perfectivo (mejorar el sistema para cumplir con las nuevas necesidades/requerimientos de los usuarios).
- Rendimiento: El tiempo de respuesta ante las diferentes operaciones que el usuario puede realizar debe ser breve.
 - o Se deben evitar esperas del usuario cuando realice operaciones sobre actividades y tareas o cuando se actualice la información del servidor.
- Extensibilidad: Se ha de implementar la aplicación de tal manera que, en caso de que se encuentre la necesidad de introducir una nueva funcionalidad, se pueda añadir fácilmente sin un alto coste de programación.
 - o Implementar la aplicación muy modularmente.
 - o Actualizar, que indique cómo y dónde modificar en caso de necesidad.

6.5- Restricciones

6.5.1- Políticas

La aplicación se desarrollará con herramientas de software libre incluyendo el servidor de despliegue, programas de compilación, entornos de desarrollo, entre otros con el objetivo de evitar inconvenientes con el licenciamiento de software que no haya sido contemplado.

Se debe tener en cuenta que se ha planteado como requisito crítico el cumplimiento de las normas de accesibilidad en contenidos web de conformidad con las WCAG 2.0 para la interface de la plataforma de colaboración.

6.5.2- Interfaces con otras aplicaciones

La plataforma tiene acceso a un repositorio de objetos virtuales de aprendizaje desarrollando en el proyecto ALTER-NATIVA. También tiene acceso a la herramienta de referencias bibliográficas Zotero.

Sin embargo el sistema está diseñado para poder acceder a cualquier sistema que ofrezca una API REST.

6.5.3- Funciones de control

Funciones de validación de los datos que se ingresan al sistema.

6.5.4- Lenguaje de programación

El proceso de desarrollo del componente de software se realizará en base al lenguaje JavaScript.

6.5.5- Protocolos de comunicación

El componente de software que será desarrollado funcionará en entornos web y los protocolos de comunicaciones que se usarán serán HTTP y TCP/IP.

6.5.6- Requisitos de Fiabilidad

Los registros de colaboración resultado de la interacción de los colaboradores que sean construidos por los usuarios deben ser almacenados para que puedan ser eliminados o consultados posteriormente por parte de los usuarios.

6.5.7- Consideraciones acerca de la seguridad

El desarrollo de las actividades de colaboración está sujeto a que el colaborador previamente haya iniciado sesión en CoSpace.

Capítulo 7: Estudios y decisiones

7.1- Justificación del entorno de desarrollo

Este proyecto se ha integrado a la web con el mismo entorno elegido por el desarrollador al principio de la creación de la aplicación, con los lenguajes JavaScript, Jade-lang, Node.js y MongoDB con las librerías correspondientes, y debe cumplir varios requisitos.

Estas facilidades o requisitos mínimos que el entorno de desarrollo elegido ha de proveer son:

- a) Ser un entorno de desarrollo de última generación, probado, multiplataforma y con documentación suficiente sobre su funcionamiento.
- b) El mismo lenguaje para cliente y servidor (javascript)
- c) Poseer librerías y plugins, lo que permitirá centrar nuestro trabajo en los objetivos marcados, consiguiendo mejores resultados.
- d) Ser abierto, es decir, que permita la comunicación con otros sistemas, ya sean gestores de bases de datos, rutinas implementadas en otros lenguajes, etc.

- **HTML5**



HTML, siglas de 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. HTML se escribe en forma de "etiquetas", rodeadas por corchetes angulares (<,>).

El lenguaje HTML es un estándar reconocido en todo el mundo y cuyas normas define un organismo sin ánimo de lucro llamado World Wide Web Consortium, más conocido como W3C.

Como se trata de un estándar reconocido por todas las empresas relacionadas con el mundo de Internet, una misma página HTML se visualiza de forma muy similar en cualquier navegador de cualquier sistema operativo. El propio W3C define el lenguaje HTML como "un lenguaje reconocido universalmente y que permite publicar información de forma global".

Por convención, los archivos de formato HTML usan la extensión .htm o .html.

HTML5 es la última evolución de la norma que define [HTML](#). El término representa dos conceptos diferentes

- Se trata de una nueva versión del *lenguaje* HTML, con nuevos elementos, atributos y comportamientos,
- y un conjunto más amplio de tecnologías que permite a los sitios Web y las aplicaciones ser más diversas y de gran alcance. Este conjunto se le llama *HTML5* a menudo reducido a sólo *HTML5*.

Diseñado para ser utilizable por todos los desarrolladores de Open Web, una referencia a página enlaza numerosos recursos sobre las tecnologías de HTML5, que se clasifican en varios grupos según su función.

- *Semántica*: lo que le permite describir con mayor precisión cuál es su contenido.
- *Conectividad*: lo que le permite comunicarse con el servidor de formas nuevas e innovadoras.
- *Desconectado y almacenamiento*: permite a las páginas web almacenar datos, localmente, en el lado del cliente y operar fuera de línea de manera más eficiente.
- *Multimedia*: permite hacer vídeo y audio en la Web abierta.
- *Gráficos y efectos 2D/3D*: permite una gama mucho más amplia de opciones de presentación.
- *Rendimiento e Integración*: proporcionar una mayor optimización de la velocidad y un mejor uso del hardware del equipo.
- *Dispositivo de Acceso*: admite el uso de varios dispositivos de entrada y salida.
- *Styling*: deja a los autores escribir temas más sofisticados.

- **CSS**



CSS es un lenguaje para definir el estilo o la apariencia de las páginas web, escritas con HTML o de los documentos XML. CSS se creó para separar el contenido de la forma, a la vez que permite a los diseñadores mantener un control mucho más preciso sobre la apariencia de las páginas.

A partir del año 2005 se comenzó a definir el sucesor de esta versión, al cual se lo conoce como CSS3 o Cascading Style Sheets Level 3. Actualmente en la definición, esta versión nos ofrece una gran variedad de opciones muy importantes para las necesidades del diseño web actual. Desde opciones de sombreado y redondeado, hasta funciones avanzadas de movimiento y transformación, CSS3 es el estándar que dominará la web durante los siguientes años.

La idea que se encuentra detrás del desarrollo de CSS es separar la estructura de un documento de su presentación. La información de estilo puede ser adjuntada tanto como un documento separado o en el mismo documento HTML. En este último podrían definirse estilos generales en la cabecera del documento o en cada etiqueta particular mediante el atributo "style".

Las ventajas de utilizar CSS (u otro lenguaje de estilo) son:

- Control centralizado de la presentación de un sitio web completo, con lo que se agiliza de forma considerable la actualización del mismo.
- Los navegadores permiten a los usuarios especificar su propia hoja de estilo local que será aplicada a un sitio web remoto, con lo que aumenta considerablemente la accesibilidad.

- **Bootstrap**



Twitter Bootstrap es un [framework](#) o conjunto de herramientas de [software libre](#) para diseño de sitios y aplicaciones web. Contiene plantillas de diseño con tipografía, formularios, botones, cuadros, menús de navegación y otros elementos de diseño basado en [HTML](#) y [CSS](#), así como, extensiones de [JavaScript](#) opcionales adicionales.

- **JADE**



Es un motor de plantillas implementado con Javascript para simplificar la sintaxis HTML.

Jade proporciona una serie de estructuras de control como bucles, sentencias if/else, etc., que hace que se pueda generar de forma dinámica documentos HTML. Este módulo se ha utilizado para generar los contenidos del sistema en HTML teniendo plantillas base para ello.

- **JavaScript**



JavaScript (abreviado comúnmente "**JS**") es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.

Se utiliza principalmente en su forma del lado del cliente (client-side), implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas aunque existe una forma de JavaScript del lado del servidor (Server-side JavaScript o SSJS). Su uso en aplicaciones externas a la web, por ejemplo en documentos PDF, aplicaciones de escritorio (mayoritariamente widgets) es también significativo.

JavaScript se diseñó con una sintaxis similar al C, aunque adopta nombres y convenciones del lenguaje de programación Java. Sin embargo Java y JavaScript no están relacionados y tienen semánticas y propósitos diferentes.

Todos los navegadores modernos interpretan el código JavaScript integrado en las páginas web. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del Document Object Model (DOM).

Tradicionalmente se venía utilizando en páginas web HTML para realizar operaciones y únicamente en el marco de la aplicación cliente, sin acceso a funciones del servidor. JavaScript se interpreta en el agente de usuario, al mismo tiempo que las sentencias van descargándose junto con el código HTML.

- **jQuery**



jQuery es una biblioteca de JavaScript, creada inicialmente por John Resig, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web. Fue presentada el 14 de enero de 2006 en el BarCamp NYC. jQuery es la biblioteca de JavaScript más utilizada.

jQuery es software libre y de código abierto, posee un doble licenciamiento bajo la Licencia MIT y la Licencia Pública General de GNU v2, permitiendo su uso en proyectos libres y privados. jQuery, al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio.

- **Node.js**



Node.js es framework (entorno de programación y librería) basado en la máquina virtual V8 de Google (la misma que se utiliza para ejecutar Javascript en el navegador Google Chrome). Esto posibilita realizar programas en Javascript que se ejecutan en el lado servidor.

El objetivo de nodejs fue la creación de programas de red altamente escalables para construir aplicaciones dirigidas por eventos con baja latencia y concurrentes. Para ello, además del intérprete de Javascript, Node.js dispone de un pool de threads y de una biblioteca de eventos que se complementan con una enorme cantidad de bibliotecas asíncronas.

A pesar de ser una tecnología muy reciente, ha tenido mucha repercusión y ha sido adoptado incluso por empresas como:

- Microsoft: en Windows Azure se usa Node.js internamente para proveer almacenamiento en la nube para aplicaciones.
- Yahoo: en Yahoo! Manhatan Node.js es el núcleo de la aplicación.
- Palm/OS: todos los servicios para los dispositivos móviles de WebOS han sido desarrollados en Node.js.
- eBay: en el desarrollo de ql.io, un lenguaje inspirado en SQL y JSON para trabajar con APIs Hypertext Transfer Protocol (HTTP).

- **Express.js**

ExpressJS es un sistema que ayuda a organizar la parte del servidor de una aplicación web basada en nodejs utilizando el patrón Model-View-Controller ([MVC](#)). El sistema permite controlar prácticamente todo lo que está relacionado con el acceso al servidor, desde rutas hasta el manejo de las peticiones y vistas. Además permite el uso de lenguajes basados en plantillas como (jade, dust,...) que también se han utilizado para el desarrollo de la aplicación. Express se está usando en aplicaciones como MySpace, LearnBoost, Storify entre otros. En el sistema se ha utilizado como enrutador para la creación de las rutas a las que tienen acceso los usuarios cuando se conectan a la interfaz web. Este módulo convierte al sistema en un servidor web, ya que hay que tener en cuenta que Node.js es una librería de eventos y no un servidor web.

- **MongoDB**



MongoDB (de la palabra en inglés “humongous” que significa enorme) es un sistema de base de datos NoSQL orientado a documentos, desarrollado bajo el concepto de código abierto.

MongoDB forma parte de la nueva familia de sistemas de base de datos NoSQL. En vez de guardar los datos en tablas como se hace en las base de datos relacionales, MongoDB guarda estructuras de datos en documentos tipo JSON con un esquema dinámico (MongoDB llama ese formato BSON), haciendo que la integración de los datos en ciertas aplicaciones sea más fácil y rápida.

El desarrollo de MongoDB empezó en octubre de 2007 por la compañía de software 10gen. Ahora MongoDB es una base de datos lista para la producción de uso y con muchas características (feature). Esta base de datos es altamente utilizada en las industrias¹ y MTV Network,² Craigslist³ y Foursquare⁴ son algunas de las empresas que utilizan esta base de datos.

El código binario está disponible para los sistemas operativos Windows, Linux, OS X y Solaris.

- **Mongoose**

Mongoose es una herramienta para modelado de objetos en entornos asíncronos que sirve de enlace con MongoDB. Gracias a Mongoose se pueden definir esquemas de objetos para poder guardarlos en mongo, reflejando restricciones, tipos de objetos y muchas más características.

Mongoose se ha utilizado para definir los esquemas de los objetos para poder guardar los datos de forma homogénea en la base de datos.

- **SSH**



SSH (Secure **SH**ell, en español: intérprete de órdenes segura) es el nombre de un protocolo y del programa que lo implementa, y sirve para acceder a máquinas remotas a través de una red. Permite manejar por completo la computadora mediante un

intérprete de comandos, y también puede redirigir el tráfico de X para poder ejecutar programas gráficos si tenemos un Servidor X (en sistemas Unix y Windows) corriendo.

Además de la conexión a otros dispositivos, SSH nos permite copiar datos de forma segura (tanto archivos sueltos como simular sesiones FTP cifradas), gestionar claves RSA para no escribir claves al conectar a los dispositivos y pasar los datos de cualquier otra aplicación por un canal seguro tunelizado mediante SSH.

- **SmartGit**



SmartGit es un software de origen alemán desarrollado en Java que facilita el acceso a los repositorios GIT/SVN en forma visual, sin líneas de comandos. Tiene una versión Community gratuita y funciona en varios entornos operativos (Windows, MacOS, Linux/Unix).

SmartGit es un entorno fácil e intuitivo para trabajar con Git pero no contiene el software Git propiamente dicho, por lo cual hay que instalar primero este.

- **Redis**



Redis es un motor de base de datos en memoria, basado en el almacenamiento en tablas de hashes (clave/valor) pero que opcionalmente puede ser usada como una base de datos durable o persistente. Está escrito en ANSI C por Salvatore Sanfilippo quien es patrocinado por VMware. Y está liberado bajo licencia BSD por lo que es considerado software de código abierto.

- **Firebug**



Es una extensión Open Source del navegador Firefox en forma de plug-in creada y diseñada especialmente para desarrolladores y programadores web. Es un paquete de utilidades con el que se puede analizar, editar, monitorizar y depurar el código

fuelle, CSS, HTML y JavaScript de una página web de forma instantánea y en línea, que permite editar y guardar cambios.

Capítulo 8: Análisis y diseño del sistema

Un requerimiento es una característica de diseño, atributo, o comportamiento de un sistema. Con las necesidades de un sistema, se establece un pacto entre los requisitos de sistema y el sistema por sí mismo, en donde se declara que se espera que el sistema haga. En la mayor parte no se cuida de como lo hace, sólo se cuida que es lo que debe hacer. En este capítulo describimos lo que hace el sistema desde el punto de vista de un usuario utilizando UML ("Unified Modeling Language").

8.1- Diseño general

En este apartado se explican las herramientas que se utilizarán para poder desarrollar las diferentes partes del diseño de este proyecto. Para ello, lo primero que se utilizará son los diagramas de casos de uso.

8.2- Diagrama de casos de uso

Los diagramas de casos de uso nos permiten diferenciar los actores que interactúan con nuestra aplicación, las relaciones entre ellos y las acciones que puede realizar cada uno dentro del sistema.

Este tipo de diagramas son fácilmente comprensibles tanto por clientes como por usuarios, representan los requisitos funcionales del sistema y se utilizan como base para un desarrollo iterativo e incremental. Los diagramas de casos de uso tienen tres elementos:

- o **Actores:** Son los usuarios del sistema (figura 3). Un actor puede ser una persona, un conjunto de personas, un sistema hardware o un sistema software. Los actores representan un rol, que puede desempeñar alguien que necesita intercambiar información con el sistema.
- o **Casos de uso:** Un caso de uso describe una forma concreta de utilizar parte de la funcionalidad de un sistema. La colección de todos los casos de uso describe toda la funcionalidad del sistema.
- o **Comunicación entre actores y casos de uso:** Cada actor ejecuta un número específico de casos de uso en la aplicación. Por eso decimos que hay comunicación entre actores y casos de uso.

8.2.1- Identificación de los actores

A continuación se identifican los actores de la aplicación. Un actor es cualquier elemento que intercambia datos con el sistema. Es una entidad externa (persona, dispositivo, proceso, subsistema, tiempo,...). Un actor siempre tendrá interacción con el sistema ya sea al inicializar el caso de uso o intercambiando información.

Los actores:

- No son parte del sistema que se construye;
- Entran información al sistema;
- Reciben información del sistema.

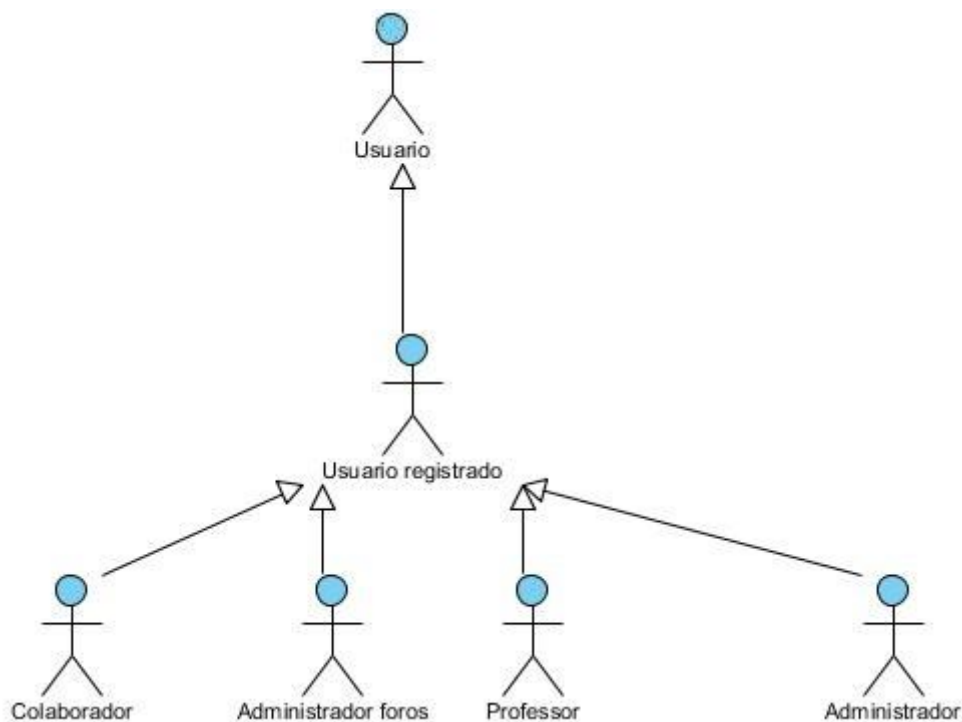


Figura 3: Actores

Todos los actores derivan del usuario no registrado (ver figura 3), que corresponde a aquella persona que aún no se ha identificado en el sistema. Siguiendo el esquema, los usuarios registrados heredan los casos de uso del usuario no registrado. Y el colaborador, administrador foros, profesor y administrador heredan los casos de uso del usuario registrado. A continuación se comentan los casos de uso principales de cada uno de los actores.

8.2.1.1- Casos de uso del usuario no registrado

Usuario no registrado: Corresponde a aquel actor que aún no se ha identificado en el sistema. Las acciones que podrá realizar son limitadas centrándose en la consulta de la información pública de la web o registrarse en el sistema (figura 4).

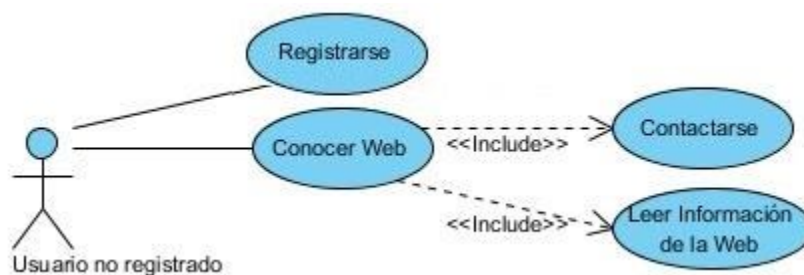


Figura 4: Diagrama de casos de uso de usuario no registrado

8.2.1.2- Casos de uso del Colaborador

Colaborador: Este tipo de usuario representa a un colaborador, podrá realizar las acciones de gestionar su perfil personal, ver noticias, ver, comentar los foros, crear carpetas, subir archivos, etiquetar archivos (figura 5).

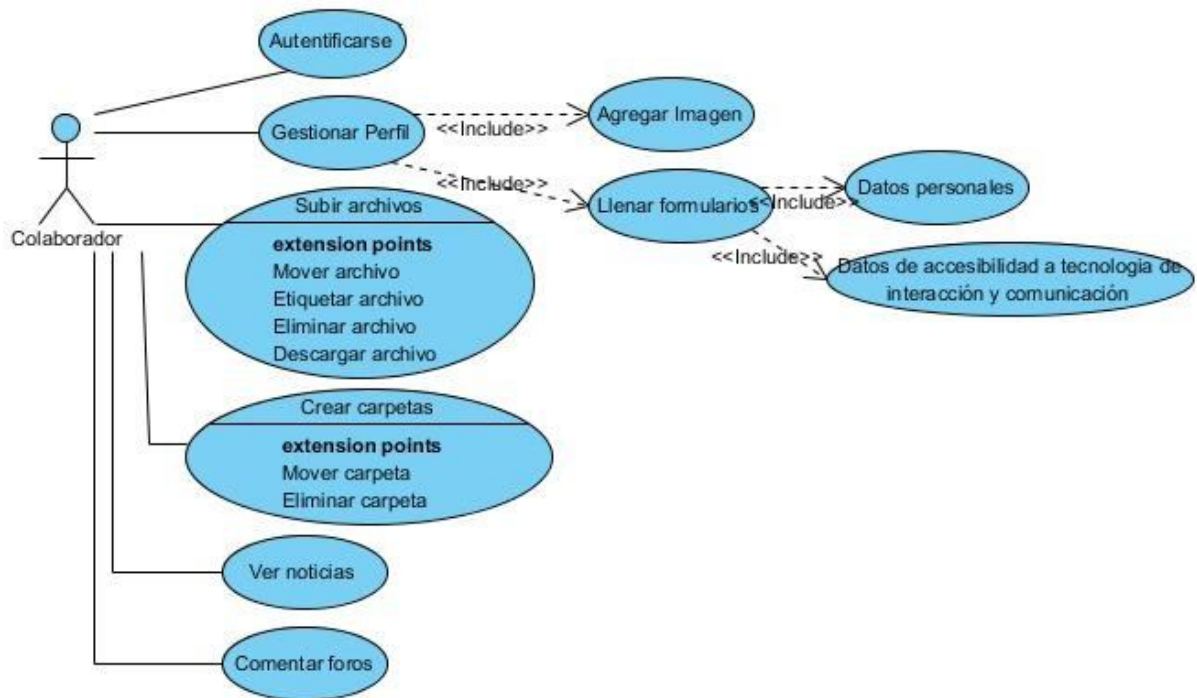


Figura 5: Diagrama de casos de uso de Colaborador

8.2.1.3- Casos de uso del colaborador administrador de foros

Administrador foros: Tanto el colaborador como el administrador de foros comparten los mismos casos de uso. La diferencia reside en que el administrador de foros, podrá crear comentar foros, y solicitar categorías (figura 6).

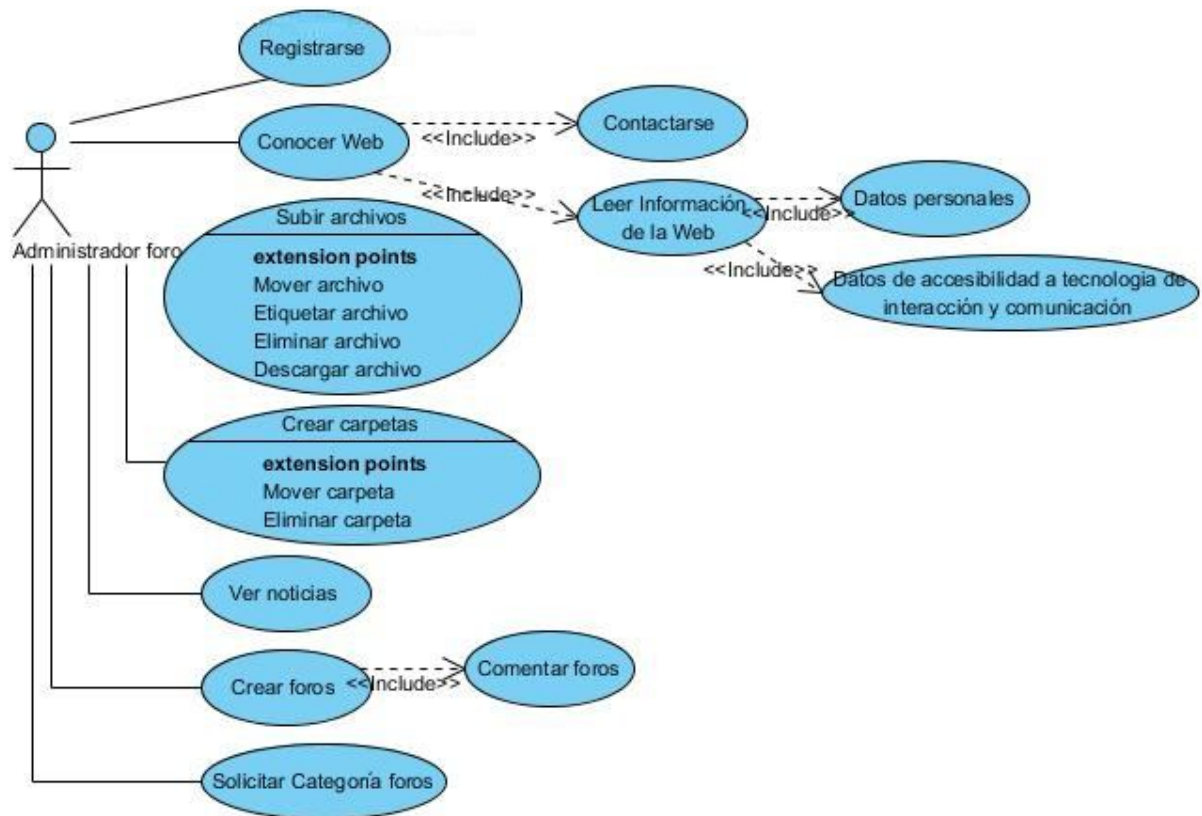


Figura 6: Diagrama de casos de uso de Colaborador administrador foros

8.2.1.4- Casos de uso del profesor

Profesor: Este tipo de usuario, representa a un profesor, podrá realizar las acciones de publicar, borrar las noticias de una comunidad, crear, modificar, comentar, y responder a un foro, listar, modificar los permisos de los usuarios de la comunidad, gestionar su perfil personal, dar de alta/baja la opción “Archivos” para que los usuarios de la comunidad visualicen la pestaña “Archivos” y puedan crear, mover carpetas, subir, mover y etiquetar archivos (figura 7).

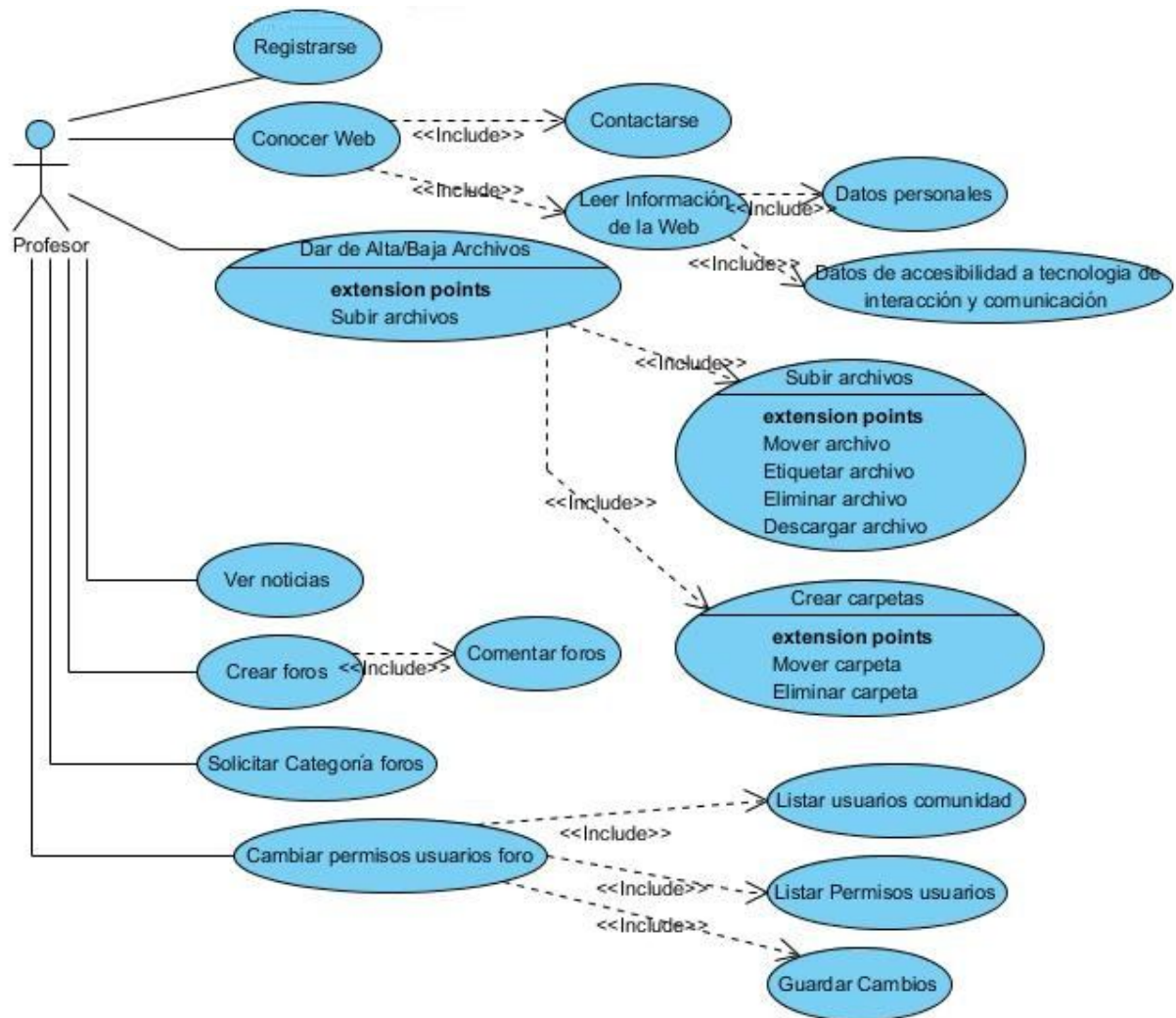


Figura 7: Diagrama de casos de uso de Profesor

8.2.1.5- Casos de uso del Administrador

Administrador: El administrador tendrá control total sobre el sistema, así además de poder realizar las mismas acciones que el profesor y el colaborador, también podrá gestionar los usuarios que tienen acceso al sistema y el contenido de la página Web (figura 8).

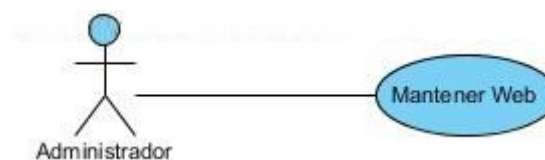


Figura 8: Diagrama de casos de uso de Administrador

8.2.2- Diagrama de caso de uso general

- Subir fichero

Caso de uso	Subir fichero
Descripción	Agregar un fichero nuevo al sistema
Actor	Todos usuarios registrados
Precondición	Usuario registrado El path del fichero debe existir de forma local al usuario El path de destino debe existir
Flujo principal	▪ Seleccionar la opción Subir fichero ▪ Guardar fichero
Subflujos	ninguno
Flujo alternativo	1.1. Sistema despliega ventana para buscar el documento a subir 2.1. Si los parámetros no cumplan la condición establecida da error.
Postcondición	El fichero se almacena en el sistema

- Etiquetar fichero

Caso de uso	Etiquetar fichero
Descripción	Etiquetar un fichero
Actor	Todos usuarios registrados
Precondición	Usuario registrado
Flujo principal	- o Seleccionar la opción Etiquetar del fichero - que debe etiquetar - o Escribir nombre de la etiqueta - o Guardar Etiqueta
Subflujos	ninguno
Flujo alternativo	1.1. El sistema muestra la plantilla para redactar el nombre de la etiqueta
Postcondición	Etiqueta creada

- Mover fichero

Caso de uso	Mover fichero
Descripción	Mover un fichero a otro directorio
Actor	Todos usuarios registrados
Precondición	Usuario registrado
Flujo principal	- o Seleccionar la opción Mover del fichero que debe mover - o Seleccionar un directorio - o Seleccionar Guardar
Subflujos	ninguno
Flujo alternativo	1.1. El sistema muestra la plantilla de todos los directorios y subdirectorios
Postcondición	Fichero movido

- Crear carpeta

Caso de uso	Crear carpeta
Descripción	Crear una carpeta
Actor	Todos usuarios registrados
Precondición	Usuario registrado
Flujo principal	- o Seleccionar la opción Crear carpeta - o Escribir nombre carpeta - o Guardar carpeta
Subflujos	ninguno
Flujo alternativo	1.1. El sistema muestra la plantilla para redactar el nombre de la carpeta
Postcondición	Carpeta creada

- Mover carpeta

Caso de uso	Mover Carpeta
Descripción	Mover una carpeta a otro directorio
Actor	Todos usuarios registrados
Precondición	Usuario registrado
Flujo principal	- o Seleccionar la opción Mover de la carpeta que debe mover - o Seleccionar un directorio - o Seleccionar Guardar
Subflujos	ninguno
Flujo alternativo	1.2. El sistema muestra la plantilla de todos los directorios y subdirectorios
Postcondición	Carpeta movida

- Publicar noticia

Caso de uso	Publicar noticia
Descripción	El usuario desea publicar una noticia
Actor	Profesor o administrador
Precondición	Usuario registrado
Flujo principal	- o Seleccionar la opción Crear noticia - o Redactar el título de la noticia - o Redactar el cuerpo de la noticia - o Seleccionar opción de enviar
Subflujos	ninguno
Flujo alternativo	1.1. El sistema muestra la plantilla para redactar la noticia
Postcondición	Noticia publicada

- Cambiar Permisos

Caso de uso	Cambiar Permisos
Descripción	Cambiar permisos usuarios de foros
Actor	Profesor
Precondición	Usuario registrado
Flujo principal	1. Seleccionar la opción Permisos 2. Seleccionar el usuario que quiere cambiar sus permisos 3. Cambiar permisos 4. Guardar cambios
Subflujos	ninguno
Flujo alternativo	1.1. El sistema listará todos usuarios y sus permisos en esa comunidad
Postcondición	Permisos cambiados

8.3- Diagrama de actividad

El diagrama de actividad se centra en el flujo de actividades involucradas en un proceso, generalmente dentro del marco de uno o varios casos de uso. Un diagrama de actividades muestra en qué orden se ejecutan las partes del proceso y cómo dependen unas de las otras.

El diagrama de actividad no proporciona información del comportamiento de un objeto o de las colaboraciones entre objetos.

En un diagrama de actividad, el proceso comienza a partir del círculo negro de inicio situado en la parte superior o izquierda del diagrama y termina en el círculo blanco/negro de final situado en la parte inferior o derecha del diagrama. Las actividades se indican con rectángulos redondeados.

Los diagramas de actividad se pueden subdividir en calles (swimlanes) para mostrar el responsable (actor, objeto, unidad organizacional, caso de uso,...) encargado de la actividad.

De cada actividad se deriva una transición que conecta con la siguiente actividad.

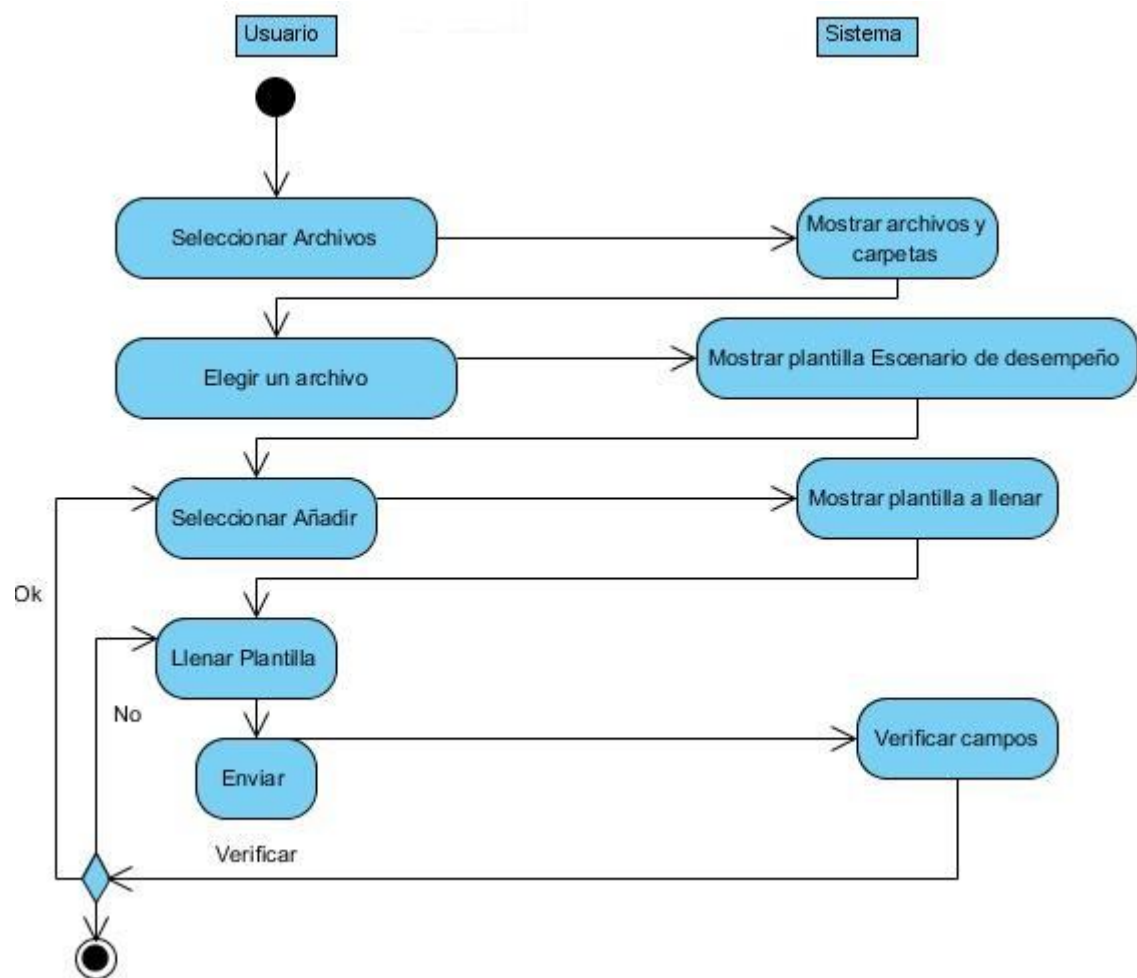


Figura 9: Diagrama de actividad Subir un archivo

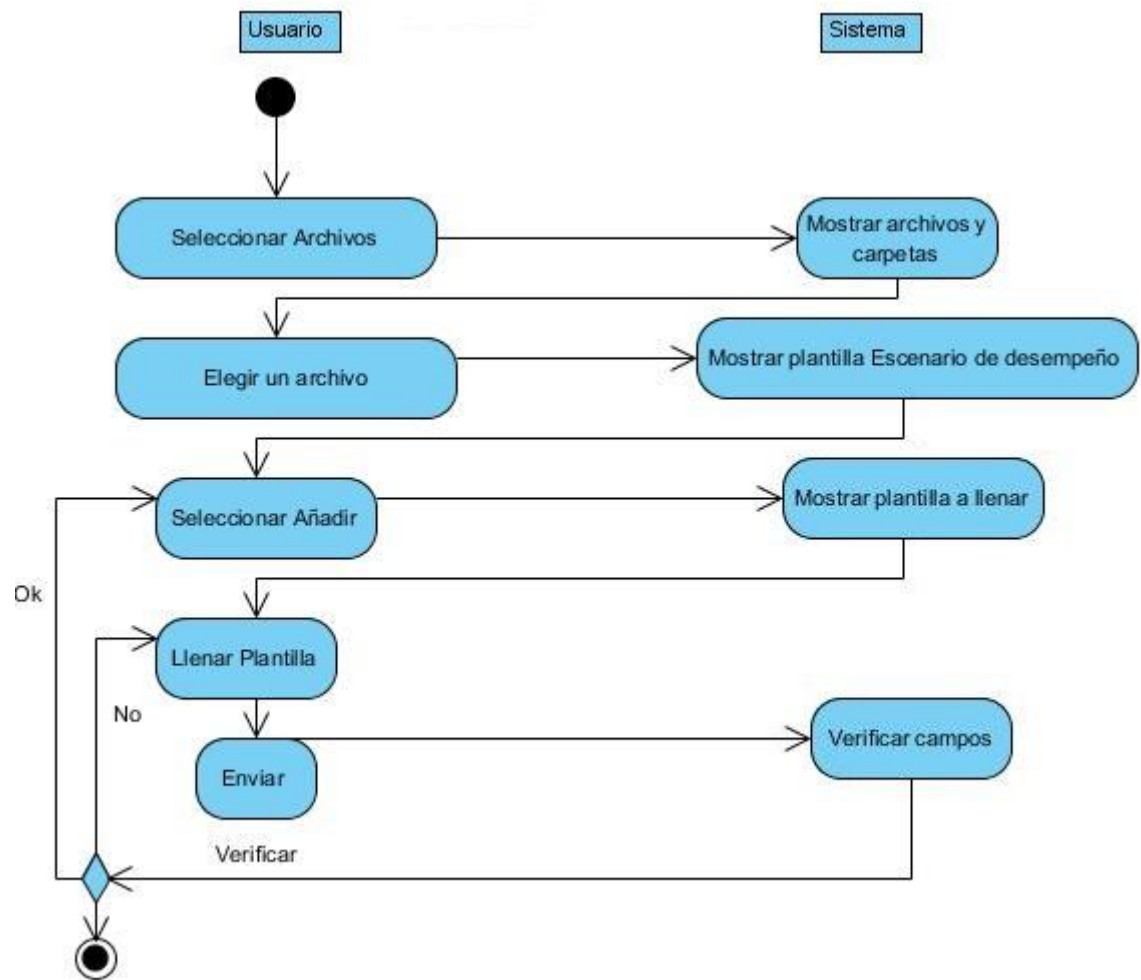


Figura 10: Diagrama de actividad Etiquetar un archivo

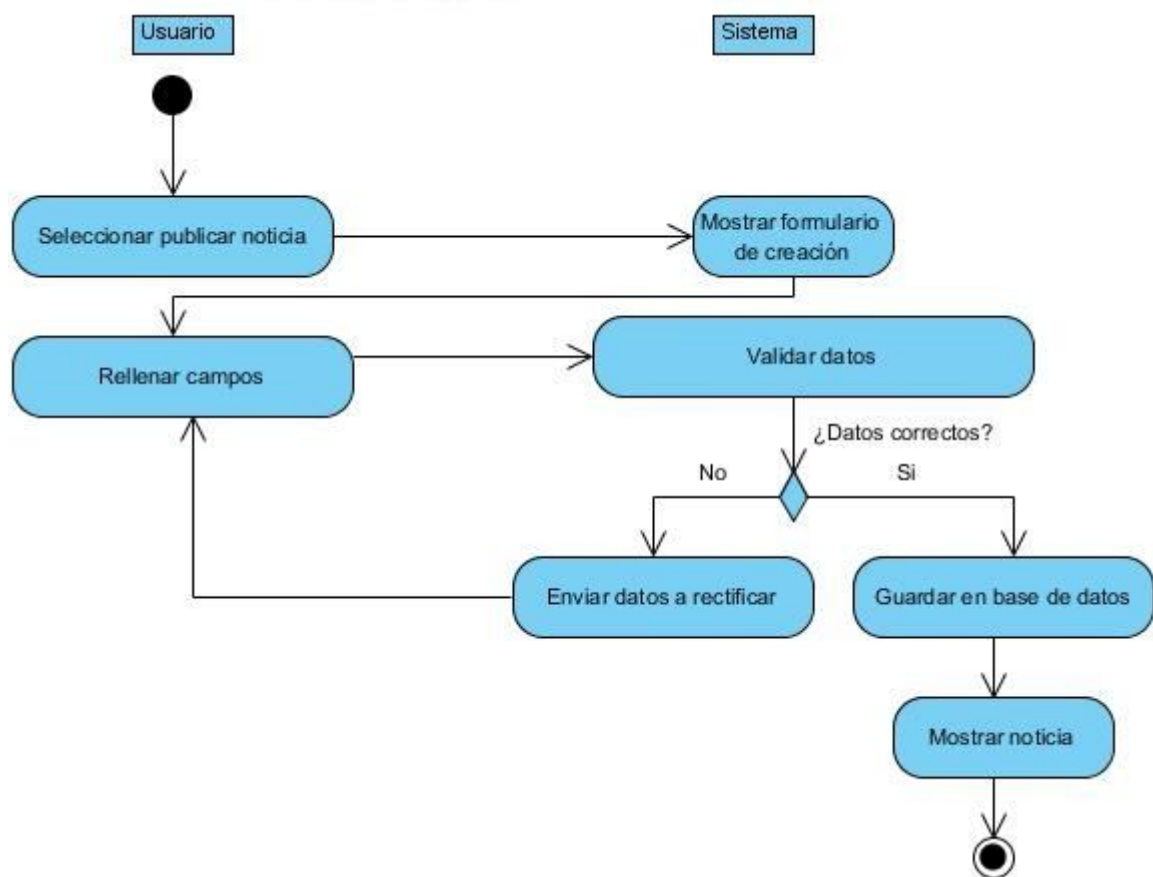


Figura 11: Diagrama de actividad Publicar una noticia

8.4- Diagrama de Entidad Relación

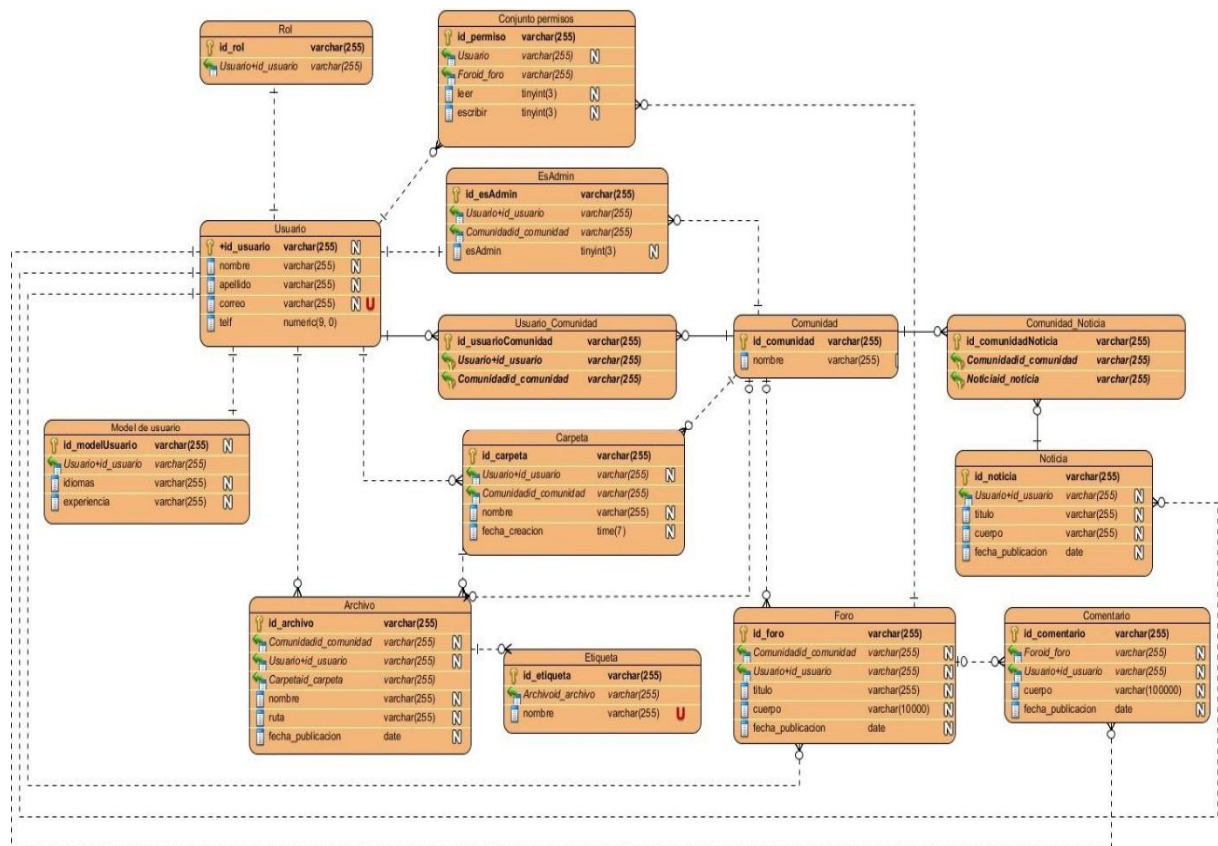


Figura 12: Diagrama de entidad relación

Como podemos ver en la figura 12, hay una serie de entidades que son las más importantes, como por ejemplo la entidad **usuario-permiso-conjunto permisos**. La relación entre **usuario** y **permiso** puede ser descrita como una relación de uno a muchos ya que cada usuario debe tener permisos (leer, escribir) de cada foro.

La relación entre **Usuario** y **esAdmin** (crear, modificar foros) es uno a varios. Un **Usuario** puede tener el privilegio administrador de todos los foros de una comunidad o más comunidades, o de ninguna.

8.5- Diagrama de clases

Un diagrama de clases es un tipo de diagrama estático que describe la estructura de un sistema mostrando sus clases y relaciones entre ellos. Los diagramas de clases son utilizados durante el proceso de análisis y diseño de los sistemas, donde se crea el diseño conceptual de la información que tendrá el sistema, y los componentes que se encargan de su funcionamiento y de las relaciones entre ellos.

A continuación se explican las clases utilizadas en este proyecto. Estas clases las podemos separar en dos grupos. Por un lado, las clases utilizadas ya existentes (ver figura 13):

- Usuario
- Rol
- Comunidad
- Foro
- Comentario

Finalmente, el resto de clases, que han sido creadas específicamente para este proyecto (ver figura 13).

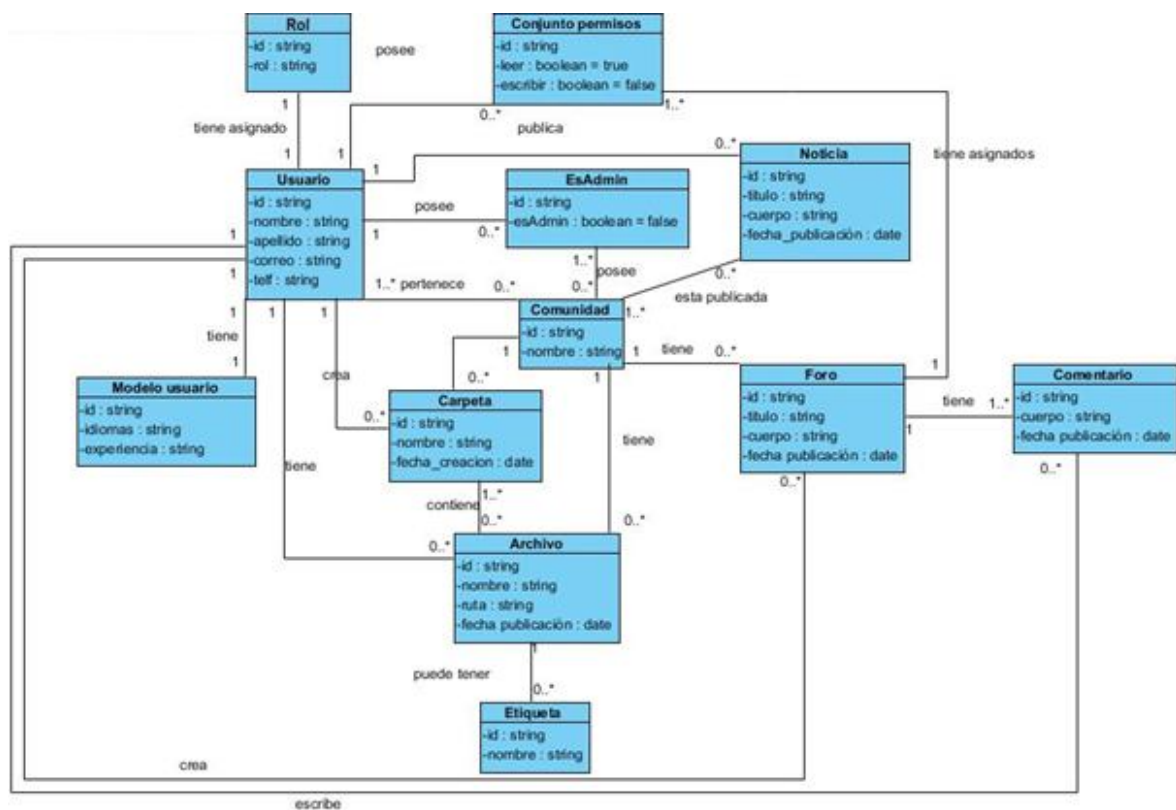


Figura 13: Diagrama de clases

8.6- Arquitectura Física

Este proyecto será una integración a la aplicación WEB ALTER-NATIVA, y la parte del servidor de la aplicación será la siguiente:

- Un servidor Web.
- Un SGBD para las tablas donde se guardarán los datos.

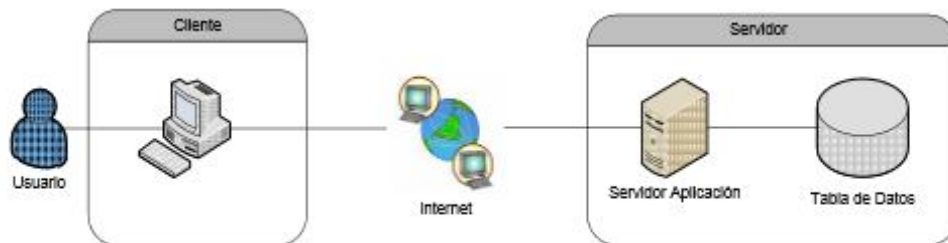


Figura 14: Arquitectura física de la web

8.6.1- Servidor Web

El servidor Web ofrece a los usuarios las herramientas necesarias para facilitar el acceso y manejo de la información. La estructura de la Web está dividida en dos partes:

- De acceso público: Esta parte equivale a la página Web, con su descripción y contacto.
- De acceso privado: Esta parte corresponde a la interfaz Web para la gestión de la información de la Base de Datos del entorno de trabajo. Para acceder a ella se utiliza un login, donde se pide el usuario y la contraseña. Dependiendo del rol de cada usuario (colaborador, profesor, Admin) las funcionalidades disponibles en esta interfaz son distintas para cada rol.

8.7- Modelo de programación

Se utilizará un modelo de tres capas para la programación.



8.7-1. Capa de presentación

Esta capa es la que ve el usuario, presenta el sistema al usuario, le comunica la información y captura la información del usuario en un mínimo proceso.

Esta capa se comunica únicamente con la capa de negocio. También es conocida como interfaz gráfica y debe tener la característica de ser “amigable” para el usuario.

8.7-2. Capa de Negocio

Aquí es donde, se reciben las peticiones del usuario y se envían las respuestas tras el proceso. Se denomina capa de negocio (e incluso de lógica de negocio) porque es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados y con la capa de datos para solicitar al gestor de base de datos almacenar o recuperar datos de él.

8.7-3. Capa de Persistencia

Es donde residen los datos y es la encargada de acceder a los mismos. Está formada por gestores de bases de datos que realizan todo el almacenamiento de los datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio.

Capítulo 9: Implementación y pruebas

Este capítulo describe los pasos que se han seguido a la hora de realizar la interfaz gráfica y detalles concretos de la implementación.

A continuación describiremos las clases desarrolladas. Explicaremos todos sus atributos y métodos.

9.1- Clases

9.1.1- Modelo usuario

El modelo de usuario contiene 14 clases, la podemos ver en la figura 15.

Los datos del modelo de usuario serán analizados para generar recomendaciones para los profesores de acuerdo a su perfil acerca de referencias bibliográficas que les puedan interesar, objetos de aprendizaje, o noticias.

Modelo de Usuario

Perfil general

- Perfil general



Perfil alternativa

- Escenarios de desempeño

Perfil especializado en educación

- Perfil General
- Perfil CAC
- Perfil CAM
- Perfil CALE

- Preferencias para el acceso a la información



Perfil de accesibilidad a tecnología de interacción y comunicación

Condicion sensorial

- Perfil limitación atencional
- Audición
- Visión
- insuficiencia motora de origen cerebral

Condicion lingüística

- Condicion lingüística

Condicion socio-economica

- Condicion socio-economica

Imagen Perfil

- Cambiar foto



Figura 15: Vista de la página del modelo de usuario

9.1.2- Noticia

La clase Noticia fue creada con el objetivo de comunicar y divulgar información de interés para los miembros de las comunidades dentro de la plataforma. Para poder crear y borrar una noticia, se tienen en cuenta los permisos del usuario de cada comunidad.

Noticia	
+id:	string
+titulo:	string
+cuerpo:	string
+fecha_publicacion:	date
+crearNoticia(idUsuario, idComunidad, titulo, cuerpo)	
+obtNoticias(idComunidad)	
+BorrarNoticia(id)	

Atributos

- + título: Variable de tipo *string* que contiene el título de la noticia.
- + cuerpo: Variable de tipo *string* que contiene el mensaje de la noticia.
- + fecha-publicación: Variable de tipo *date* que contiene la fecha que se ha publicado la noticia.

Funciones

- + crearNoticia: Función que crea una noticia en una o más comunidades.
- + obtNoticias: Mostrar todas las noticias de una comunidad.
- + BorrarNoticia: Borrar una noticia de una comunidad.

9.1.3- Conjunto de permisos

Un permiso es simplemente un conjunto de permisos, es decir, un conjunto de acciones que un usuario puede hacer en diferentes contextos (leer, y/o escribir comentarios en un foro, en un hilo de discusión).

A la hora de crear un foro, ya se crean todos los permisos posibles para los usuarios de la comunidad donde se ha creado el foro, según el rol de cada usuario, si el usuario tiene como rol usuario, se crea la clase conjunto Permisos con los privilegios leer = true, escribir = false, si el usuario tiene como rol administrador de la comunidad se crean los privilegios leer = true y escribir = true.

Conjunto Permisos	
+id:	string
+leer:	boolean
+escribir:	boolean
+crearCjtPermiso(idUsuario, idForo)	
+borrarCjtPermisosUsuario(idUsuario, idForo)	
+borrarCjtPermisosForos(idForo)	
+borrarCjtPermisosUsuario(idUsuario)	

```

+borrarCjtPermisosUsuarioComunidad(idUsuario, idComunidad)
+borrarCjtPermisos(idComunidad)
+obtCjtPermisosUsuario(idUsuario)
+obtPermisosForos(idForo)
+obtPermisosUsuarioForo(idUsuario, idForo)
+actualizarCjtPermisosUsuario(idUsuario, idForo)

```

Atributos

- + leer: Variable de tipo *booleano* si inicializa siempre con un true.
- + escribir: Variable de tipo *booleano* si inicializa según el rol de usuario, si tiene role administrado de la comunidad si inicializa con un true, sino false.

Funciones

- + crearCjtPermiso: Función que crea un conjunto de permisos para un foro de una comunidad.
- + borrarCjtPermisosUsuario: Función que borra todos los permisos de un usuario en un foro.
- + borrarCjtPermisosForos: Función que borra todos los permisos de todos usuarios de un foro.
- + borrarCjtPermisosUsuario: Funcion que borra todos los permisos de un usuario.
- + borrarCjtPermisosUsuarioComunidad: Función que borra todos los permisos de un usuario en una comunidad.
- + borrarCjtPermisos: Función que borra todos los permisos de una comunidad.
- + obtCjtPermisosUsuario: Función que nos retorna los permisos de un usuario.
- + obtPermisosForo: Función que nos retorna los permisos de todos los usuarios en un foro.
- + obtPermisosUsuarioForo: Función que nos retorna los permisos de un usuario en un foro.
- + actualizarCjtPermisosUsuario: Función que modifica y actualiza los permisos de un usuario en un foro.

A continuación se muestran algunos ejemplos de código de algunas funciones:

Código de la función para crear un conjunto de permisos en un foro para todos los usuarios de una comunidad.

```

// Crear permisos a los usuarios de una comunidad
//E:: idUser: identificador usuario, idsForo: identificador foro, idComu: ident comunidad
//S:: crear permisos del usuario respecto a su role (administrador o usuario)
ConjuntoPermisos.statics. crearCjtPermisos = function(idsUser,idForum, idComu, fn) {
    var longitud = idsUser.length;
    var i, role=null, idUser;

    // función que nos retorna el rol de un usuario, consultando la tabla de permisos
    var getRoleUser = function(id, idCom, callback) {

```

```

        models.permissions.find({ 'user' : id, 'resource': idCom}, function (err, docs){
            if(err){
                return fn(err);
            }
            callback(docs[0].role);
        });
    }

    for(i=0; i<=longitud; i++){ // crear permisos para todos los usuarios de la comunidad
        idUser = idsUser[i].id;
        var that = this;
        var j=0;
        getRoleUser(idUser,idComu,function(role){
            // defaults values for all users not admin
            if(role==1){ // role user
                var ConjuntoPermisos = new that({
                    'read': true
                    , 'write': false
                    , 'userId': idsUser[j].id
                    , 'communityId': idComu
                    , 'forold': idForum
                });
            } else { // if admin
                var ConjuntoPermisos = new that({
                    'read': true
                    , 'write': true
                    , 'userId': idsUser[j].id
                    , 'communityId': idComu
                    , 'forold': idForum
                });
            } //end else
            j=j+1;
            newCjtPermisos.save(function (err) {
                if (err) return fn(err);
                fn(null, newCjtPermisos.id);
            });
        });
    }; // end for
};

```

Ejemplo de código de la función buscar permisos de un foro

```

// buscar los usuarios de un foro
//E:: forold: identificador de foro
//S:: retorna doc o err
ConjuntoPermisos.statics. obtPermisosForo = function(idfor,fn){
    models. conjuntoPermisos.find({ 'forold' : idfor }, function (err, doc){
        if(err){
            return fn(err);
        }
    }
}

```

```

        if(!doc || (doc.length <= 0))
        {
            return fn('No se ha encontrado el dato en la base de datos...');
        }
        return fn(null,doc);
    });
};

```

Función que busca los permisos de un usuario

```

// buscar los usuarios de un foro
//E:: forold: identificador de foro
//S:: retorna doc o err
ConjuntoPermisos.statics.obtPermisosUsuarioForo = function(idfor, idUser, fn){
    models.conjuntoPermisos.find({ 'forold' : idfor, 'userId':idUser}, function (err,
doc){
        if(err){
            return fn(err);
        }
        if(!doc || (doc.length <= 0))
        {
            return fn('No se ha encontrado el dato en la base de datos...');
        }
        return fn(null,doc);
    });
};

```

Código de la función que actualiza el conjunto de permisos de un usuario en un foro:

```

//Función para actualizar los permisos de un usuario en un foro
ConjuntoPermisos.statics. actualizarCjtPermisosUsuario = function(id,forold,leer,
escribir, fn){
    models.conjuntoPermisos.update({ 'userId' : id, 'forold' : forold}, { 'leer' : leer,
'escribir' : escribir}, function(err,numberAffected){
        if(err)
            return fn(err);
        fn(null, numberAffected)
    });
};

```

Código de la función borrar todos los conjuntos de permisos en una comunidad

```

//Borrar todos los permisos de un usuario en una comunidad
//E:: idUser: identificador usuario, idCom: identificador comunidad
//S:: objeto borrado en la tabla conjunto de permisos
ConjuntoPermisos.statics. borrarCjtPermisosUsuarioComunidad = function(idUser,
idCom, fn) {
    models.conjuntoPermisos.remove({'userId': idUser, 'communityId':
idCom}, function(err, docs) {
        if (err) return fn(err);
    });
};

```

```

fn(null);

});
};

```

9.1.4- EsAdmin

Los permisos EsAdmin de usuarios de una comunidad se asignan en el momento de agregar colaboradores a la comunidad.

Esta clase actúa como clase de control, se encarga de crear y comentar los foros de una comunidad.

EsAdmin	
+Id:	string
+esAdmin:	boolean
+crearPermisoAdmin(idUsuario, idComunidad)	
+borrarPermisoAdminUsuario(idUsuario, idComunidad)	
+borrarPermisoAdminComunidad(idComunidad)	
+obtPermisoAdminUsuario(idUsuario, idComunidad)	
+obtPermisoAdminComunidad(idComunidad)	
+actualizarPermisoAdmin(idUsuario, idComunidad)	

Atributos

- + esAdmin: Variable de tipo *booleano* si inicializa según el rol de usuario, si tiene role administrado de la comunidad si inicializa con un true, sino false.

Funciones

- + crearPermisoAdmin: Función que crea el permiso esAdmin para todos los foros de una comunidad.
- + borrarPermisoAdminUsuario: Función que borra todos los permisos de un usuario.
- + borrarPermisoAdminComunidad: Función que borra todos los permisos de un usuario de una comunidad.
- + obtPermisoAdminUsuario: Función que nos retorna los permisos de un usuario.
- + obtPermisoAdminComunidad: Función que nos retorna los permisos de todos los usuarios en una comunidad.
- + actualizarPermisoAdmin: Función que modifica y actualiza los permisos de un usuario.

9.1.5- Archivo

La clase Archivo, ha sido diseñada para que proporcione áreas de almacenamiento compartidas con acceso a los materiales por parte de todos usuarios de una comunidad.

Archivo	
+Id:	string
+nombre:	string
+ruta:	string
+ fecha_publicacion:	string
+crearArchivo(idUsuario, idComunidad)	
+borrarArchivo(idArchivo)	
+borrarArchivoUsuario(idUsuario, idComunidad)	
+borrarArchivoComunidad(idComunidad)	
+borrarArchivoCarpeta(idCarpeta)	
+obtArchivoUsuario(idUsuario, idComunidad)	
+obtArchivoComunidad(idComunidad)	
+moverArchivo(idCarpeta)	

Atributos

- + nombre: Variable de tipo *string* que contiene el nombre de archivo.
- + ruta: Variable de tipo *string* que contiene la ruta de archivo.
- + fecha_publicacion: Variable de tipo *date* que contiene la fecha de descarga de archivo.

Funciones

- + crearArchivo: Función que crea el archivo en una comunidad.
- + borrarArchivo: Función que borra un archivo en una comunidad.
- + borrarArchivoUsuario: Función que borra un archivo de un usuario en una comunidad.
- + borrarArchivoComunidad: Función que borra todos los archivos subidos de una comunidad.
- + borrarArchivoCarpeta: Función que borra todos los archivos en una carpeta.
- + obtArchivoUsuario: Función que nos retorna los archivos de un usuario en una comunidad.
- + obtArchivoComunidad: Función que nos retorna los archivos de una comunidad.
- + moverArchivo: Función que mueve un archivo a otro directorio

9.1.6- Etiqueta

La clase Etiqueta se crea para organizar archivos por proyecto sin tener que moverlos a una carpeta, así resulta sencillo encontrar los archivos etiquetados sin importar dónde estén, a más se puede etiquetar un archivo más de una vez.

Etiqueta	
+Id:	string
+nombre:	string
+crearEtiqueta(idArchivo)	
+borrarEtiqueta(idArchivo)	
+borrarEtiquetas(idArchivo)	

Atributos

- + nombre: Variable de tipo *string* que contiene el nombre de la etiqueta.

Funciones

- + crearEtiqueta: Función que crea una etiqueta de un archivo en una comunidad.
- + borrarEtiqueta: Función que borra una etiqueta de un archivo en una comunidad.
- + borrarEtiquetas: Función que borra todas las etiquetas de un archivo en una comunidad.

9.1.7- Carpeta

La clase Carpeta se crea para organizar elementos (archivos y carpetas).

Carpeta	
+Id:	string
+nombre:	string
+fecha_publicacion:	date
+crearCarpeta()	
+borrarCarpeta(idCarpeta)	
+borrarDirectorio(idCarpeta)	
+obtElementosCarpeta(idCarpeta)	
+moverCarpeta(idCarpeta)	

Atributos

- + nombre: Variable de tipo *string* que contiene el nombre de la carpeta.
- + fecha_creación: Variable de tipo *date* que contiene la fecha de creación de la carpeta.

Funciones

- + crearCarpeta: Función que crea una carpeta en una comunidad.
- + borrarCarpeta: Función que borra una carpeta en una comunidad.
- + borrarDirectorio: Función que borra una carpeta con sus elementos, carpetas y archivos que contiene en una comunidad.
- + obtElementosCarpeta: Función que nos retorna los elementos, carpetas y archivos que contiene la carpeta.
- + moverCarpeta: Función que mueve una carpeta a otro directorio.

9.2- Diseño de la base de datos

En la actualidad, dependiendo del modelo de administración de datos que se siga, existen distintos tipos de bases de datos: jerárquicas, en red, transaccionales, relacionales, multidimensionales, orientadas a objetos, documentales, deductivas, etc. La base de datos del sitio Cospace se podría clasificar como una base de datos relacional, debido a que está compuesta por distintas tablas de datos (relaciones) pero no presenta **ninguna interconexión** entre los campos de las tablas. La principal consecuencia de este hecho es la duplicidad de registros, es decir, se almacena **contenido duplicado**.

La tabla ROL almacena el tipo de rol (usuario, o administrador de la comunidad) de cada usuario a la hora de agregar un usuario a una comunidad, siendo la clave primaria la identificador de Rol, y la clave foránea el identificador de usuario.

La tabla Usuario almacena los datos de un colaborador (nombre y apellidos, correo electrónico, teléfono, contraseña), siendo la clave primaria la identificador de usuario.

La tabla ESADMIN, almacena el permiso de si es administrador o no de los foros por una comunidad, siendo la clave primaria la identificador de esAdmin, y las claves foráneas, el id de usuario e id de comunidad.

La tabla CONJUTOPERMISOS, almacena el permiso de un usuario de si puede ver, comentar un foro, siendo las claves foráneas, el id de usuario, el id de foro.

La tabla CARPETA, almacena los datos de una carpeta (nombre, fecha de creación, nombre de creador, directorio: quiere decir en que directorio se ubica), siendo las claves foráneas el id del usuario, el id de la carpeta, el id de la comunidad.

La tabla ETIQUETA, almacena las etiquetas de un archivo (nombre), siendo la clave foránea el id de archivo.

La tabla MODELO DE USUARIO, almacena los datos de perfil general, perfil especializado en educación, y perfil de accesibilidad a tecnología de interacción y comunicación, siendo la clave foránea el id de usuario.

La tabla NOTICIA, almacena las noticias de una comunidad o comunidades (título, cuerpo, fecha de creación, nombre de creador), siendo las claves foráneas el id de usuario, y el id de comunidad.

9.3- Pruebas de funcionamiento

Como indicamos en el análisis. Todos los casos de uso se han sometido a pruebas de funcionamiento con el relleno de una ficha por cada caso de uso para ver su estado y si cumple los requisitos que se pusieron en un primer momento. Estas pruebas han sido realizadas por el proyectante.

A continuación se muestra las fichas de las pruebas realizadas sobre la aplicación.

Para ello se han seguido los diferentes casos de uso que se plantearon en el análisis del proyecto.

Prueba nº. 1	Satisfactoria: Si
Caso de uso: Rellenar Perfil general	
Comentarios:	
o El usuario puede rellenar sus datos de perfil general de forma correcta.	

Prueba nº. 2	Satisfactoria: Si
Caso de uso: Modificar Perfil general	
Comentarios:	
o El usuario puede modificar sus datos de perfil general de forma correcta.	

Prueba nº. 3	Satisfactoria: Si
Caso de uso: Rellenar Escenario de desempeño	
Comentarios:	
o El usuario puede rellenar sus datos de Escenario de desempeño de forma correcta.	

Prueba nº. 4	Satisfactoria: Si
Caso de uso: Modificar Escenario de desempeño	
Comentarios:	
o El usuario puede modificar sus datos de Escenario de desempeño de forma correcta.	

Prueba nº. 5	Satisfactoria: Si
Caso de uso: Borrar Escenario de desempeño	
Comentarios:	
o El usuario puede borrar sus datos de Escenario de desempeño de forma correcta.	

Prueba nº. 6	Satisfactoria: Si
Caso de uso: Rellenar Perfil CAC	
Comentarios:	
o El usuario puede rellenar sus datos de Perfil CAC de forma correcta.	

Prueba nº. 7	Satisfactoria: Si
Caso de uso: Modificar Perfil CAC	

Comentarios:
o El usuario puede modificar sus datos de perfil CAC de forma correcta.

Prueba nº. 8	Satisfactoria: Si
Caso de uso: Rellenar Perfil CAM	
Comentarios:	
o El usuario puede modificar sus datos de perfil CAM de forma correcta.	

Prueba nº. 9	Satisfactoria: Si
Caso de uso: Borrar Perfil CAM	
Comentarios:	
o El usuario puede modificar sus datos de perfil CAM de forma correcta.	

Prueba nº. 10	Satisfactoria: Si
Caso de uso: Rellenar Perfil CALE	
Comentarios:	
o El usuario puede rellenar sus datos de perfil CALE de forma correcta.	

Prueba nº. 11	Satisfactoria: Si
Caso de uso: Modificar Perfil CALE	
Comentarios:	
o El usuario puede modificar sus datos de perfil CALE de forma correcta.	

Prueba nº. 12	Satisfactoria: Si
Caso de uso: Rellenar Condición sensorial - Audición	
Comentarios:	
o El usuario puede rellenar sus datos de Condición sensorial - Audición de forma correcta.	

Prueba nº. 13	Satisfactoria: Si
Caso de uso: modificar Condición sensorial - Audición	
Comentarios:	
o El usuario puede modificar sus datos de Condición sensorial - Audición de forma correcta.	

Prueba nº. 14	Satisfactoria: Si
Caso de uso: Rellenar Condición sensorial - Visión	
Comentarios:	

o El usuario puede rellenar sus datos de Condición sensorial - Visión de forma correcta.
--

Prueba nº. 15	Satisfactoria: Si
Caso de uso: modificar Condición sensorial - Visión	
Comentarios:	
o El usuario puede modificar sus datos de Condición sensorial - Visión de forma correcta.	

Prueba nº. 16	Satisfactoria: Si
Caso de uso: Rellenar Condición sensorial – insuficiencia motora de origen cerebral	
Comentarios:	
o El usuario puede rellenar sus datos de Condición sensorial – insuficiencia motora de origen cerebral de forma correcta.	

Prueba nº. 17	Satisfactoria: Si
Caso de uso: Modificar Condición sensorial – insuficiencia motora de origen cerebral	
Comentarios:	
o El usuario puede modificar sus datos de Condición sensorial – insuficiencia motora de origen cerebral de forma correcta.	

Prueba nº. 18	Satisfactoria: Si
Caso de uso: Rellenar Condición lingüística	
Comentarios:	
o El usuario puede rellenar sus datos de Condición lingüística de forma correcta.	

Prueba nº. 19	Satisfactoria: Si
Caso de uso: Modificar Condición lingüística	
Comentarios:	
o El usuario puede modificar sus datos de Condición lingüística de forma correcta.	

Prueba nº. 20	Satisfactoria: Si
Caso de uso: Rellenar Condición socio-económica	
Comentarios:	
o El usuario puede rellenar sus datos de Condición socio-económica de forma correcta.	

Prueba nº. 21	Satisfactoria: Si
Caso de uso: Modificar Condición socio-económica	
Comentarios:	
o El usuario puede modificar sus datos de Condición socio-económica de forma correcta.	

Prueba nº. 22	Satisfactoria: Si
Caso de uso: Subir foto usuario en el modelo de usuario	
Comentarios:	
o El usuario puede subir su foto de forma correcta.	

Prueba nº. 23	Satisfactoria: Si
Caso de uso: Cambiar foto usuario en el modelo de usuario	
Comentarios:	
o El usuario puede cambiar su foto de forma correcta.	

Prueba nº. 24	Satisfactoria: Si
Caso de uso: Crear una noticia	
Comentarios:	
o El administrador de la comunidad puede crear una noticia de forma correcta.	

Prueba nº. 25	Satisfactoria: Si
Caso de uso: Borrar una noticia	
Comentarios:	
o El administrador de la comunidad puede borrar una noticia de forma correcta.	

Prueba nº. 26	Satisfactoria: Si
Caso de uso: Subir fichero	
Comentarios:	
o El usuario puede subir un fichero de forma correcta.	

Prueba nº. 27	Satisfactoria: Si
Caso de uso: Descargar fichero	
Comentarios:	
o El usuario puede descargar un fichero de forma correcta.	

Prueba nº. 28	Satisfactoria: Si
Caso de uso: Mover fichero	
Comentarios:	
o El usuario puede mover un fichero a otro directorio de forma correcta.	

Prueba nº. 29	Satisfactoria: Si
Caso de uso: Borrar fichero	
Comentarios:	
o El usuario puede borrar un fichero de forma correcta.	

Prueba nº. 30	Satisfactoria: Si
Caso de uso: Etiquetar fichero	
Comentarios:	
o El usuario puede etiquetar un fichero de forma correcta.	

Prueba nº. 31	Satisfactoria: Si
Caso de uso: Crear carpeta	
Comentarios:	
o El usuario puede crear una carpeta de forma correcta.	

Prueba nº. 32	Satisfactoria: Si
Caso de uso: Borrar carpeta	
Comentarios:	
o El usuario puede borrar una carpeta de forma correcta.	

Prueba nº. 33	Satisfactoria: Si
Caso de uso: Mover carpeta	
Comentarios:	
o El usuario puede mover una carpeta de forma correcta.	

Prueba nº. 34	Satisfactoria: Si
Caso de uso: Ver un foro	
Comentarios:	
o El usuario puede Ver un foro de forma correcta.	

Prueba nº. 35	Satisfactoria: Si
Caso de uso: Comentar un foro	
Comentarios:	
o El usuario puede comentar un foro de forma correcta.	

Prueba nº. 36	Satisfactoria: Si
Caso de uso: Administrar un foro	
Comentarios:	
o El administrador de foros puede gestionar foros de forma correcta.	

Prueba nº. 37	Satisfactoria: Si
Caso de uso: Cambiar permisos usuarios	
Comentarios:	
o El usuario puede cambiar permisos a usuarios de forma correcta.	

9.4- Acceso a la aplicación

Para acceder a la aplicación hay que entrar en el siguiente enlace:

<http://boppo.udg.edu:8000/colabora3/>

9.5- Problemas encontrados durante la realización del proyecto

La dificultad principal del proyecto ha resultado estar en el aprendizaje de las diferentes herramientas: JavaScript, Nodejs, Mongodb, Jade-lang. También nos hemos encontrado con una serie de puntos en los que habrá que hacer especial mención. La resolución de estos problemas ha obligado a investigar mucho y conocer a fondo el funcionamiento de las librerías utilizadas.

- **Ejemplos de estos problemas con soluciones:**

A la hora de bajar los cambios al servidor mediante el servidor SSH, siempre tenía que buscar soluciones porque no funcionaban los cambios que funcionaban en el servidor local de pruebas, la solución fue trabajar con el control de versiones SmartGit.

La dificultad principal del proyecto ha resultado estar en el aprendizaje de las diferentes herramientas: JavaScript, Nodejs, Mongodb, Jade-lang. También bajar los cambios al servidor mediante el servidor SSH, siempre tenía que buscar soluciones alternativas porque no funcionan los cambios como en el servidor local de pruebas, la solución fue trabajar con el control de versiones SmartGit.

Otra dificultad fue como listar las carpetas y subcarpetas, la manera que se ha solucionado fue por ejemplo tenemos una lista de carpetas:

[Fotos, Apuntes], [Naturaliza],[Mates],[Algebra, Análisis]

La primera lista tiene las carpetas iniciales, para listar las subcarpetas de cada una, empezamos por la primera carpeta, Fotos está en la posición (0,0), y en total hay una carpeta, eso significa que sus subcarpetas están en la lista 1 (carpeta Naturaliza), si contamos el número de carpetas hasta Naturaliza son 3, eso significa que subcarpetas de Naturaliza están en la posición 3 que es una lista vacía. Ahora vamos a la segunda carpeta de la primera lista (Apuntes), el número de carpetas son 2, pues sus subcarpetas están en la posición 2 (Mates), ahora contamos las carpetas hasta Mates en total son 4, las subcarpetas de Mates están en la posición 4 (Algebra i Análisis) y así sucesivamente.

La resolución de estos problemas ha obligado a investigar mucho y conocer a fondo el funcionamiento de las librerías utilizadas.

Capítulo 10: Resultados

En este capítulo se mostrarán los resultados obtenidos siguiendo la metodología descrita en el apartado 3. Estos resultados se pueden ver en algunas capturas de pantalla que acompañan las explicaciones.

10.1- Modelo de usuario

En la figura 14 podemos ver la interfaz de usuario para el modelo de usuario, como se puede ver es sencilla y comprensible.

A continuación se muestra unos ejemplos de formularios del modelo de usuario, las otras se mostraron en el manual de usuario.

Modelo de Usuario

Condicion lingüística

Primera Lengua (L1)

☐ Español

☐ Quechua

☐ Aymara |

☐ Miskito

☐ Zapoteco

☐ Mixteco

☐ Lengua de Señas Colombiana

☐ Lengua de Señas Mexicana

☐ Lengua de Señas Nicaragüense

Segunda Lengua (L2)

☐ Español

☐ Castellano Escrito

☐ Quechua

☐ Aymara

☐ Miskito

☐ Zapoteco

☐ Mixteco

Nivel de conocimiento del Español

Nivel de conocimiento de la Lengua Materna o natural

Otras lenguas

Figura 16: vista de modelo de usuario – Condición Lingüística

Modelo de Usuario

Visión

Tipo de limitación visual

☐ Total

☐ Baja

☐ Congénita

☐ Adquirida

☐ Súbita

☐ Gradual

Manejo de ayuda Visuales

☐ Tecnologías Analógicas

☐ Tecnología computacional

Mediaciones en el aprendizaje

☐ Braille |

☐ Abaco

☐ Lector de pantalla ¿Cuál?

☐ Calculadoras parlantes

☐ Sistemas de reconocimiento de textos en tinta

☐ Uso de libros hablados

☐ Textos convertidos en audio con voz artificial (sintetizada)

☐ Móviles adaptados con lector de pantalla |

Figura 17: vista de modelo de usuario – Visión

Modelo de Usuario

Perfil CAM

Experiencias curriculares para la formación matemática y didáctica

☐ 0 a 1 añitos

☐ 2 a 5 años

☐ 6 a 10 años

☐ Más de 11 años

Experiencias en la formación de una didáctica de la didáctica de las matemáticas de educadores matemáticas

☐ 0 a 1 año

☐ 2 a 5 años

☐ 6 a 10 años |

☐ Más de 11 años

Experiencias en ambientes de aprendizaje interculturales y pluritecnológicos en la educación matemáticas.

☐ 0 a 1 año

☐ 2 a 5 años

☐ 6 a 10 años

☐ Más de 11 años

Figura 18: vista de modelo de usuario – Perfil CAM

10.2- Sistema de gestión de archivos

La figura 19 muestra la vista de sistema de gestión de archivos, como se puede ver hay carpetas creadas, y archivos compartidos por diferentes usuarios.

The screenshot shows a web interface titled 'Documentos'. At the top, there are buttons for 'Nueva Carpeta' and 'Subir un archivo'. Below this, there are two main sections: 'Carpetas' and 'Ficheros individuales'.

Carpetas

Nombre	Creador	Borrar	Mover
Fotos	rachid mahmoudi	BORRAR	MOVER
libros	rachid mahmoudi	BORRAR	MOVER
Documentos	rachid mahmoudi	BORRAR	MOVER
Peliculas	rachid mahmoudi	BORRAR	MOVER

Ficheros individuales

Nombre	Etiqueta	Creador	Fecha creacion	Bajar	Borrar	Mover	Etiquetar
tarjeton_emacs.pdf		rachid mahmoudi	19/7/2014 13:46	BAJAR	BORRAR	MOVER	ETIQUETAR
zoho.jpg		admin admin	19/7/2014 14:19	BAJAR	BORRAR	MOVER	ETIQUETAR
cssForo.txt		rachid mahmoudi	19/7/2014 10:15	BAJAR	BORRAR	MOVER	ETIQUETAR
Koala.jpg	foto	admin admin	2/5/2014 10:9	BAJAR	BORRAR	MOVER	ETIQUETAR

At the bottom left, there is a 'Volver' button.

Figura 19: vista de archivos y carpetas

10.3- Noticias

En la figura 20 se pueden ver las noticias de una comunidad en la parte derecha de la página.

The screenshot shows a community interface. On the left, there are navigation links for 'Archivos', 'Repositorio', and 'Zotero', each with a 'SI' button. Below these are 'Actividades' with buttons for 'Trabajar con Actividades', 'Ver foros', and 'Administrar Actividades'. Further down is 'Administrar la comunidad' with buttons for 'Editar Info', 'Colaboradores', and 'Eliminar'.

On the right, there is a 'Noticias' sidebar. It contains two news items:

- Segonda Noticia** by rachid mahmoudi, dated 19/7/2014 14:27. The text says: 'es una notica de prueba de funcionamiento de las publicaciones.' There is a 'BORRAR' button below it.
- Noticia prueba** by rachid mahmoudi, dated 19/7/2014 14:26. The text says: 'podeis acceder a los **archivos** para descargar los'. There is a 'BORRAR' button below it.

At the bottom of the sidebar, there is a 'CREAR NOTICIA' button.

Figura 20: vista de noticias

10.4- Permisos

En la siguiente figura 21 se puede ver los foros ordenados per categoría, con cada foro su lista de usuarios con permisos.

Ver foros [Nuevo foro](#) [Nueva Categoría](#)

Repositorio		Fecha creación	Permisos
Foros			
	Hamming En est video se explica el funcionamiento del codigo de Hamming	7/8/2014	
	demo 25 audio mp3	7/8/2014	
	prueba para el curso zip de prueba para luego adjuntar el del curso	7/8/2014	
	titulo repo resumen repo	7/8/2014	
	titulo repo resumen repo	7/8/2014	

Zolero		Fecha creación	Permisos
Foros			
	Videos No fue posible recuperar el resumen de esta referencia	7/8/2014	

[Volver](#)

Figura 21: vista de foros de una comunidad con el permiso esAdmin

El acceso a los permisos de un foro, se puede ver en la siguiente página (figura 22)

Tabla de permisos de los usuarios del foro

* La opción Admin, si marca para dar permiso a crear y modificar los foros de esta comunidad

Nombre	Apellido	Leer	Comentar	Admin*	Guardar
rachid	mahmoudi	☒	☒	☒	Guardar
admin	admin	☒	☒	☒	Guardar

[Volver](#)

Figura 22: vista los permisos de usuarios

Ahora mostramos las vistas de cada permiso, en la siguiente figura (figura 23) se muestra la vista para un usuario con permiso de leer, ya se observa en la barra superior que no puede Crear foros ni cambiar permisos a otros usuarios.

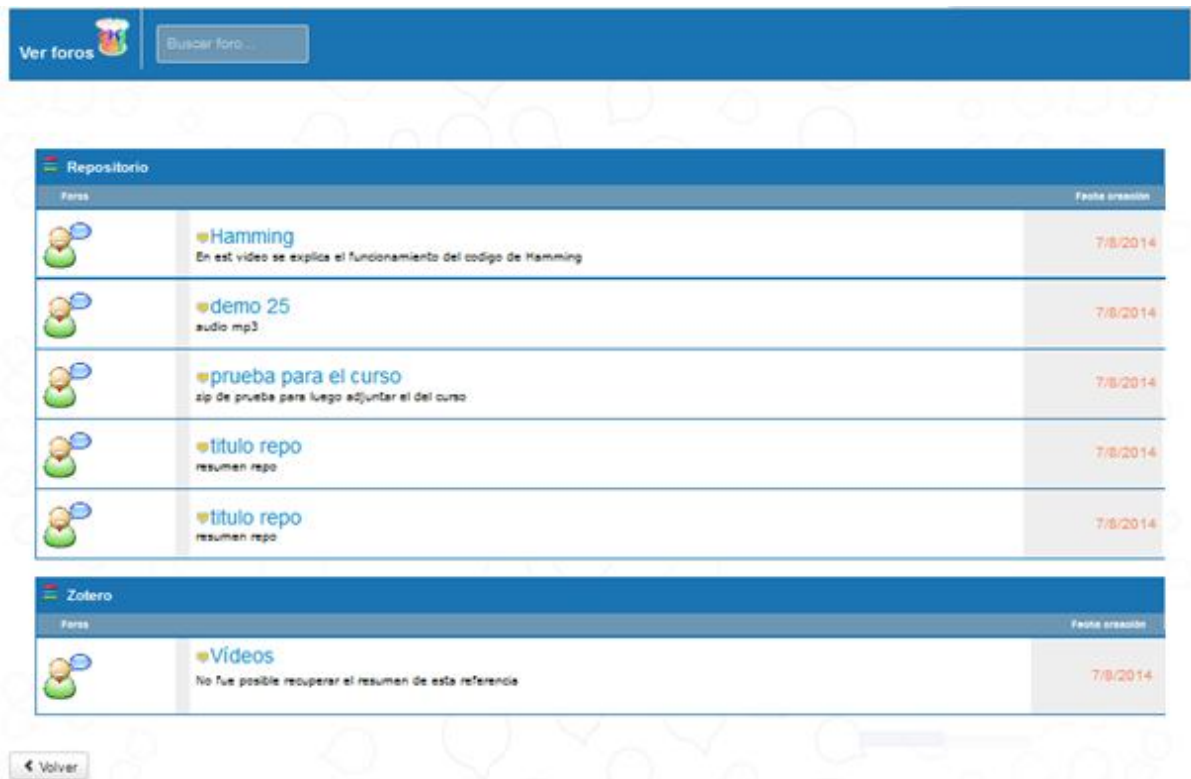


Figura 23: vista de los foros de una comunidad con el permiso leer o escribir



Figura 24: vista de un foro de una comunidad con el permiso leer



Figura 25: vista de un foro de un usuario con el permiso leer y escribir

En caso que un usuario no tenga ningún permiso, en el momento de entrar a los foros le saldrá la notificación que no hay foros para este usuario.

Capítulo 11: Conclusiones

Durante la realización del proyecto he podido comprobar lo útiles que son los conocimientos adquiridos durante la carrera en asignaturas como Ingeniería del Software y Alternativas de diseño de aplicaciones.

La realización de este proyecto ha permitido adquirir nuevos conocimientos sobre distintas tecnologías que actualmente tienen mucha relevancia, llegándose a las siguientes conclusiones:

- Se ha aprendido a diseñar correctamente, separando el contenido de la presentación, utilizando hojas de estilo (CSS), lo que facilita tener distintos estilos para un mismo portal web.
- Se tiene que diseñar teniendo siempre en cuenta las funcionalidades que debe aportar una aplicación y pensando en el usuario, dejando a un lado ideas o intereses personales. También hay que tener muy presente la usabilidad, ya que facilita la interacción usuarios - portal web, lo que aporta satisfacción de los mismos y aumenta el prestigio del portal web.
- La administración dinámica de un portal web es fundamental cuando se maneja una gran cantidad de información, cualquier usuario sin unos conocimientos amplios de informática será capaz de mantener el sitio.
- Se ha profundizado en la programación web, utilizando los lenguajes JavaScript, Nodejs, y Jade-lang para interaccionar con la base de datos MongoDB.

Por último, hay que destacar, que he podido poner en práctica todo lo aprendido en la universidad, desde el comienzo del proyecto con la toma de requisitos, pasando por la planificación, diseño de la aplicación y base de datos, implementación de las mismas y por supuesto durante la realización de la presente memoria. He comprobado que todo lo aprendido ha sido de utilidad, aunque cuando cursaba las asignaturas no me pareciese que tuviera mucho que ver con informática. Además he podido poner a prueba estos conocimientos en el mundo laboral y he podido ver que todo ha sido útil y me veo completamente capacitado para poder afrontar cualquier adversidad que pueda surgirme en el ámbito laboral.

Capítulo 12: Trabajo futuro

Tras la realización de este proyecto, se plantean distintas líneas de trabajo:

- Mejorar la accesibilidad del portal web, de modo que éste pueda ser accedido tanto por personas con algún tipo de discapacidad, como por cualquier otra persona y desde cualquier tipo de dispositivo.
- Mejorar la interfaz gráfica
- Un nuevo sistema de control de acceso en Archivos y Carpetas, permitiendo a los administradores tener control sobre quién puede ver y administrar el contenido.
- Opción búsqueda de archivos por nombre de usuario, nombre de archivo, extensión, o por nombre de etiqueta en la parte de Archivos.

Capítulo 13: Bibliografía

- ✓ "Diseño Inclusivo: Marco Metodológico Para El Desarrollo De Sitios Web Accesibles." *Diseño Inclusivo: Marco Metodológico Para El Desarrollo De Sitios Web Accesibles*. Web. 19 Aug. 2012.
<http://www.nosolousabilidad.com/articulos/disenio_inclusivo.htm>.
- ✓ *Metodología Para La Creación De Sitios Web Orientados a La Enseñanza De Las Ciencias Médicas*. Web. 12 Sep. 2012.
<http://bvs.sld.cu/revistas/mciego/vol5_01_99/articulos/a1_v5_0199.html>.
- ✓ All Questions. (n.d.). Retrieved October 21, 2012, from <http://stackoverflow.com/questions>
- ✓ Node.js v0.10.30 Manual & Documentation. (n.d.). Retrieved October 26, 2013, from <http://nodejs.org/api/domain.html>
- ✓ Tutorials for JFreeChart, Lucene, Selenium, DTD, Powerpoint, Tcl/Tk, XML, Software Engineering and many more latest technologies. (n.d.). Retrieved August 19, 2012, from <http://www.tutorialspoint.com/>

- ✓ Introducción a JavaScript. (n.d.). Retrieved July 15, 2012, from <http://librosweb.es/javascript/>
- ✓ Bootstrap 3, el manual oficial. (n.d.). Retrieved May 1, 2012, from http://librosweb.es/bootstrap_3/
- ✓ CSS avanzado. (n.d.). Retrieved August 12, 2012, from http://librosweb.es/css_avanzado/
- ✓ Build software better, together. (n.d.). Retrieved November 10, 2012, from <https://github.com/>
- ✓ Jade. (n.d.). Retrieved February 5, 2012, from <http://jade-lang.com/tutorial/>
- ✓ Yohan Graterol, *MongoDB en Español*, TOMO 1. EL PRINCIPIO

Capítulo 14: Anexos

CoSpace es un subproyecto del proyecto ALTER-NATIVA con el objeto de consolidar un espacio colaborativo, en las dimensiones educativa y tecnológica, como escenario para la conformación y el desarrollo de comunidades de práctica para la integración de tecnologías de la información y la comunicación, con el fin de cubrir las necesidades educativas en contextos de diversidad, NEED.

Y con la misión: Establecer vínculos académicos en red con instituciones de educación superior, entidades interesadas y personas vinculadas a la educación, formal y no formal, de poblaciones con necesidades educativas diversas, NEED, desarrollando estrategias que promuevan la aplicación de tecnologías de la información y la cooperación entre las áreas pedagógicas y tecnológicas, para reducir la disparidad en los logros de aprendizaje, la brecha digital, la desigualdad en el acceso a oportunidades educativas de calidad; aumentando el nivel de participación de personas en riesgo de exclusión en los procesos educativos.

Open source o **Código abierto** es la expresión con la que se conoce al software distribuido y desarrollado libremente. Se focaliza más en los beneficios prácticos (acceso al código fuente) que en cuestiones éticas o de libertad que tanto se destacan en el

PFC. Proyecto Fin de Carrera.

Capítulo 15: Manual de usuario

En esta apartado se explicará las diferentes acciones que puede hacer cada tipo de usuario con la aplicación.

A continuación recordamos de los tipos de usuarios en la plataforma:

▪ **Usuario colaborador**

Sin embargo, si el usuario registrado es un usuario *Colaborador* puede consultar su modelo de usuario y cambiar su contraseña. Si pertenece a una comunidad, ver las noticias, ver los archivos y los foros de esa comunidad.

Un usuario colaborador, puede tener permisos a nivel de los foros como:

- o **Usuario colaborador con permiso Leer:** Los usuarios tienen privilegios mínimos y normalmente no están autorizados para escribir, es decir, no pueden participar en las discusiones de los foros.
- o **Usuario colaborador con permiso comentar** (Leer + Escribir): Los usuarios tienen por lo general menos privilegios dentro de una comunidad, pueden ver el foro y participar en las discusiones.
- o **Usuario colaborador sin permisos:** Los usuarios sin permisos no podrán ver foros ni participar en ellos.
- o **Usuario colaborador Administrador de foros:** Si el usuario registrado es del tipo *Colaborador* y administra Foros, dispone más de las funciones que tiene un usuario Colaborador, tiene funciones como ver, crear, modificar y eliminar los foros de la comunidad donde pertenece.

▪ **Usuario administrador de la comunidad**

Si el usuario registrado es del tipo *administrador* de la comunidad, entonces a nivel de esa comunidad:

- Puede listar los usuarios
- Agregar, quitar usuarios a esa comunidad
- Cambiar permisos a los usuarios de esa comunidad
- Cambiar nombre de la comunidad
- Eliminar comunidad
- Ver/crear foros
- Cambiar permisos a los usuarios respecto al foro
- Alta/Baja pestaña Archivos
- Crear/eliminar carpetas
- Crear/eliminar ficheros
- Modificar perfil general
- Ver/crear noticias
- Eliminar noticias
- **Usuario Administrador General**

El usuario registrado de tipo administrador dispone, más de las funciones que tiene un usuario administrador de una comunidad, de las funciones de dar de alta y de baja a los colaboradores de toda la plataforma, de las comunidades, de las actividades, de las tareas, aprobar usuarios registrados, administrar (listar, aprobar, crear, modificar, eliminar) categorías, agregar usuarios a una comunidad, cambiar permisos a usuarios de una comunidad.

15.1- Registro e inicio de sesión

Paso 1: ingresar a la plataforma CoSpace mediante el siguiente enlace

<http://boppo.udg.edu:8000/colabora3/> (ver figura 26)



Figura 26: vista al acceder a la plataforma CoSpace

Paso 2: ingresar a la opción de registro y diligenciar todos los datos solicitados.

Inclusive Learning CoSpace

✓ Registrate Inicio Ayuda

+ MOSTRAR PREFERENCIAS DE VISUALIZACIÓN

Crea tu cuenta en Colabora

Correo

Contraseña

Confirmar contraseña

Primer nombre

Apellido

Teléfono

Registrarse

Volver

Colabora 2012 Mapa del sitio Terminos y condiciones Ayuda Contáctenos

Figura 27: vista registrar en CoSpace

Paso 3: una vez realizado el registro, se debe esperar a que el administrador del sistema de una aprobación del mismo.



Figura 28: vista registro pendiente de la aprobación

Paso 4: desde la interfaz inicial ir a la opción ingresar. Digitar el correo y contraseña con la cual se realizó el registro.



Figura 29: vista ingresar a CoSpace

Si su registro fue aprobado por el administrador de la plataforma, le deben parecer los iconos que permiten acceder a las funcionalidades de: comunidades, actividades y tareas (figura 30).

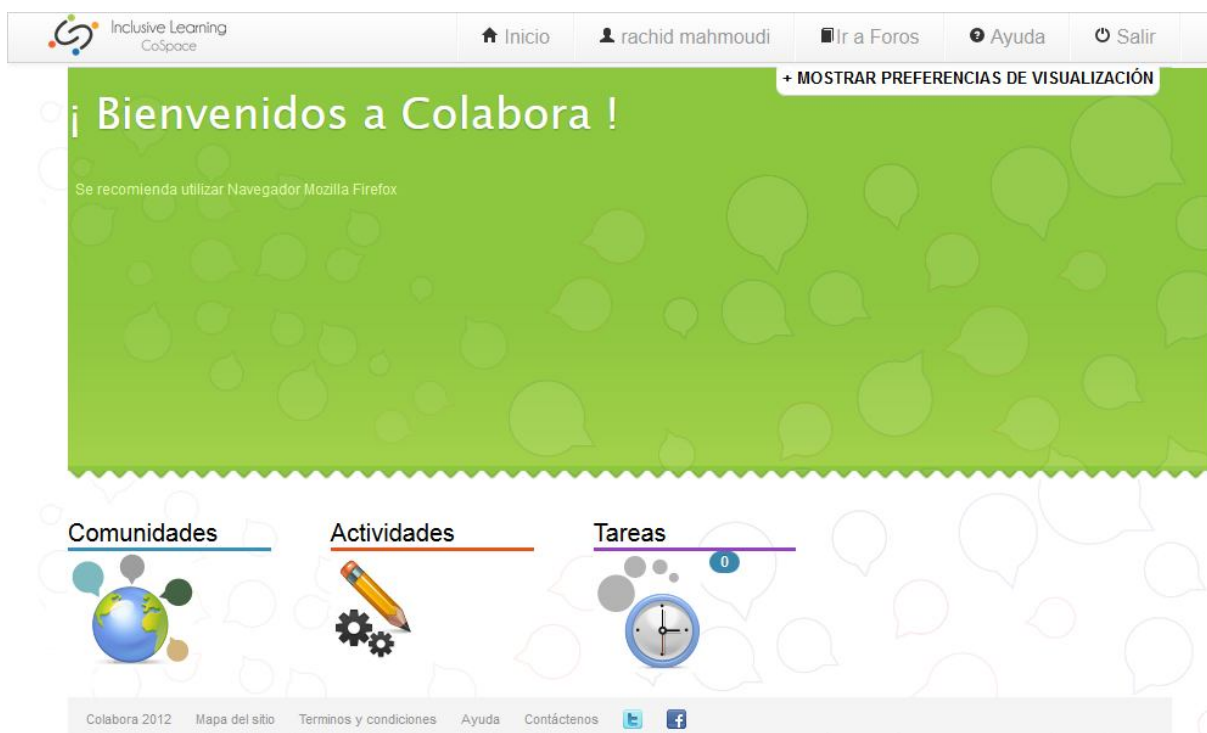


Figura 30: vista inicial después de ingresar al sistema

A partir de la figura 30, empiezan todas las acciones que vemos a continuación.

15.2- Modelo de usuario

La puede acceder cualquier tipo de usuario.

Al entrar al modelo de usuario, hay que acceder al enlace que tiene el nombre de usuario, que se encuentra en la parte superior derecha del enlace que lleva a la página de inicio.

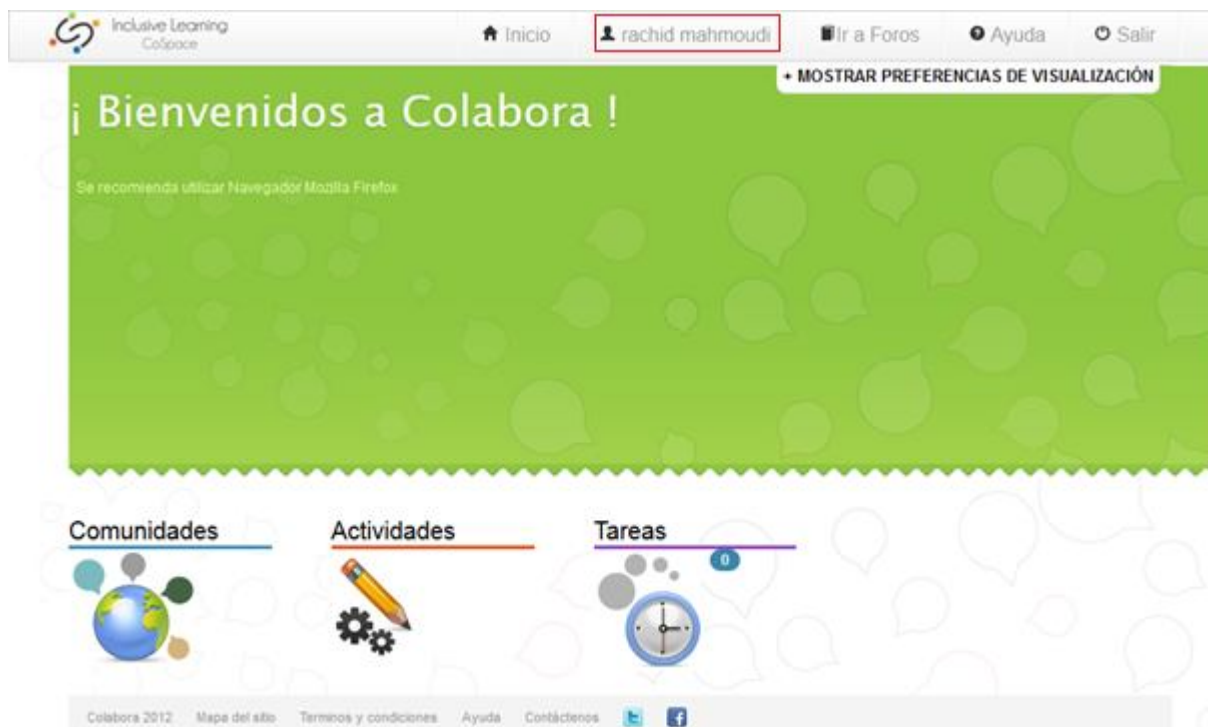


Figura 31: vista inicial después de ingresar al sistema

En esta página (figura 32), aparecen dos opciones para modificar el perfil:

1. Editar tu Perfil
2. Cambiar contraseña
3. Modelo de Usuario



Figura 32: vista al ingresar al link del nombre de usuario

Ingresar a la opción Modelo de usuario

Inclusive Learning CoSpace

Inicio Ir a Foros Ayuda Salir

+ MOSTRAR PREFERENCIAS DE VISUALIZACIÓN

Inicio / Tu Perfil / Modelo de Usuario

Modelo de Usuario

Perfil general

- Perfil general

Perfil alternativa

- Escenarios de desempeño
- Perfil especializado en educación**
 - Perfil General
 - Perfil CAC
 - Perfil CAM
 - Perfil CALE
- Preferencias para el acceso a la información

Perfil de accesibilidad a tecnología de interacción y comunicación

- Condicion sensorial**
 - Perfil limitación atencional
 - Audición
 - Visión
 - Insuficiencia motora de origen cerebral
- Condicion lingüística**
 - Condicion lingüística
- Condicion socio-economica**
 - Condicion socio-economica
- Imagen Perfil**
 - Cambiar foto

Volver

Figura 33: vista del modelo de usuario

A continuación veremos algunos ejemplos de formularios de algunos perfiles.

15.2-1. Escenario de desempeño

Al ingresar al link Escenario de desempeño del Modelo de usuario, nos muestra la figura 34.

Inclusive Learning
CoSpace

Inicio Ir a Foros Ayuda Salir

+ MOSTRAR PREFERENCIAS DE VISUALIZACIÓN

Inicio / Tu Perfil / Modelo de Usuario / Escenarios de desempeño

Modelo de Usuario

Escenarios de desempeño

Por cada escenario de desempeño al que pertenezca debe diligenciar el formulario utilizando el botón "Añadir"

Nombre del escenario	Escenarios de desempeño	Cargo que desempeña en el escenario	Tipo de dedicación en el escenario	Borrar
----------------------	-------------------------	-------------------------------------	------------------------------------	--------

Añadir

Volver

Figura 34: vista del escenario de desempeño

Ahora añadimos un escenario de desempeño.

Clic al botón Añadir, nos muestra el formulario de la figura 35.

Inicio / Tu Perfil / Modelo de Usuario / Escenarios de desempeño / Añadir

Todos los campos marcados con * son obligatorios

Modelo de Usuario

Escenarios de desempeño

Escenarios de desempeño *

☐ Secretarías de educación
☐ Institutos asesores en educación
☐ Universidades
☐ Colegios
☐ Normales
☐ Instituciones no gubernamentales
☐ Otro

Participaciones en el escenario de desempeño:

☐ Estudiantes en procesos de formación
☐ Funcionarios gubernamentales o no gubernamentales
☐ Formadores de profesores y funcionarios Directivos
☐ Coordinadores o diseñadores de programas o currículos
☐ Dirección
☐ Administración
☐ Coordinación
☐ Planeación
☐ Docencia
☐ Investigación
☐ Formación

Tipo de escenario: *

☐ Gubernamental
☐ No gubernamenta

Nombre del escenario: *

Cargo que desempeña en el escenario: *

☐ Ministro
☐ Secretario de educación
☐ Director de departamento
☐ Asesor
☐ Funcionario
☐ Rector
☐ Vicerector
☐ Decano
☐ Director de programa
☐ Coordinador de programa
☐ Profesor
☐ Estudiante

Tipo de dedicación en el escenario: *

☐ Tiempo completo
☐ Medio tiempo
☐ Tiempo parcial

Volver

Figura 35: vista del formulario de escenario de desempeño

Al rellenar el formulario, nos saldrá una notificación según el cas, si falta algún campo o si los datos se han guardado con éxito.

A la siguiente figura 36, como podemos ver el escenario de desempeño agregado, ya tenemos tres opciones con cada escenario de desempeño:

- Modificar
- Borrar
- Consultar

O agregar otro escenario de desempeño.

Inclusive Learning CoSpace

Inicio Ir a Foros Ayuda Salir

+ MOSTRAR PREFERENCIAS DE VISUALIZACIÓN

Inicio / Tu Perfil / Modelo de Usuario / Escenarios de desempeño

Modelo de Usuario

Escenarios de desempeño

Por cada escenario de desempeño al que pertenezca debe diligenciar el formulario utilizando el botón "Añadir"

Nombre del escenario	Escenarios de desempeño	Cargo que desempeña en el escenario	Tipo de dedicación en el escenario	Borrar
Estudiante	Secretarías de educación	Estudiante	Tiempo parcial	BORRAR

Añadir

Volver

Figura 36: vista de escenario de desempeño

15.2-2. Perfil general

En la figura 37, se puede ver el formulario del link *perfil general* del modelo de usuario.

Todos los campos marcados con * son obligatorios

Modelo de Usuario

Perfil general

Ciudad *

País *

Comunidad a la que pertenece: *

Área (Matemáticas, Ciencias, Lenguaje...) *

Nivel de Estudios alcanzado: *

- ☐ Primaria
- ☐ Secundaria
- ☐ Técnica-Tecnológica
- ☐ Universitaria
- ☐ Pos-gradual

Nivel de conocimiento en accesibilidad Web: *

- ☐ Experto
- ☐ Principiante

Aplicación de TIC:

Capacidad de diferenciar navegadores y de identificar ventajas para el trabajo didáctico: *

- ☐ Alta
- ☐ Media
- ☐ Baja |

Conocimiento de programas básicos para el trabajo académico: Windows (Word, Excel, power point), graficadores, etc., que le permitan procesar textos en distintos formatos y con diferentes fines educativos: producción de documentos, diseño de materiales, etc. *

- ☐ Alta
- ☐ Media
- ☐ Baja

Conocimiento y capacidad de búsqueda y de comparación y de selección de opciones de procesadores de textos. *

- ☐ Alta
- ☐ Media
- ☐ Baja

Capacidad de acceder en la web a sitios para el trabajo académico: enciclopedias, bibliotecas, repositorios, foros, bases de datos bibliográficas, propuestas de actividades para el aula y proyectos educativos. *

- ☐ Alta
- ☐ Media
- ☐ Baja

[Volver](#)

Figura 37: vista de escenario de desempeño

15.3- Noticias

Para entrar a la parte de noticias de una comunidad, es a partir de la página de inicio, hay que acceder al enlace que tiene el nombre “Comunidades” o la imagen debajo del nombre, que se encuentra en la parte inferior izquierda de la página de inicio (figura 38).



Figura 38: vista inicial después de ingresar al sistema

Al pulsar el link “Comunidades”, se muestra a continuación la figura 39.

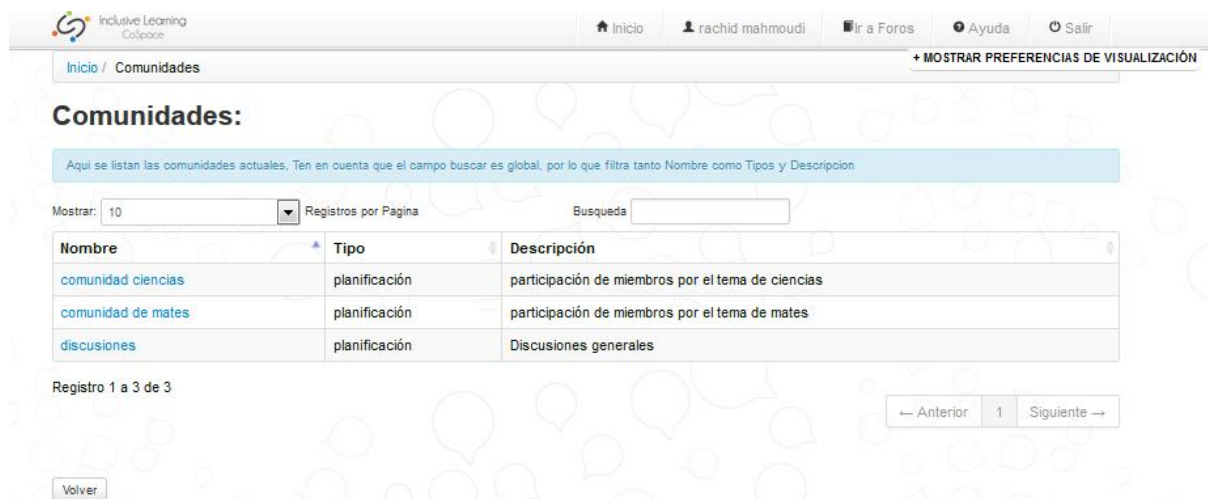


Figura 39: vista al acceder a “Comunidades”

En la figura 39, contiene dos comunidades donde se pertenece el usuario, cada comunidad tiene una vista diferente, la primera donde es el rol de usuario Colaborador (figura 40), y la segunda donde es el rol de usuario como administrador de comunidad (figura 41).

Ahora al pulsar al link de la comunidad “Comunidad ciencias”, se muestra la figura 40.

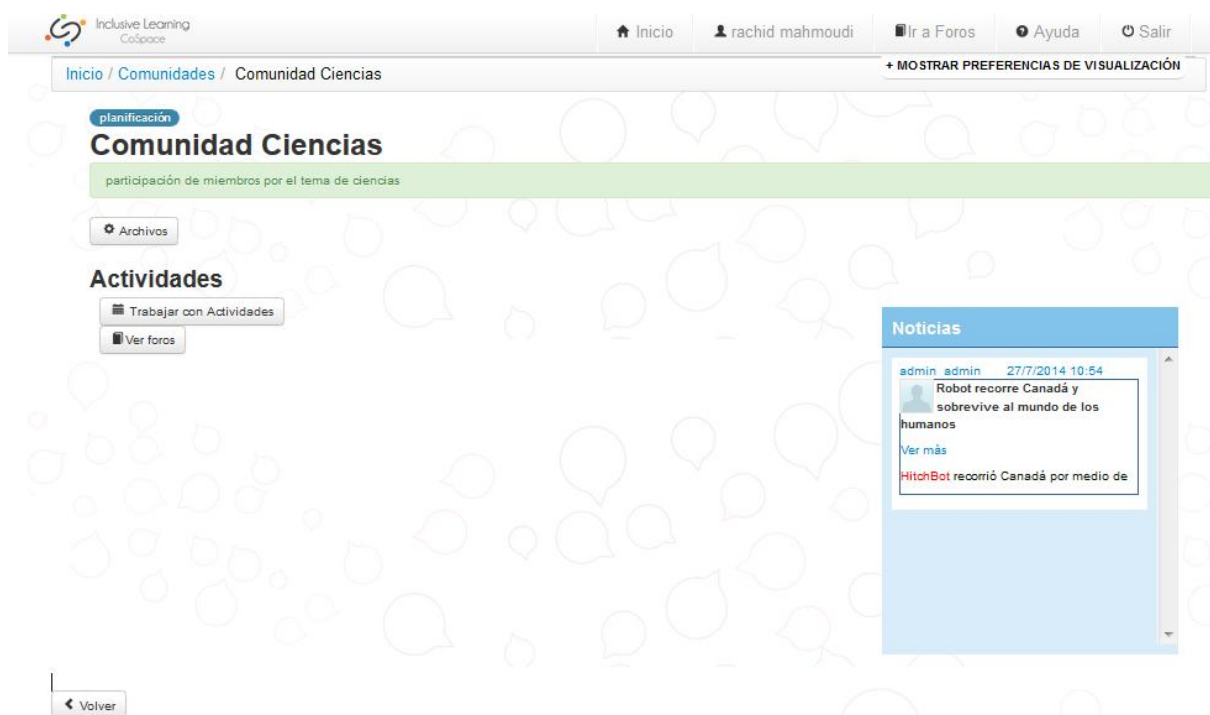


Figura 40: vista al acceder a una comunidad con el rol usuario

En la figura 40 el usuario no podrá crear ni borrar noticias, solamente tiene permiso a ver.

La segunda comunidad donde es el nombre “Comunidad mates”, al pulsar el link se muestra la siguiente figura 41:

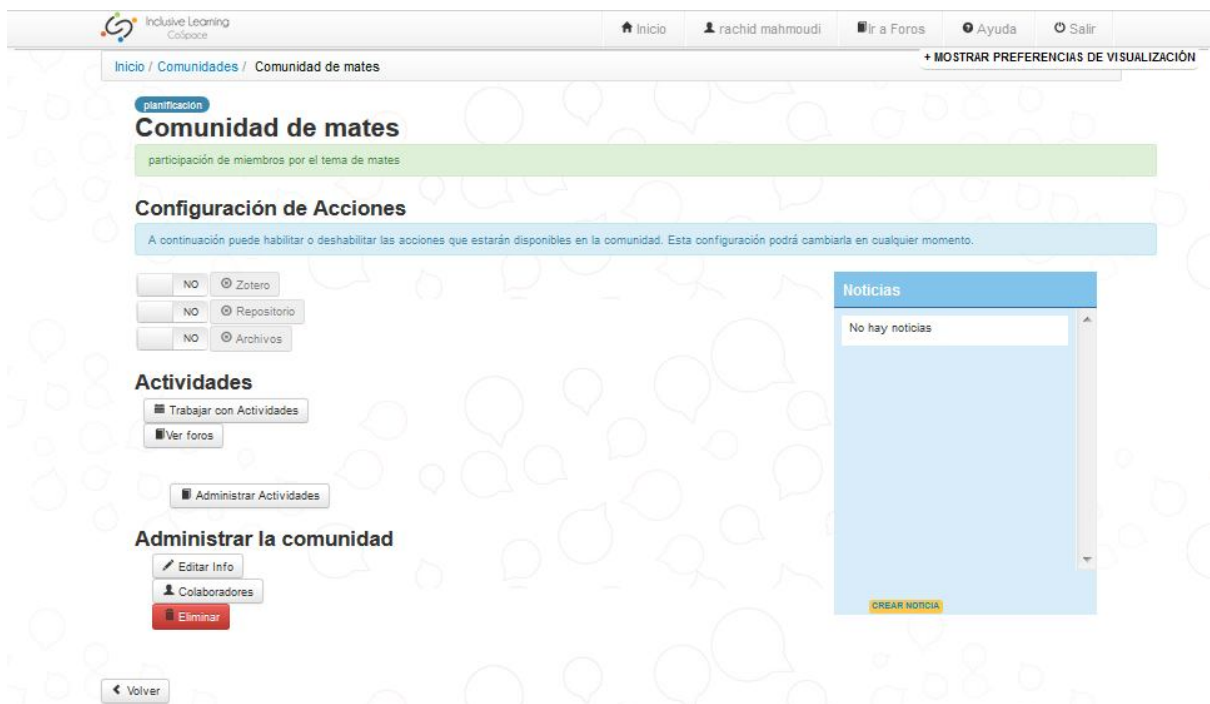


Figura 41: vista al acceder a una comunidad con el rol administrador de comunidad

Para realizar una de las tareas esenciales dentro de una comunidad, pulsamos sobre el botón "Crear Noticia" que vemos en la figura 41.

A continuación nos aparecerá la ventana de crear noticia, donde se podrá redactar el título de el texto de la noticia para después enviarlo (figura 42).

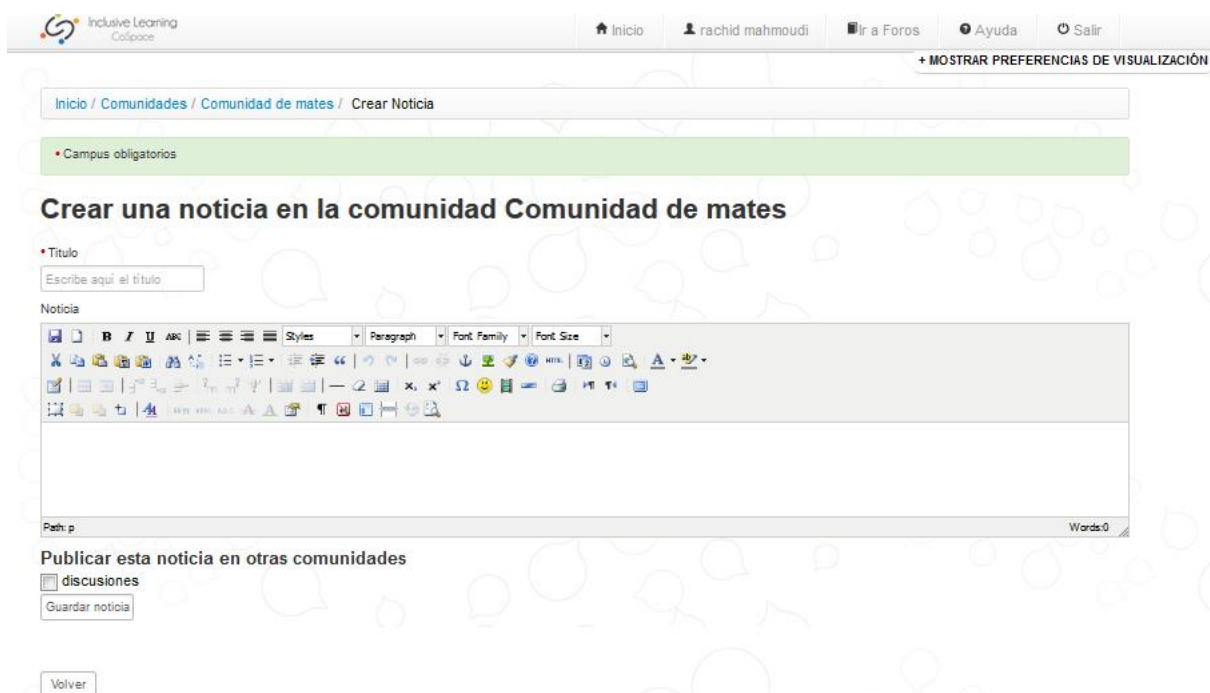


Figura 42: vista crear una noticia

También en la hora de crear una noticia, hay otra opción de publicar esa noticia en otras comunidades donde es ese usuario administrador.

Como se puede apreciar en la imagen anterior (figura 42), publicar una noticia es muy fácil, sólo debemos escribir un título, una descripción y elegir en quinas comunidades queremos publicar y enviar.

Una vez completados estos 3 campos, con pulsar el botón "Guardar noticia", la noticia se publicará en la comunidad como se muestra la figura 43.

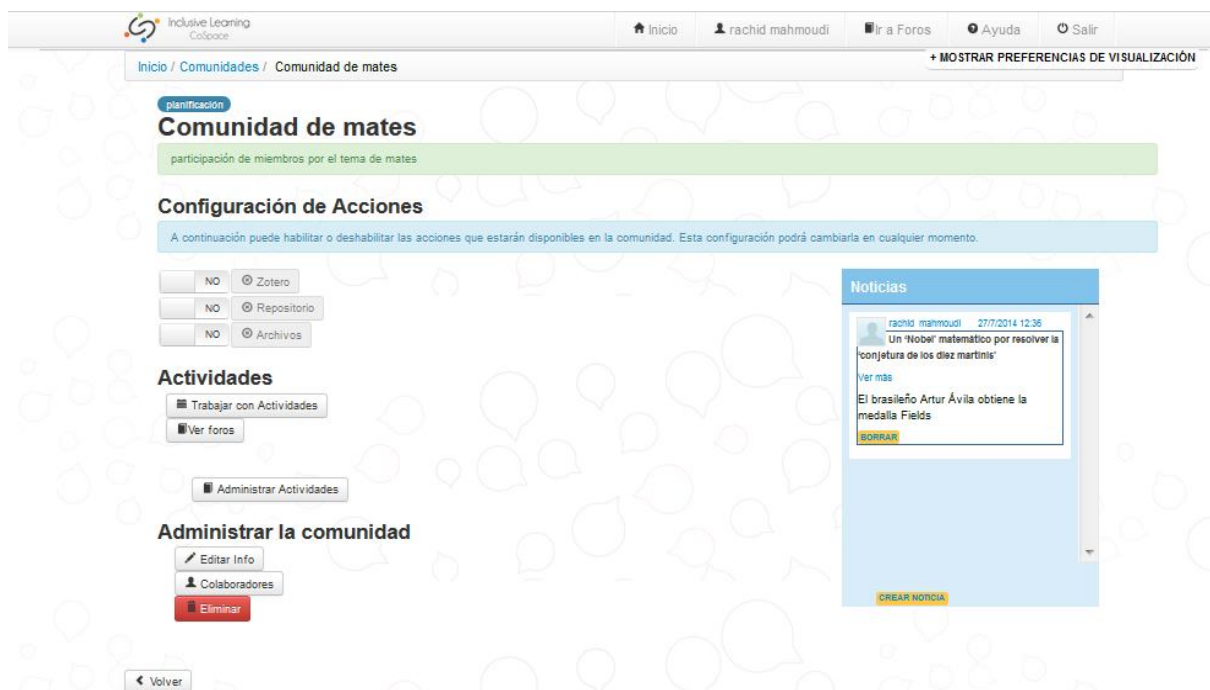


Figura 42: vista de la comunidad con una noticia publicada

Eliminar una noticia

En caso si decidimos eliminar una noticia, debemos pulsar el botón "Borrar" que está debajo de cada noticia.

Una vez pulsado el botón "Eliminar", se mostrará un mensaje de confirmación si se quiere eliminar la noticia.

En caso de pulsar el botón "No", no se elimina, en caso de pulsar "Si" automáticamente el mensaje desaparece de nuestra lista de "noticias".

15.4- Archivos

Una de las funciones básicas dentro de una comunidad para un administrador de comunidad es activar la opción "Archivos" para que todos los tipos de usuarios puedan visualizar, y trabajar con el sistema de gestión de archivos.

Para activar “Archivos” en la página de comunidad pulsamos el botón activar (figura 43)

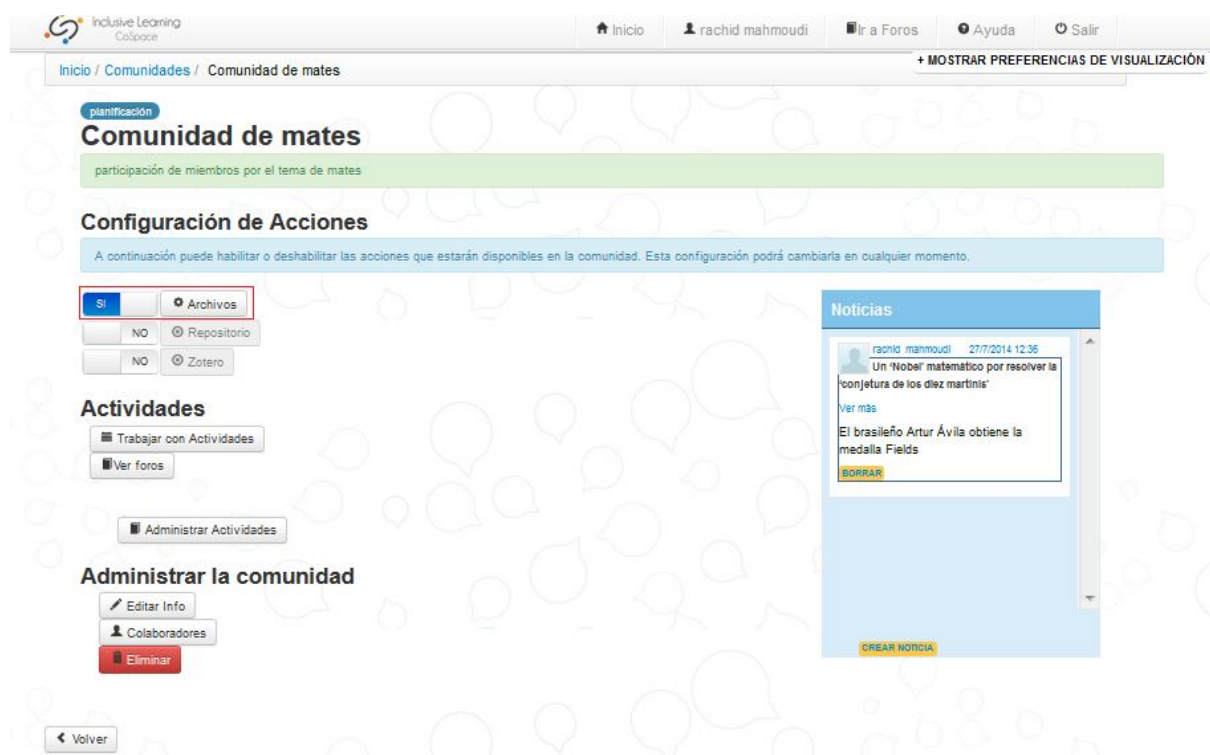


Figura 43: vista de la comunidad con opción “Archivos” activo

Para trabajar con los archivos nos dirigimos a la pestaña de “Archivos” que se puede ver en la figura 43. Al hacerlo, nos dirigirá a otra página como se ve en la figura 44.

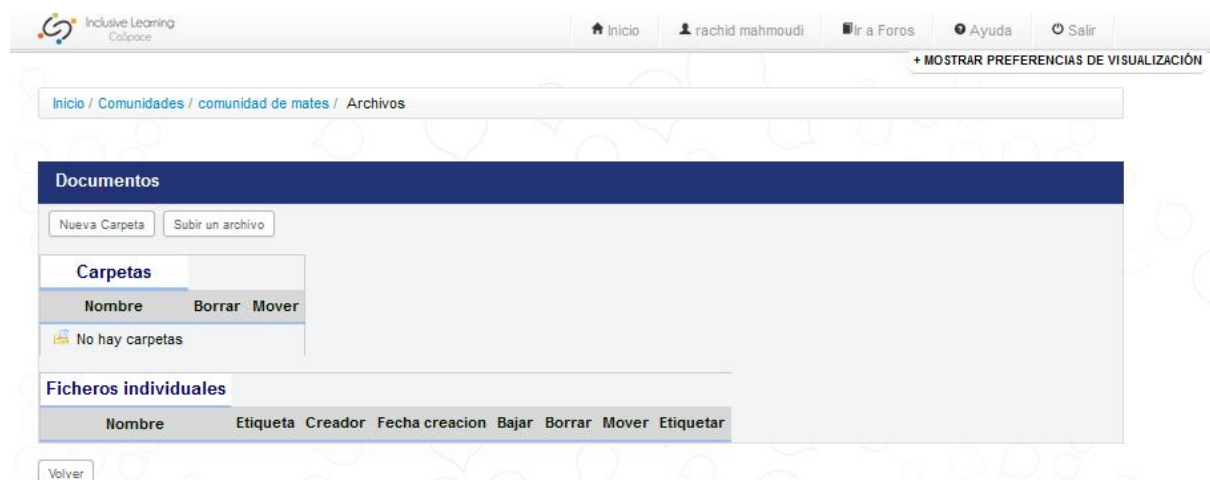


Figura 44: vista “Archivos”

Para crear una carpeta o directorio, basta con pulsar el botón “Nueva Carpeta”, nos muestra la figura 45.

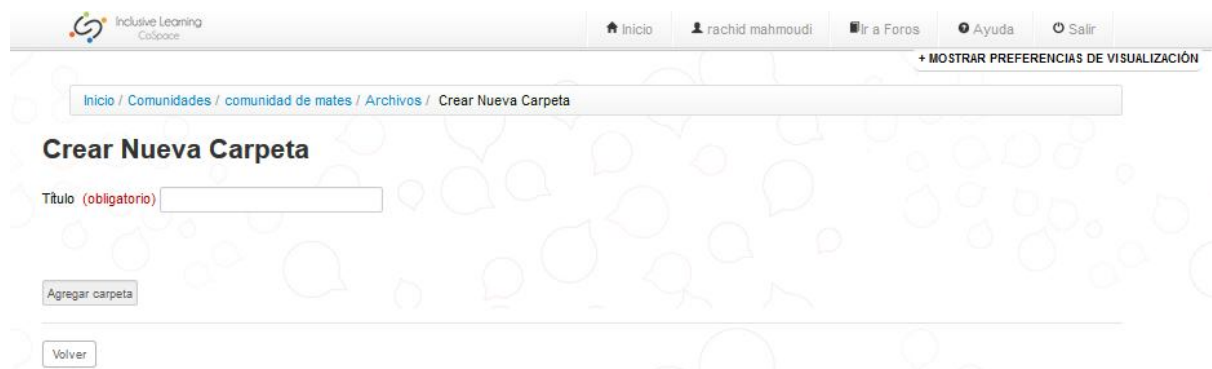


Figura 44: vista Crear nueva carpeta

Una vez dentro de “Archivos” escribimos el nombre de la carpeta, nos muestra la siguiente figura (figura 45).

También podemos agregar fichero mediante el botón “Subir Archivos”.

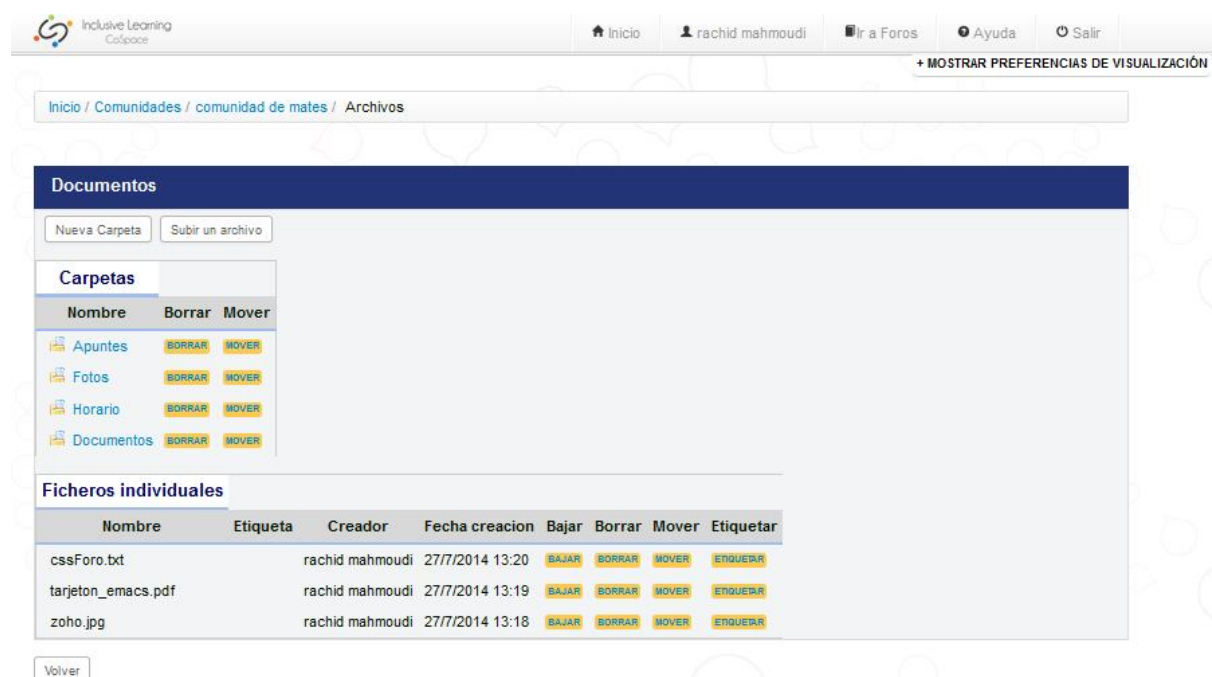


Figura 45: vista “Archivos” con carpeta creada y ficheros subidos

Cada carpeta tiene 3 opciones:

- Acceder a ella y crear más carpetas y subcarpetas y subir ficheros.
- Mover la a otro directorio.
- Borrar

Ahora vamos con el primer opción Accedimos a la carpeta por ejemplo “Apuntes” y creamos dos carpetas “Teoría” y “Practica”, también subimos archivos, y dentro de Apuntes creamos 3 carpetas “Tema 1”, “Tema 2” y “Ejemplos”.

El segundo opción es mover cualquier carpeta a otro directorio (figura 46), en nuestro caso movimos la carpeta “Ejemplos” dentro del directorio “Practica”.

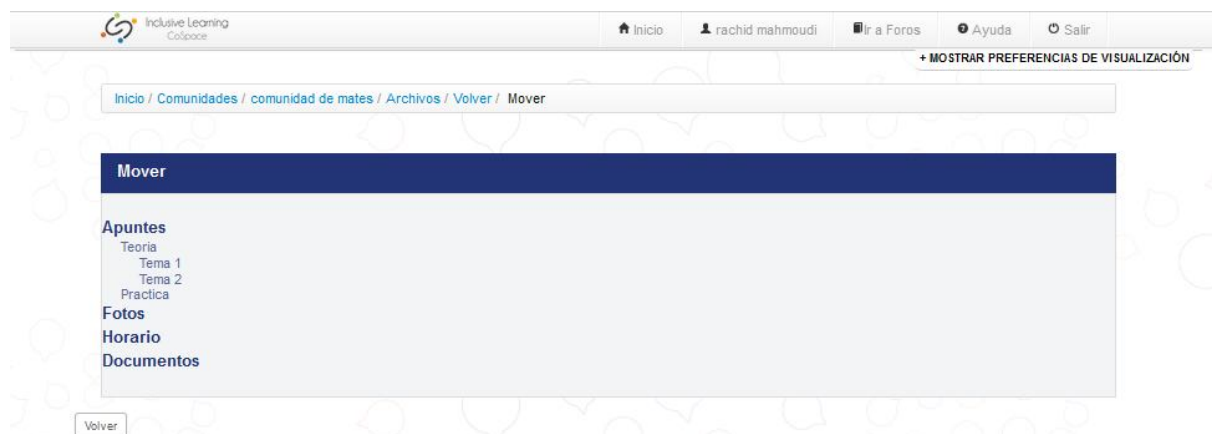


Figura 46: vista Mover una carpeta a otro directorio

Basta con apretar cualquier nombre de carpeta para mover la, la carpeta movida con sus archivos y subcarpetas.

Al final la opción “Borrar”, una vez pulsado el botón "Borrar", se mostrará un mensaje de confirmación si se quiere eliminar la carpeta.

En caso de pulsar el botón "No", no se elimina, en caso de pulsar "Si" automáticamente el mensaje desaparece de nuestra lista de "Archivos" con sus archivos y sus subcarpetas.

Cada fichero tiene 4 opciones:

- Subir
- Mover
- Etiquetar
- Borrar

Al subir un fichero basta con pulsar el botón “Subir un archivo”.

En el caso de mover un archivo es lo mismo que una carpeta, pulsar “Mover” un archivo, nos muestra el árbol de directorios, y elegimos donde queremos mover.

Y para etiquetar un archivo, pulsamos el botón “Etiquetar”, basta con escribir nombre de la etiqueta, y si puede etiquetar un archivo más de una vez, como se muestra la figura 47.

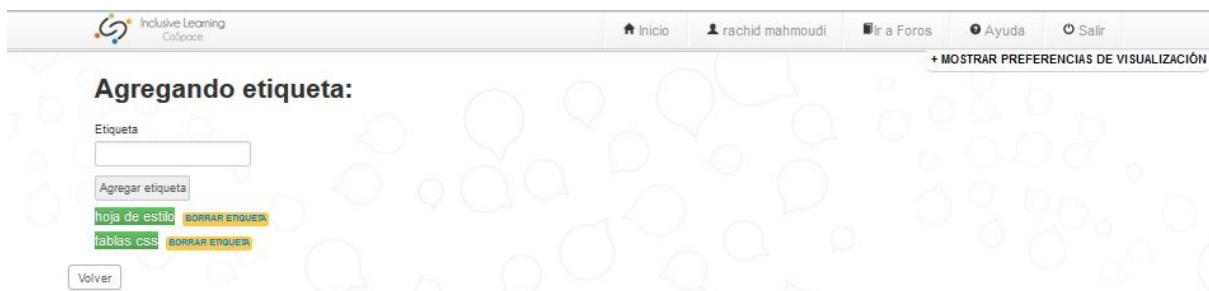


Figura 47: vista Etiquetar un archivo

Como vemos en la figura 47, si puede borrar una etiqueta pulsando el botón “Borrar Etiqueta”.

Finalmente para borrar un archivo, debemos pulsar el botón "Borrar". Una vez pulsado el botón "Borrar", se mostrará un mensaje de confirmación si se quiere eliminar el archivo.

En caso de pulsar el botón "No", no se elimina, en caso de pulsar "Si" automáticamente el archivo desaparece de nuestra lista de "Archivos".

15.5- Gestión de permisos

Al pulsar el botón “Ver foros” (figura 43), nos dirigirá según el permiso de usuario, a la figura 48 el usuario tiene como rol Colaborador, o la figura 49 si tiene rol administrador de comunidad.

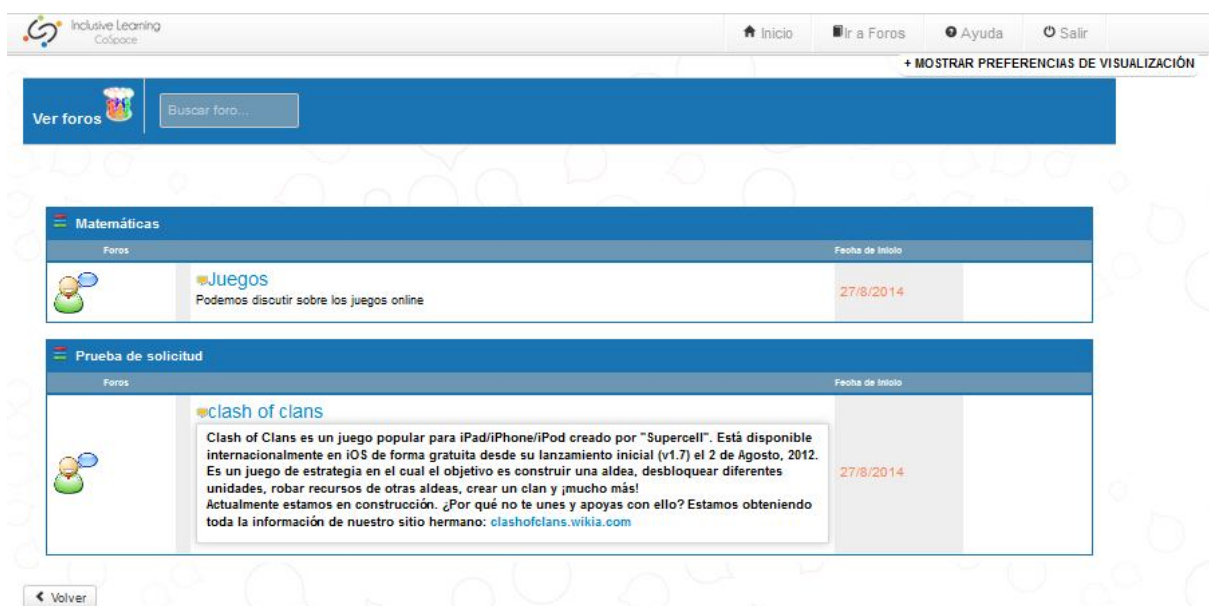


Figura 48: vista de foros de una comunidad con permiso Colaborador

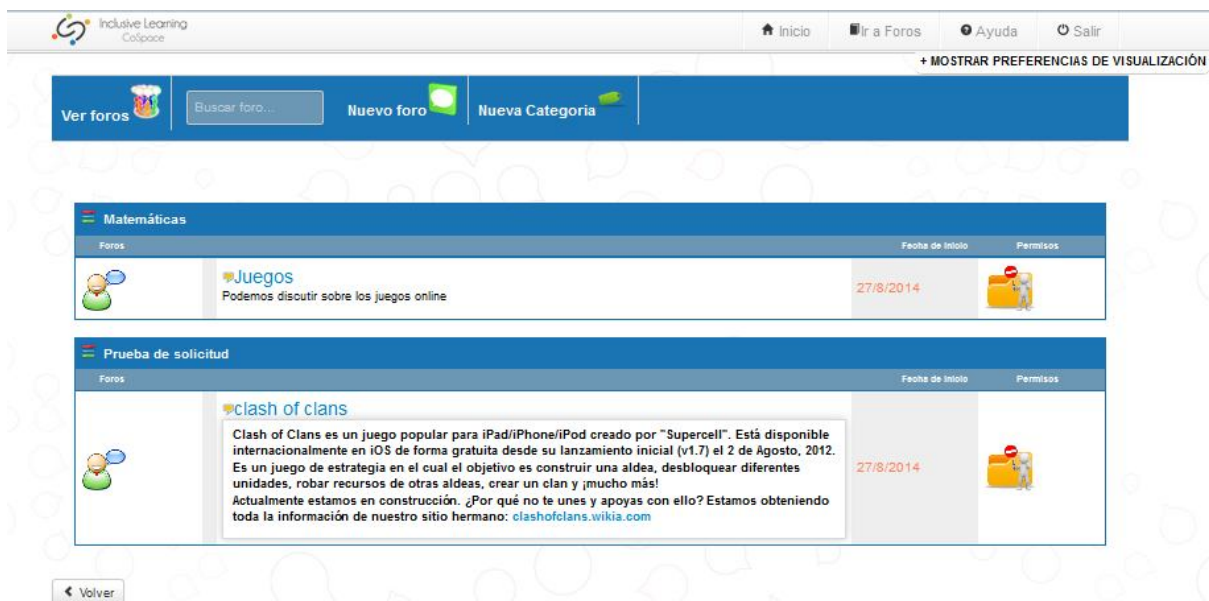


Figura 49: vista de foros de una comunidad con rol administrador

En la figura 49 vemos que un usuario con el rol administrador de una comunidad, tiene opciones de Crear foros, y cambiar Permisos a los usuarios.

Al acceder a la opción Permisos de un foro, nos muestra la figura 50, con todos los usuarios de la comunidad.

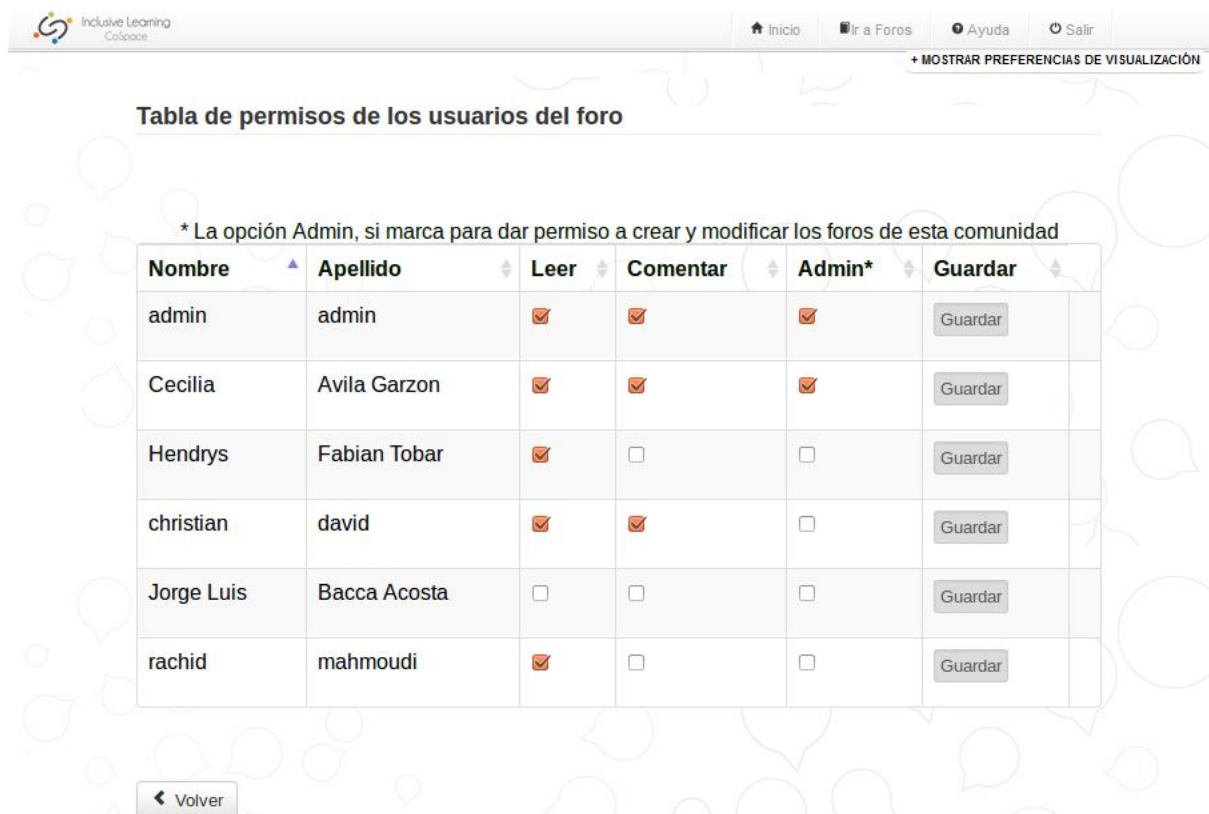


Figura 50: vista de permisos de usuarios de un foro

A continuación mostramos varias figuras según cada permiso.

- Con permiso Leer.

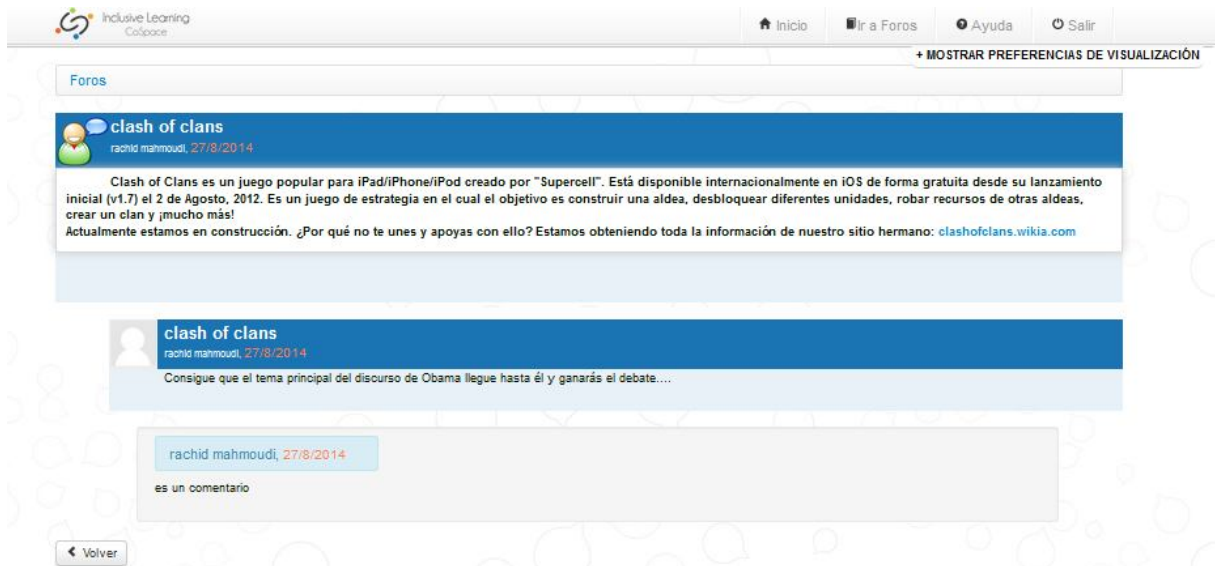


Figura 51: vista de un foro con permiso de leer

En la figura 51, el usuario solo puede ver los foros sin tener permiso de comentar.

- Con permiso escribir

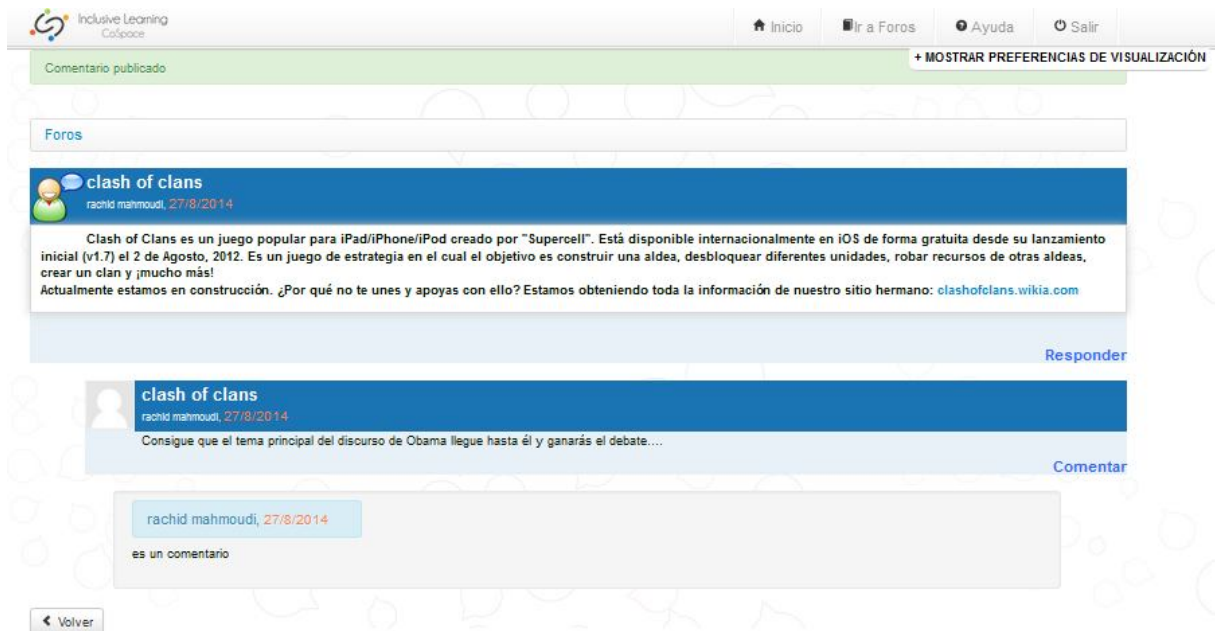


Figura 52: vista de un foro con permiso de escribir

En la figura 52 el usuario tiene permiso de “Responder” o “Comentar” a un foro.

- Con permiso Administrador de foros

Como vemos en la figura 53, que el usuario con rol administrador de comunidad, tiene permiso crear foros, pero no tiene permiso a modificar los permisos de usuarios.

Inclusive Learning CoSpace

Inicio
Ir a Foros
Ayuda
Salir

+ MOSTRAR PREFERENCIAS DE VISUALIZACIÓN

Ver foros
Nuevo foro
Nueva Categoría

Matemáticas

Foros

Fecha de inicio

Juegos

Podemos discutir sobre los juegos online

27/8/2014

Prueba de solicitud

Foros

Fecha de inicio

clash of clans

Clash of Clans es un juego popular para iPad/iPhone/iPod creado por "Supercell". Está disponible internacionalmente en iOS de forma gratuita desde su lanzamiento inicial (v1.7) el 2 de Agosto, 2012. Es un juego de estrategia en el cual el objetivo es construir una aldea, desbloquear diferentes unidades, robar recursos de otras aldeas, crear un clan y ¡mucho más! Actualmente estamos en construcción. ¿Por qué no te unes y apoyas con ello? Estamos obteniendo toda la información de nuestro sitio hermano: clashofclans.wikia.com

27/8/2014

Volver

Figura 53: vista de foros con permiso administrador de foros