

**SINCRONIZACIÓN POR DEMANDA DE ARCHIVOS ASOCIADOS A  
ACTIVIDADES DE USUARIOS EN PLATAFORMAS DE APRENDIZAJE EN  
LÍNEA. CASO DE ESTUDIO MOODLE.**



**UNIVERSIDAD DEL CAUCA**

**Cesar Iván Zambrano Espinosa**

**Juan Manuel Chagüendo Guerrero**

**Universidad del Cauca  
Facultad de Ingeniería Electrónica y Telecomunicaciones  
Línea de Investigación en Informática Educativa  
Departamento de Sistemas  
Grupo I+D en Tecnologías de la Información  
Popayán  
2014**

**SINCRONIZACIÓN POR DEMANDA DE ARCHIVOS ASOCIADOS A  
ACTIVIDADES DE USUARIOS EN PLATAFORMAS DE APRENDIZAJE EN  
LÍNEA. CASO DE ESTUDIO MOODLE.**



**UNIVERSIDAD DEL CAUCA**

**Cesar Iván Zambrano Espinosa**

**Juan Manuel Chagüendo Guerrero**

Trabajo de grado para optar al título de Ingenieros de Sistemas

Director:

**M.Sc. ING. Erwin Meza Vega**

**Universidad del Cauca  
Facultad de Ingeniería Electrónica y Telecomunicaciones  
Línea de Investigación en Informática Educativa  
Departamento de Sistemas  
Grupo I+D en Tecnologías de la Información  
Popayán  
2014**

## AGRADECIMIENTOS

*La meta hoy alcanzada, es el resultado de constantes esfuerzos y sacrificios, tanto personales, como de nuestras familias y personas allegadas, es motivo de orgullo haber logrado culminar este reto y saber que no hubiera sido posible sin la ayuda de ese gran grupo de apoyo que nos ha respaldado, por todo lo anterior, elevo mi agradecimiento a Dios por brindarnos salud, sabiduría y permitirnos estar rodeados de todas esas valiosas personas que han sido elementales para este logro.*

*Agradezco a nuestro director de tesis por toda la asesoría brindada, por sus oportunas apreciaciones y enseñanzas, pero sobre todo por el trato siempre amable y cordial que nos brindó no solo durante la ejecución del presente proyecto, si no durante toda nuestra carrera universitaria.*

*A mi familia por su amor, comprensión y apoyo incondicional, especialmente a mi madre Gloria María Espinosa quien ha sido el motor que me ha impulsado toda mi vida, a mi esposa y compañera de buenas y malas Marcela Vidal, quien transformo mi vida por completo, me ha apoyado en todas las locuras y proyectos que hemos decidido realizar y me dio el mejor regalo posible, Samuel Zambrano, quien llego a nuestras vidas para llenarlas de color y alegría demostrándonos desde que nació que no importa que tan difíciles sean las circunstancias, con amor y sacrificio es posible salir adelante.*

*A mis hermanos que siempre han sido un ejemplo de honestidad y lucha, que me han enseñado la importancia de la unión familiar y el sacrificio por salir adelante, a mi padre por permitirme venir a este mundo, A mi familia Vidal Gámez quienes me acogieron como un miembro más de su familia y me han ofrecido su cariño y apoyo todo el tiempo.*

*A mi compañero de Tesis Juan Manuel Chagüendo, por su trabajo siempre diligente y oportuno, y a Nelsy Hoyos quien nos brindó el apoyo y paciencia necesaria para lograr culminar este proyecto.*

*Agradecer a mis colegas, socios y hermanos de nuestra familia y empresa IDETI quienes nos han brindado su comprensión y apoyo para lograr finalizar esta meta. También a todos mis amigos y compañeros que nos brindaron su apoyo en distintos momentos de nuestra carrera, en especial a los participantes del Focus Group, a Silvana Fernández y Camilo Cerón quienes mostraron su apoyo incondicional en los momentos finales de este proyecto.*

*Finalmente, agradecer a la Facultad de Ingeniería Electrónica y Telecomunicaciones, a la Universidad del Cauca y a todos los ingenieros que nos brindaron sus enseñanzas durante nuestra carrera, en especial a los ingenieros Francisco Pino, Carlos Ardila y Julio Ariel Hurtado quienes colaboraron en actividades de suma importancia en el presente proyecto.*

Cesar Ivan Zambrano

*Hoy al terminar este proceso y culminar esta etapa en mi desarrollo profesional, miro hacia atrás y veo que el camino recorrido ha estado lleno de sacrificios, esfuerzos y satisfacciones, en este momento de orgullo por haber conseguido la meta que algún día me tracé quisieran brindar mis agradecimientos a esas personas que han estado ahí apoyándome.*

*Agradezco a mi señora esposa Nelsy Hoyos quien durante todos estos años de esfuerzo siempre estuvo a mi lado alentandome a siempre seguir para adelante en todas las circunstancias buenas y malas, quien se convirtió en mi apoyo fundamental durante estos últimos años.*

*A mi madre Ana Nalcy Guerrero, mi padre Oswaldo Chagüendo, mi hermano Cesar Eduardo Chagüendo y a toda mi familia que siempre creyeron en mí y me han apoyado para siempre seguir adelante y cumplir con todos los objetivos y proyectos.*

*A mi compañero de tesis Cesar Ivan Zambrano quien decidió embarcarse conmigo en la realización de esté logro, que con su trabajo dedicado permitió que las cosas marcharan de la mejor manera posible, como también agradezco a Marcela Vidal por el apoyo brindado.*

*A mi director de tesis por ser esa guía en el camino, por ser la persona que nos brindó su asesoría, quien puso a nuestra disposición sus conocimientos y siempre la mayor disponibilidad para ayudarnos en los momentos que lo necesitábamos siempre de la forma más cordial y amable posible.*

*A mis colegas y socios de nuestra familia IDETI quienes nos brindaron su apoyo y comprensión en momentos adversos y nos alentaron a seguir adelante y no desfallecer. Además un agradecimiento especial a Juan Camilo Cerón, Silvana Fernández por su apoyo incondicional en las últimas etapas este trabajo.*

*Finalmente un agradecimiento a la Facultad de Ingeniería Electrónica y Telecomunicaciones, a la Universidad del Cauca y a los diferentes ingenieros quienes nos acogieron y nos formaron durante estos años de academia, en especial a los ingenieros Francisco Pino, Carlos Ardila y Julio Ariel Hurtado quienes colaboraron durante la ejecución del proyecto.*

*Juan Manuel Chagüendo Guerrero*

## TABLA DE CONTENIDO

<b>INTRODUCCIÓN.....</b>	<b>10</b>
<b>OBJETIVOS.....</b>	<b>12</b>
OBJETIVO GENERAL.....	12
OBJETIVOS ESPECÍFICOS.....	12
<b>CAPITULO I.    MARCO TEÓRICO.....</b>	<b>13</b>
1.1    SINCRONIZACIÓN DE ARCHIVOS.....	13
1.1.1    Propuestas relacionadas al proceso de sincronización.....	13
1.2    CLOUD COMPUTING.....	23
1.2.1    CLOUD COMPUTING MODELOS DE SERVICIOS Y MECANISMOS DE IMPLEMENTACION.....	25
1.2.1.1    Software as a service (Saas).....	25
1.2.1.2    Platform as a Service (Paas).....	26
1.2.1.3    Infrastructure as a service (IaaS).....	26
1.2.1.4    Cloud Público.....	26
1.2.1.5    Cloud Privado.....	27
1.2.1.6    Cloud Híbrido.....	27
1.3    PLATAFORMAS DE APRENDIZAJE EN LÍNEA.....	27
1.3.1    Entornos Virtuales de Aprendizaje.....	27
1.3.2    Ambientes Virtuales de Aprendizaje - LMS.....	28
1.3.3    Plataforma de aprendizaje en línea.....	31
1.3.4    HERRAMIENTAS Y RECURSOS DE LAS PLATAFORMAS DE APRENDIZAJE EN LINEA.....	31
<b>CAPITULO II.    ADAPTACION DE LA METODOLOGÍA AGIL UP.....</b>	<b>35</b>
2.1    METODOLOGIA AGIL UP.....	35
2.1.1    Disciplinas de AUP.....	36
2.1.2    Fases de AUP.....	37
2.2    ADAPTACION DE LA METODOLOGIA AGIL UP.....	38
2.2.1    Fase de inicio.....	39
2.2.2    Fase de Elaboración.....	42
2.2.3    Fase de Construcción.....	45
2.2.4    Fase de Transición.....	50
<b>CAPITULO III.    FASE DE INICIO.....</b>	<b>53</b>
3.1    ARQUITECTURA CLOUD COMPUTING.....	53
3.1.1    Componentes de Arquitectura del Modelo Cloud Computing.....	53
3.2    DEFINICION DE REQUERIMIENTOS GENERALES.....	55
3.3    PROPUESTA DE ARQUITECTURA DE ALTO NIVEL.....	61
3.4    REQUERIMIENTOS DEL PROTOTIPO.....	63
3.5    SELECCIÓN DE HERRAMIENTAS DE DESARROLLO.....	64
<b>CAPITULO IV.    FASE ELABORACION.....</b>	<b>67</b>
4.1    ACERCAMIENTO INICIAL DE VALIDACION DE ARQUITECTURA PROPUESTA CON EL METODO ATAM.....	67
4.2    ARQUITECTURA ADAPTADA PARA EL PROTOTIPO.....	72

---

4.2.1	Descripción de los casos de uso.....	73
4.2.2	Diseño de prototipos de interfaz de usuario.....	74
4.2.3	Diagrama entidad - relación, diagrama relacional, script de base de datos....	74
<b>CAPITULO V. FASE DE CONTRUCCIÓN Y TRANSICIÓN .....</b>		<b>76</b>
5.1	Fase de Construcción.....	76
5.1.1	PRIMERA ITERACIÓN.....	78
5.1.2	SEGUNDA ITERACIÓN .....	81
5.1.3	TERCERA ITERACIÓN .....	83
5.2	Limitaciones Del Prototipo.....	84
5.3	Pruebas.....	85
5.4	Fase de Transición.....	85
<b>CAPITULO VI. VALIDACIÓN .....</b>		<b>87</b>
6.1	Focus Group .....	87
6.1.1	Fase de planeamiento de la investigación .....	87
6.1.1.1	Definición del problema de investigación .....	87
6.1.1.2	Preparación de materiales y métodos.....	88
6.1.2	Fase de definición de grupos de discusión .....	92
6.1.2.1	Selección de participantes.....	92
6.1.2.2	Segmentación de grupos.....	93
6.1.3	Fase de conducción de la sesión de debate .....	94
6.1.3.1	Captura de Información .....	94
6.1.3.2	Rol del Moderador.....	94
6.1.3.3	Productos de trabajo.....	94
6.1.4	Fase de Análisis de Información y Reporte de Resultados .....	95
6.1.4.1	Análisis de Información.....	95
6.1.4.2	Reporte de Resultados .....	95
6.1.4.3	Análisis de encuesta.....	97
<b>CONCLUSIONES.....</b>		<b>104</b>
<b>RECOMENDACIONES Y TRABAJO FUTURO.....</b>		<b>105</b>
<b>REFERENCIAS .....</b>		<b>106</b>

## LISTA DE FIGURAS

Figura 1. Sincronizador de Archivos - Balasubramaniam y Pierce .....	14
Figura 2. Esquema General Algoritmo Propuesto por S. Balasubramaniam y Benjamín C. Pierce .....	16
Figura 3. Modelo de sincronización resolución de conflictos .....	17
Figura 4. Estados de transición de un sistema de archivos.....	18
Figura 5. Estructura Básica Algoritmo de Actualización de archivos remotos.....	19
Figura 6. Proceso de Deduplicación Genérico. ....	22
Figura 7. Ampliación Proceso de Deduplicación. ....	23
Figura 8. Esquema General Computación en la nube.....	25
Figura 9. Plataforma de Educación en Línea como intersección de los módulos y/o funcionalidades ofrecidas en sistemas e-learning .....	31
Figura 10. ciclo de vida de AUP .....	35
Figura 11. Arquitectura General Cloud Computing.....	53
Figura 12. Arquitectura General del Mecanismo de Sincronización. ....	62
Figura 13. Arquitectura adaptada para el desarrollo del prototipo .....	72
Figura 14. Diagrama de casos de uso usuarios registrados.....	73
Figura 15. Collage de Prototipos de Interfaz .....	74
Figura 16. Versión Final Modelo Entidad - Relación.....	75
Figura 17. ElearningSyncWebClient .....	76
Figura 18. ElearningSyncDesktopClient.....	77
Figura 19. Esquema de archivos Servidor de sincronización .....	79
Figura 20. Esquema de archivos Aplicación Cliente Web .....	80
Figura 21. Esquema de archivos Aplicación Cliente Escritorio.....	82
Figura 22. Código final de declaración del servicio Web y funciones asociadas. ....	84

## LISTA DE TABLAS

Tabla 1. Comparativa LMS en cuanto a Usuarios. ....	29
Tabla 2. Comparativa LMS respecto a Gestión y/o sincronización de Archivos y notificaciones por actualización. ....	29
Tabla 3. Herramientas y recursos comunes en plataformas de aprendizaje en línea. ....	32
Tabla 4. Actividades seleccionadas en la adaptación de la fase de Inicio .....	39
Tabla 5. Actividades seleccionadas en la adaptación de la fase de elaboración .....	42
Tabla 6. Actividades seleccionadas en la adaptación de la fase de construcción .....	45
Tabla 7. Actividades seleccionadas en la adaptación de la fase de transición .....	50
Tabla 8. Resumen de Requisitos .....	56
Tabla 9. Requerimientos que no serán incluidos en el alcance del prototipo .....	63
Tabla 10. Herramientas seleccionadas para el desarrollo del prototipo .....	64
Tabla 11. Árbol de Atributos de Calidad.....	68
Tabla 12. Descripción de Escenario #1 ATAM.....	69
Tabla 13. Estructura de la sesión de Focus Group .....	88
Tabla 14. Posibles Participantes de Focus Group.....	93



## **LISTADO DE ANEXOS**

Anexo 1. Documento de Especificación de Requerimientos Mecanismo.

Anexo 2. Diagramas del Sistema.

Anexo 3. Manual de usuario administrador.

Anexo 4. Manual de usuario.

Anexo 5. Plan de Pruebas Para el Mecanismo

Anexo 6. Artículo. PROPUESTA DE ARQUITECTURA PARA SOPORTAR LA SINCRONIZACIÓN POR DEMANDA DE ARCHIVOS ASOCIADOS A ACTIVIDADES DE USUARIOS EN PLATAFORMAS DE APRENDIZAJE EN LÍNEA.

## INTRODUCCIÓN

En la actualidad se ha generalizado el uso de comunidades virtuales de aprendizaje o plataformas de aprendizaje en línea dentro de diferentes actividades académicas para múltiples usuarios con diferentes intereses. Estas comunidades requieren la gestión de gran cantidad de archivos asociados a diferentes temáticas y actividades que deben estar disponibles para los usuarios. Un estudio preliminar ha permitido observar que la gestión de estos archivos (bien sea: de texto, presentaciones, audio, video, etc.) se realiza en diferentes sistemas de aprendizaje en línea<sup>1</sup> en forma manual, es decir, los usuarios interesados en seguir un curso (estudiantes) deben estar monitoreando constantemente sus archivos y acceder a ellos uno por uno, convirtiendo esto en un proceso poco eficiente cuando existe gran cantidad de elementos agregados a un curso. De forma similar, para los usuarios docentes este es un proceso tedioso, puesto que deben estar gestionando continuamente los archivos en los momentos oportunos con relación al desarrollo de las actividades dentro del curso. Por todo lo anterior es conveniente observar en qué medida este proceso puede ser apoyado para que la tarea de gestionar archivos por parte de los usuarios se pueda realizar en una forma más fluida.

Gracias al desarrollo del modelo de Cloud Computing existen actualmente diferentes herramientas<sup>2</sup> que permiten almacenar todo tipo de recursos en servidores externos, logrando una mayor disponibilidad y facilidades para compartirlos y trabajar en forma colaborativa. Las herramientas existentes ofrecen además aplicaciones que permiten a los usuarios sincronizar los archivos disponibles en sus cuentas en diferentes dispositivos, con la limitante que la sincronización en la mayoría de los casos sólo se puede realizar de una manera total o que no permite mayor grado de gestión del proceso de sincronización. Esto ocasiona con frecuencia que los archivos que no son usados por el usuario de forma activa sigan ocupando espacio dentro de los dispositivos en los cuales se tiene habilitada la sincronización con alguna de estas herramientas.

Partiendo de esta situación problemática, se planteó la siguiente pregunta de investigación ¿Cómo adaptar un mecanismo empleando el modelo de Cloud Computing que permita la gestión de la sincronización de archivos orientada a las actividades y o preferencias de los usuarios de una comunidad virtual de aprendizaje?

La respuesta a esta interrogante fue resuelta en la medida que se abordó la investigación y análisis de información sobre el modelo de Cloud Computing, las plataformas virtuales de aprendizaje y los algoritmos de sincronización. Todo lo anterior permitió realizar la propuesta de arquitectura para el mecanismo de sincronización de archivos por demanda asociados a actividades de usuario en plataformas de aprendizaje en línea, presentada en este trabajo.

---

<sup>1</sup> Dokeos, Claroline, Moodle, etc.

<sup>2</sup> Dropbox, Google Drive, SkyDrive, etc

Una vez definida la propuesta de arquitectura y tomando como caso de estudio la plataforma de aprendizaje en línea Moodle, se procedió a realizar la implementación de un prototipo que lograra como objetivo principal realizar el proceso de sincronización por demanda de archivos, de manera posterior se realizó la validación del mismo por medio del método de Focus Group.

La estructura de este documento está organizada de la siguiente manera: inicialmente se presentan los objetivos del proyecto aprobados por el Consejo de Facultad de Ingeniería Electrónica y Telecomunicaciones. Posteriormente se presentan los diferentes capítulos en los que se muestra el trabajo realizado y para finalizar se muestran las conclusiones y trabajo futuro. A continuación se presenta una breve descripción del contenido de cada capítulo.

Capítulo 1. En este capítulo se encuentra el marco teórico dividido en tres componentes: i) el proceso de sincronización de archivos y las propuestas existentes, ii) el modelo de Cloud Computing y las características del mismo, iii) las plataformas de aprendizaje en línea.

Capítulo 2. Presenta las actividades realizadas en torno al proceso llevado a cabo para la implementación del prototipo, iniciando con una descripción de las características generales de la metodología de desarrollo Ágil UP (AUP) y la adaptación de la misma realizada para la ejecución del prototipo y el planteamiento de la arquitectura propuesta. Posteriormente se abordan los conceptos sobre la arquitectura del modelo de computación en la nube y se establecen los requisitos necesarios para realizar la propuesta de arquitectura, con el listado de requerimientos existentes es posible determinar que dichos requerimientos exceden el tiempo y los recursos que se tienen para dicha implementación por lo que es necesario limitar el alcance del prototipo. Por último este capítulo muestra el proceso de selección de herramientas realizado para la implementación del prototipo.

Capítulo 3. Este capítulo contiene la información relacionada a las actividades ejecutadas durante la fase de elaboración de la metodología AUP, centrándose en el diseño de la arquitectura del prototipo.

Capítulo 4. En este capítulo se abordan las fases de construcción y transición de AUP, para lo cual se muestra los objetivos logrados dentro de cada una de las tres iteraciones realizadas en esta fase, posteriormente se definen las limitaciones del prototipo desarrollado y las actividades ejecutadas durante la fase de transición.

Capítulo 5. Presenta el proceso desarrollado para la planificación de la validación del prototipo implementado en esta propuesta por medio del método de Focus Group, las fases llevadas a cabo, los materiales y recursos de trabajo generados durante el proceso y por último el análisis de la información obtenida por medio de la encuesta realizada de forma individual a los participantes de la sesión y las apreciaciones entregadas por los expertos en la actividad de debate grupal realizada.

## **OBJETIVOS**

A continuación se presentan los objetivos del proyecto, tal como fueron aprobados en el anteproyecto por el Consejo de la Facultad de Ingeniería Electrónica y Telecomunicaciones.

### ***OBJETIVO GENERAL***

Proponer un mecanismo basado en el modelo Cloud Computing que permita realizar la sincronización por demanda de archivos asociados a actividades de usuarios en plataformas de aprendizaje en línea, caso de estudio Moodle.

### ***OBJETIVOS ESPECÍFICOS***

- Proponer<sup>3</sup> una arquitectura fundamentada en el modelo de Cloud Computing, que permita sincronizar por demanda de los archivos asociados a actividades por los usuarios en plataformas de aprendizaje en línea.
- Implementar la arquitectura propuesta en un prototipo de sistema para la sincronización por demanda de archivos asociados a actividades de usuario.
- Validar el prototipo mediante su integración con una de las plataformas Moodle de la Universidad del Cauca<sup>4</sup>.

---

<sup>3</sup> Se emplea el término proponer en referencia a que se deberá adaptar o diseñar la arquitectura adecuada para el mecanismo propuesto, según lo encontrado en el desarrollo del mismo proyecto.

<sup>4</sup> La validación del prototipo se realizara específicamente por medio de un curso en la plataforma Moodle.

## CAPITULO I. MARCO TEÓRICO

### 1.1 SINCRONIZACIÓN DE ARCHIVOS

La sincronización de archivos es el proceso encargado de mantener de forma consistente y coherente las réplicas de un sistema de archivos, actualizadas en distintos dispositivos y/o ubicaciones. El principal reto de este proceso consiste en gestionar las réplicas conflictivas que se pueden generar cuando uno o varios archivos son modificados en el mismo momento por varios usuarios, desde distintos dispositivos. Esto crea la necesidad de implementar una serie de políticas y procedimientos que permitan resolver los conflictos. [1][2]

A nivel general, la sincronización se compone de tres procesos básicos: detección de actualizaciones, reconciliación y resolución de conflictos, los cuales se describen a continuación.

**Detección de actualizaciones:** El proceso de sincronización hace uso de un mecanismo encargado de revisar los cambios generados por los usuario en alguna de las instancias del sistema de archivos, cada vez que un elemento de dicho sistema es modificado este mecanismo se activa detectando las actualizaciones realizadas y preparando la ejecución del proceso de reconciliación.

**Reconciliación:** una vez detectadas las actualizaciones requeridas, este mecanismo se encarga de realizar la propagación de los cambios realizados hacia las demás instancias del sistema de archivos. En este proceso se suelen generar conflictos producidos por cambios simultáneos realizados por distintos usuarios en instancias de un mismo archivo.

**Resolución de conflictos:** cada propuesta de sincronización de archivo realiza de forma particular el proceso de resolución de conflictos, pero en general cuando se detecta un conflicto entre una o más actualizaciones, se elige una actualización con base en una serie de políticas y descarta las demás.

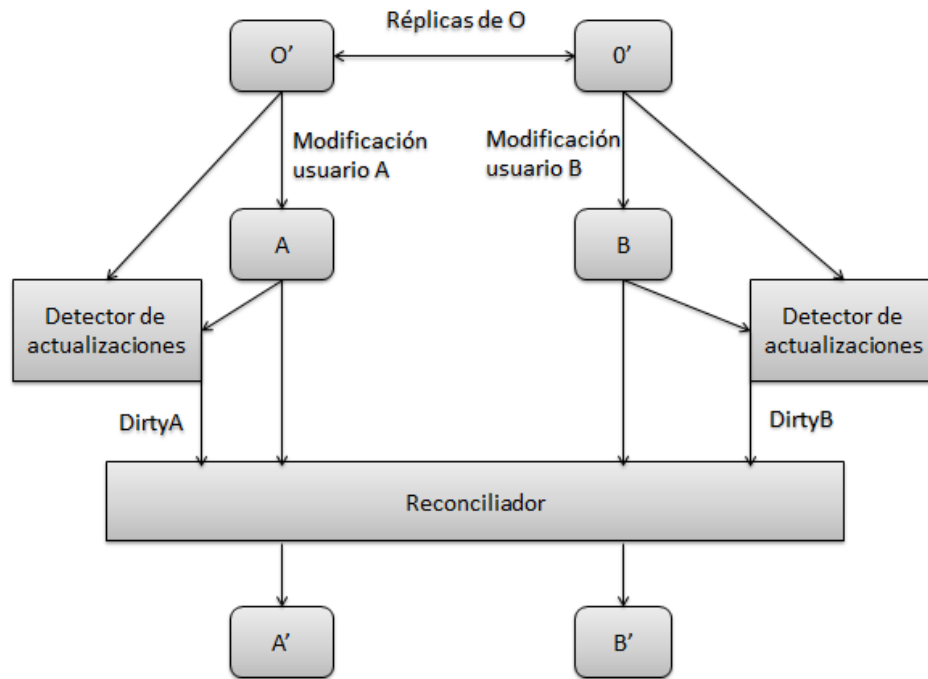
Existen diferentes propuestas enfocadas a optimizar estos procesos, y cada una de ellas agrega una serie de ventajas y desventajas. La mayor diferencia entre ellas radica en la estrategia de reconciliación y resolución de conflictos, donde algunas se enfocan en reemplazar un archivo por su nueva versión (lo que es poco eficiente debido a la cantidad de información replicada innecesariamente cuando los cambios son pequeños) y manejan de forma distinta la resolución de los posibles conflictos encontrados. En la siguiente sección se presentan algunas propuestas estudiadas en el presente trabajo.

#### 1.1.1 Propuestas relacionadas al proceso de sincronización

Dentro de la exploración realizada es posible identificar algunas propuestas referentes a la forma de realizar el proceso de sincronización:

### 1.1.1.1 Propuesta inicial de Balasubramaniam y Pierce

La primera propuesta que se estudio es la presentada por S. Balasubramaniam y Benjamín C. Pierce [3] en la que se describe el modelo de sincronización presentado en la Figura 1.



**Figura 1. Sincronizador de Archivos - Balasubramaniam y Pierce**  
Adaptado de [3]

En el esquema anterior se representa inicialmente el estado inicial de un archivo ( $O$ ), que es replicado en dispositivos accedidos por diferentes usuarios como  $O'$ . Cuando un usuario modifica la réplica existente en su dispositivo, se genera una versión para ese usuario, en este caso el usuario A y B modifican sus réplicas y se generan las versiones A y B del archivo  $O$ . Posteriormente se activa el detector de actualizaciones, el cual identifica los cambios desde el estado inicial  $O$  y genera un listado de cambios a aplicar que se pueden identificar como *DirtyA* y *DirtyB*. Las Versiones A y B y los archivos *DirtyA* y *DirtyB* son tomados para la realización de los proceso de reconciliación y resolución de conflictos, que se analizarán más adelante.

#### Detección de actualizaciones

En cuanto a la detección de actualizaciones presente en la propuesta de S. Balasubramaniam y Benjamín C. Pierce [3], los autores proponen los siguientes mecanismos:

- Detector de actualización trivial

Es la implementación de detección de actualizaciones más simple, la cual se basa en el cumplimiento de un predicado o una condición que evalúa los cambios en un archivo. Cuando se cumple dicho predicado, el detector toma la instancia del archivo y cambia su estado a *dirty* (sucio), haciendo que el reconciliador tome dicha instancia como conflictiva y tenga que comparar contra el resto de réplicas o instancias existentes del mismo archivo.

- Detector exacto de actualización

Este mecanismo de detección toma como base una copia del estado de la última actualización de un archivo que se encuentra ubicada en el servidor de sincronización, y la compara contra cada instancia de este archivo en cualquier dispositivo, buscando encontrar los fragmentos que han sufrido cambios. Esto permite una detección más precisa de la actualización, con la desventaja que aumenta el consumo de los recursos debido a la utilización de mayor espacio en disco y el incremento de tiempo de ejecución debido a la comparación constante y completa entre la copia del estado del archivo y la última actualización.

- Detector de actualizaciones simple por *modtime*

Este tipo de mecanismo realiza la detección de cambios en los archivos de una forma más eficiente en cuanto a tiempo de ejecución, pero menos precisa en cuanto a la detección real de actualizaciones, debido a que en este modelo el criterio para detectar una actualización es la diferencia entre las horas de modificación (*modtime*) de las réplicas de un archivo y la última replica actualizada en el servidor. Este mecanismo se torna impreciso debido a que en determinados casos las fechas de modificación de las instancias de los archivos pueden presentar incongruencias debido a las diferentes configuraciones de horario, y a los manejos en la información de fechas de los archivos de cada sistema operativo, lo que causaría comparaciones erróneas y posibles conflictos que no deberían existir en el proceso de reconciliación.

- Detector de actualizaciones por *modtime* y número de *l-nodo*

Debido a que la representación del sistema de archivos usado por los autores es un árbol, donde cada archivo se representa como un nodo, se cuenta con un número de identificación por nodo (*l-nodo*) que se calcula a partir del contenido de cada archivo. De esta manera, es posible proponer una modificación al detector anterior en el que además de detectar las actualizaciones por la fecha de modificación, se tenga en cuenta el *l-nodo* mejorando la precisión de las detecciones, debido a que especifica un criterio adicional para evitar los posibles conflictos presentados debido a la configuración de fechas. Los autores probaron agregar en vez de un *l-nodo*, un identificador por directorio (*l-dir*), que permitía reducir el almacenamiento de identificadores ya que sería uno solo por varios archivos, pero también reducía la precisión y eficiencia, debido a que si cambiaba un *l-dir* se debía comparar todos los archivos dentro del directorio.

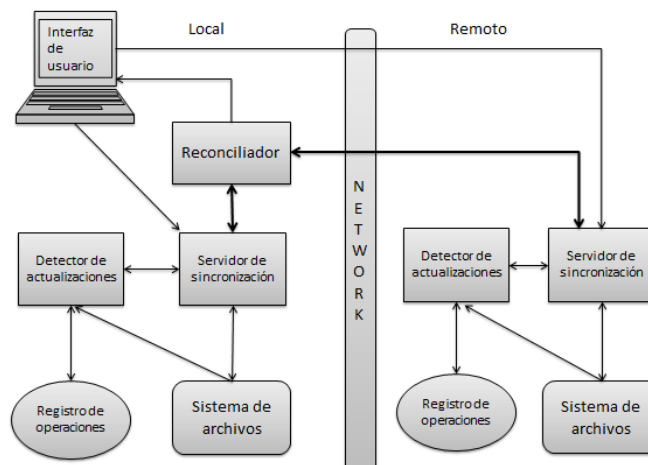
- Detector de actualizaciones en línea

Una propuesta diferente pero que según los autores, en ese entonces era de difícil implementación (sobre todo en sistemas operativos Unix), es la detección de actualizaciones por medio del seguimiento a los cambios realizado al sistema de archivos desde el sistema operativo ante las acciones de los usuarios. Esta propuesta es la empleada por la mayoría de sincronizadores actuales.

## Reconciliación

En cuanto al proceso de reconciliación propuesto, el reconciliador es el encargado de realizar los procesos de propagación de las actualizaciones y el manejo de la existencia de copias conflictivas, definiendo para este proceso una serie de predicados que serán evaluados al momento de propagación de las actualizaciones y la detección de algún conflicto asociado al momento de realizar las propagaciones. Tras realizar todo el proceso de sincronización se obtienen las respectivas copias A' y B' que son las copias sincronizadas de O. En esta propuesta no se profundiza en el mecanismo para detección y resolución de conflictos, simplemente se establece que si no existen conflictos, se procede a realizar la propagación de las actualizaciones. Si existen conflictos no se realiza ninguna operación de propagación.

Entre las características destacables del algoritmo desarrollado en esta propuesta se puede decir, que el flujo de información para el proceso de sincronización es bidireccional, es decir que (la detección de actualizaciones se realiza en ambos sentidos), el mecanismo de reconciliación se ejecuta únicamente en el lado cliente y no existen mecanismos de resolución de conflictos. El esquema general de dicho algoritmo se puede visualizar en la Figura 2



**Figura 2. Esquema General Algoritmo Propuesto por S. Balasubramaniam y Benjamín C. Pierce**

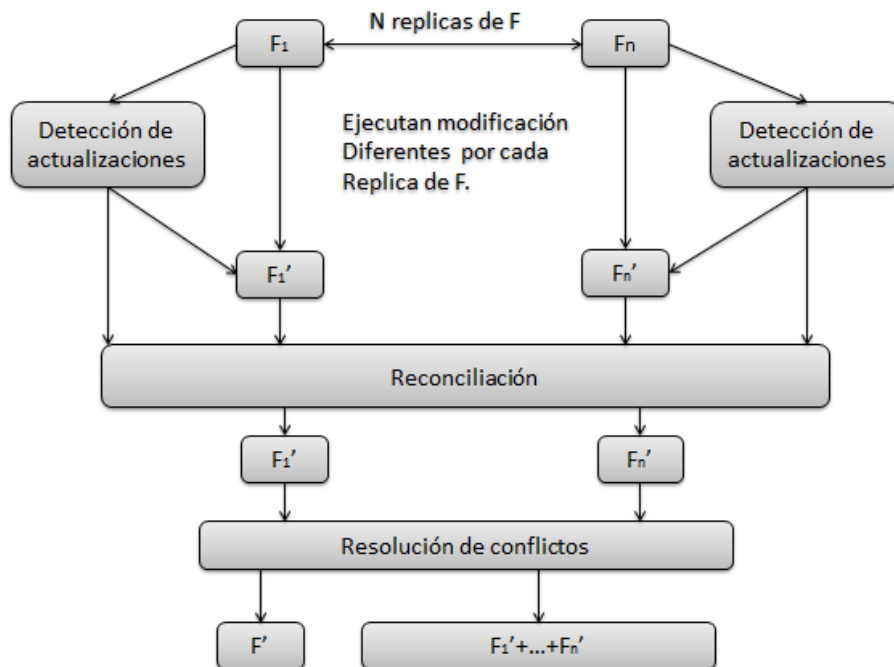
Adaptado de [3]



### 1.1.1.2 Propuesta de Ramsey y Csirmaz

Otra de las propuestas relacionadas con el proceso de sincronización es la presentada por Norman Ramsey y Elod Csirmaz [4] en el cual se plantea una aproximación algebraica para la sincronización de archivos, con la finalidad de establecer pautas generales que pueden ser aplicadas o adaptadas para implementación de un sincronizador de archivos.

Dentro de esta propuesta se plantea que el proceso de sincronización se archivos se debe realizar como se muestra en la Figura 3:



**Figura 3. Modelo de sincronización resolución de conflictos**

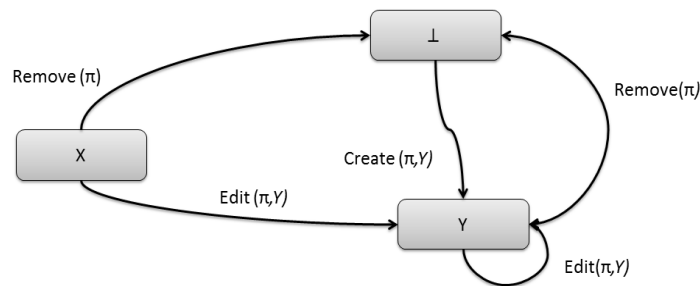
Abstracción propia de [4]

En esta propuesta los autores plantean una aproximación hacia una mecanismo formal para la implementación de un sincronizador de archivos. Dicho mecanismo consta de tres pasos claramente marcados, que a su vez complementan el trabajo expuesto por S. Balasubramaniam y Benjamín C. Pierce [3] en su propuesta. Estos paso se abordan de la siguiente forma: (1) Detección de conflictos, (2) Reconciliación y (3) Resolución de conflictos.

Dentro de esta propuesta se establece una serie de operaciones que se realizan sobre un archivo o un sistema de archivos, abstraídas de modelos mentales. Las operaciones que se consideran para esta propuesta se establecen a continuación.

- create ( $\pi, X$ ): Crear un archivo o un directorio de archivos en  $\pi$ .
- remove ( $\pi$ ): Eliminar el archivo o directorio presente en  $\pi$ .
- rename ( $\pi, n$ ): Renombrar un archivo o directorio presente en  $n$ , dejándolo en la misma jerarquía.
- move ( $\pi, \pi'$ ): Mover de  $\pi$  a  $\pi'$ .
- derive ( $\pi$ ): Cambiar o un archivo o directorio existente, en un camino que puede ser reproducido mecánicamente.
- edit ( $\pi, X$ ): Cambiar un archivo o directorio existente en  $X$ , que no se puede reproducir mecánicamente.

Estas operaciones sobre los archivos representan un flujo de transiciones que puede tener un sistema de archivos, dicho flujo de transiciones se puede observar en la Figura 4.



**Figura 4. Estados de transición de un sistema de archivos**  
Tomado de [4]

Con esto se plantea en esta propuesta una serie de operaciones algebraicas para manejar la sincronización de archivos, permitiendo tener una base teórica adaptable y que cuenta con una serie de pautas para poder dar solución a la resolución de conflictos entre las diferentes copias de un sistema de archivos.

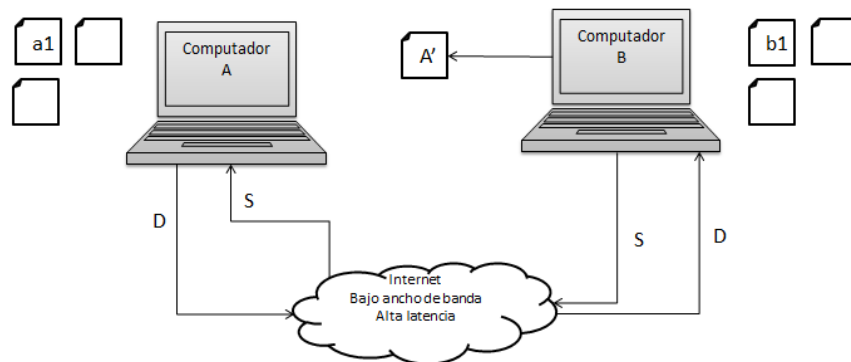
### 1.1.1.3 Rsync

Es una herramienta de código abierto y extensamente usada que emplea el algoritmo diseñado por Andrew Tridgell [5] para actualización eficiente de datos remotos. Dicho algoritmo se diseñó para lograr los siguientes objetivos:

- Trabajar sobre cualquier tipo de dato, no solo texto
- El tamaño del total de datos comprimido debe aproximarse a el tamaño de un archivo de diferencias comprimido
- El algoritmo debe ser rápido para archivos de gran tamaño y grandes colecciones de archivos.

- El algoritmo no debe suponer ningún conocimiento previo de los dos archivos, pero debe aprovechar similitudes, si existen.
- El número de envíos de ida y vueltas debe mantenerse al mínimo para reducir el efecto de la latencia de transmisión.
- El algoritmo debe ser computacionalmente eficiente en relación al uso de recursos.

La estructura básica del algoritmo parte de la existencia de dos réplicas a1 y b1 de un mismo archivo A en 2 computadores A y B conectados por medio de una red de bajo ancho de banda y alta latencia. Primero B envía a A algún dato S basado en los cambios realizados a la réplica b1, el sistema A compara su réplica a1 con el dato S y envía a B algún dato D, B se encarga de construir el nuevo archivo A' basado en b1, S y D. [6][7][8]



**Figura 5. Estructura Básica Algoritmo de Actualización de archivos remotos.**  
Abstracción propia de [5]

Para establecer el algoritmo de actualización de archivos remotos de Rsync se realizaron dos intentos iniciales:

El primero consistía en partir la réplica b1 del archivo A en n bloques de igual tamaño y calcular para los límites de cada bloque una firma, dicho listado de firmas se empleaba como el dato S y se enviaba al sistema A, luego el sistema A divide la réplica a1 en n bloques de igual tamaño, calcula la firma de cada bloque y compara estas firmas con las del dato S. Por último se enviaba hacia B un dato D, compuesto por los bloques que eran iguales para que B construya la nueva versión A'. El problema principal de este algoritmo es que solo encuentra coincidencias en los bloques que se encuentran en los mismos límites, de tal forma que si los cambios realizados en b1 son pocos y son al inicio del archivo, no se encontraran coincidencias en ningún bloque y se envía completo.

El segundo intento se diferencia en la manera de partir el archivo a1 en bloques. En este caso se crean el número de bloques posibles byte por byte y se calcula la firma para cada bloque, logrando que el proceso de detección de cambios sea totalmente exacto. El

problema principal de esta aproximación consiste en el costo computacional que implica calcular firmas fuertes para todos los bloques posibles. Si se intenta reducir el costo computacional calculando una firma que solo dependa de un número pequeño de datos de cada bloque, el resultado sería posibles errores en la detección de actualizaciones por tener firmas débiles. Para mayor claridad se muestra el siguiente ejemplo: si los bloques son de 20 bytes y se calcula la firma de todos los bloques posibles el costo será muy alto. Si con el objetivo de reducir costo computacional se crea la firma con solo los primeros 4 bytes del bloque, la firma sería demasiado débil y todos los bloques que compartan los primeros 4 bytes se tomarían como iguales al compararlos.

La solución final encontrada consiste en emplear la partición de bloques posibles en un archivo (segundo intento) y calcular para cada bloque dos firmas diferentes. La primera firma busca reducir el costo computacional, a pesar de crear una firma débil. La segunda firma (más costosa y más fuerte) se calcularía solo para los bloques de archivos que coincidan en la firma débil para evitar errores de detección. El algoritmo descrito hasta el momento está enfocado en la correcta detección de actualizaciones y es válido aclarar que este algoritmo emplea el protocolo de single round<sup>5</sup> es decir la detección de actualizaciones se realiza en una sola vuelta.

En cuanto al proceso de reconciliación de Rsync, este se realiza de manera unidireccional, es decir el proceso de reconciliación se ejecuta solo en un lado del sistema. Después de finalizado el proceso de reconciliación se retorna la nueva versión del archivo a los clientes.

#### **1.1.1.4 Unisón**

Es una herramienta de sincronización de archivos, que permite la sincronización de múltiples réplicas de un archivo de forma local o remota. Dentro sus consideraciones de diseño se tomó como una elección importante que Unison es una herramienta que trabaja a nivel de usuario, permitiéndole de esta forma acoplarse a diferentes sistemas operativos. Adicionalmente el hecho que Unison trabaje a nivel de usuario implica que la forma en la se comporta está restringida, dado que cuando el sincronizador es iniciado la única información con la que cuenta es el estado actual de cada archivo en cada réplica por lo que se obtiene es la información de la último estado de cada archivo. Esto conlleva

---

<sup>5</sup> Según la investigación realizada se puede definir que existen dos protocolos en los que se basan las propuestas realizadas para mejorar el proceso de sincronización, dichos protocolos son single-round (vuelta simple) o multi-round (múltiples vueltas). El protocolo multi-round hace uso del método de divide y vencerás en forma recursiva sobre esquemas de hash básicos para detectar los cambios realizados en los archivos remotos, mientras que el protocolo single round emplea mecanismos de fragmentación de tamaño fijo (fixed-size) o tamaño variable para comparar los contenidos del sistema de archivos.

a que el sincronizador actué como un conciliador o fusionador de estados actuales de archivos.

Unison plantea un manejo del sistema de archivos basado en árboles donde los nodos internos del árbol son los directorios y los nodos externos (hojas) son los archivos; en la representación de cada nodo del árbol se utiliza los nombres de directorios y/o archivos respectivamente. En este sentido se toma como insumo de entrada: un árbol con los estados iniciales de los archivos y directorios y un árbol adicional que se produce como salida sincronizada. El proceso de sincronización es una función que mapea árboles en pares: el árbol de entrada y el árbol de salida.

En la especificación de Unison se considera una estructura, en la cual en primera instancia se define el núcleo partiendo de la premisa en la que se asumirán los sistemas de archivos como árboles. Se tendrá que para el proceso de sincronización se tomarán los estados del sistema de archivos, por lo que es necesario mantener un archivo que guarde los cambios realizados para poder hacer una comparación posterior en busca de nuevos cambios. Otro aspecto tenido en cuenta es la propagación de cambios, aspecto que se afronta bajo la condición de *que ningún cambio realizado por el usuario debe ser sobre escrito* [9] por lo que se emplea listas de bloques de archivos tanto en las aplicaciones clientes como en el servidor, de tal manera que el proceso de reconciliación se puede realizar de manera bidireccional, por lo tanto el proceso de sincronización se puede dividir en 2 fases: Detección – Actualización y Reconciliación.

La fase de Detección – Actualización se encarga de detectar las actualizaciones generadas en un archivo desde ambos lados desde la última sincronización, luego se intercambian los listados de cambios y se ejecuta en ambos lados el proceso de reconciliación, otra diferencia de esta propuesta respecto a Rsync es que este algoritmo está basado en el protocolo de sincronización multi-round. Vale la pena resaltar que tanto Unison como Rsync son herramientas que centran su esfuerzo en proveer sincronización de archivos por demanda.[7]

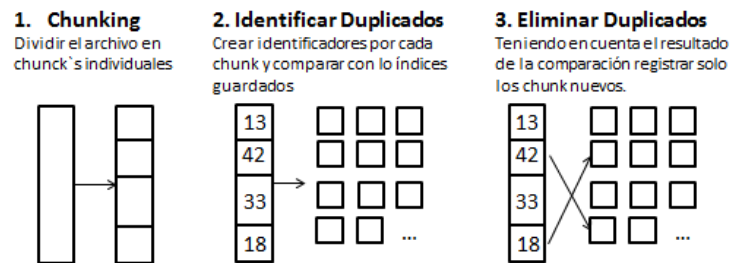
#### **1.1.1.5 Dropbox**

Dropbox [10] es una herramienta popular entre usuarios finales a los que les proporciona un espacio de almacenamiento gratuito inicial de 2 GB<sup>6</sup>, a la vez que emplea Amazon S3 como plataforma de almacenamiento de archivos, Está disponible para múltiples plataformas, con las cuales se integra por medio de una aplicación de escritorio que opera en conjunto con el administrador de archivos. Permite sincronizar los archivos de un usuario en múltiples dispositivos de forma simultánea y ofrece adicionalmente la posibilidad de acceder dichos archivos en dispositivos no sincronizados a través de una aplicación web.

---

<sup>6</sup> El espacio gratuito de Dropbox puede ser ampliado mediante la realización diferentes acciones, por ejemplo por medio de invitaciones a nuevos usuarios.

La detección de cambios que realiza Dropbox está basada en el algoritmo *Rsync-like chunking*, que se encarga de verificar los cambios realizados en el sistema de archivos en un dispositivo. Posteriormente divide dichos cambios en *partes* o *chunks*, realiza un proceso de cifrado de datos para aumentar la seguridad de la información y replica los *chunks* cifrados hacia el servidor en forma de diferencia binaria [11]. De esta forma se logra reducir el ancho de banda necesario para la ejecución exitosa del proceso de sincronización. En el servidor se ejecuta el proceso de reconciliación por medio del algoritmo *live synchronization or continuous reconciliation* que se encarga de realizar la sincronización y detección de errores de manera continua lo que permite ofrecer sincronización por demanda. En lo relacionado al almacenamiento remoto, Dropbox emplea la técnica denominada *deduplication*, la cual utiliza el registro de *chunks* para identificar cuando dos partes o más del archivo son idénticas. En dicho caso, se registra la parte repetida una sola vez y se relaciona a las partes a las que está asociado, disminuyendo de esta manera la cantidad de información en el servidor. A continuación se muestra una adaptación del proceso de deduplicación genérico.



**Figura 6. Proceso de Deduplicación Genérico.**

Adaptado de [7]

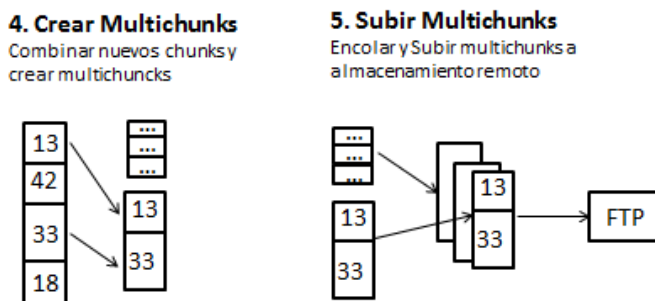
Es importante resaltar que Dropbox ofrece a sus usuarios la posibilidad de configurar una sincronización selectiva, en la cual el usuario puede escoger una sincronización diferente para cada dispositivo definiendo las carpetas y subcarpetas a sincronizar (incluyendo las carpetas compartidas por otros usuarios).

### 1.1.1.6 Syncany

El algoritmo de sincronización propuesto para Syncany [7] está fuertemente relacionado con el concepto de deduplicación descrito anteriormente, este algoritmo se diferencia de las propuestas de Unison y Rsync en la detección de cambios en los archivos. Mientras Unison y Rsync emplean un listado de bloques de archivos global a todos los clientes que utilizan un archivo, Syncany emplea *trace logs* (historias de archivo) individuales en cada cliente, lo que permite reconocer posibles conflictos en las actualizaciones e identificar como solucionar dicho conflicto.

En el proceso de sincronización se comparan las historias de archivo de todos los clientes y se detecta si hay conflictos. Dicha comparación se realiza basada en los identificadores, las versiones de los archivos, el valor de checksums<sup>7</sup> y los registros de tiempo local. Si se encuentra un conflicto, el algoritmo identifica después de la comparación un ganador y en ese caso la versión ganadora es restaurada en el repositorio y la versión con conflicto es publicada en el repositorio con un nombre diferente.

Syncany propone mejorar el almacenamiento de datos y la cantidad de peticiones al servidor por medio de una ampliación del proceso de deduplicación, asociada a la técnica de partición de bloque de *multichunking*. Esta ampliación consiste en aumentar dos pasos nuevos al proceso de deduplicación genérico presentado en la Figura 6, el paso 4 busca crear *multichunks* combinando los chunks que deben registrarse y el paso 5 se encarga de encolar y enviar por la red solo los multichunks creados. La Figura 7 muestra una adaptación que permite mostrar cómo se amplía el proceso de deduplicación.



**Figura 7. Ampliación Proceso de Deduplicación.**  
Adaptado de [7]

## 1.2 CLOUD COMPUTING

El modelo *cloud computing* (computación en la nube), a pesar de no ser un término muy nuevo, ha cambiado sustancialmente la forma en la que se concibe el universo informático, dando paso hacia un modelo de computación que no depende exclusivamente de uno o una serie de computadores, sino que aprovecha los recursos en teoría ilimitados disponibles a través de Internet.

El Instituto Nacional de Estándares y Tecnología de los Estados Unidos (NIST) define la computación en la nube como: “*Cloud Computing es un modelo para habilitar el acceso a un conjunto de servicios computacionales (e.g. Redes, servidores, almacenamiento, aplicaciones y servicios) de manera conveniente y por demanda, que pueden ser*

<sup>7</sup> valor calculado desde el contenido del archivo, generalmente se utilizan los algoritmos de cifrado Sha1 y MD5 para calcular dicho valor.

*rápidamente provisionados y liberados con un esfuerzo administrativo y una interacción con el proveedor del servicio mínimos.” [12] [13]*

Otra definición complementaria plantea que *“Este modelo emerge como un nuevo paradigma capaz de proporcionar recursos de cálculo y de almacenamiento que, además, resulta especialmente apto para la explotación comercial de las grandes capacidades de cómputo de proveedores de servicios en Internet.” [14]*

Otro aspecto importante de la computación en la nube se ilustra en la siguiente definición [15] *“La nube es un paralelo entre un sistema informático distribuido que consiste en una colección de ordenadores virtualizados interconectado que se provisionan y se presentan de forma dinámica como uno o varios recursos de computaciones unificados y basados en acuerdos de nivel de servicio (SLA) establecido a través de la negociación entre el proveedor de servicios y los consumidores ”*

De otro lado en [16] se plantea: *“Las nubes son una gran piscina de fácil utilización y recursos virtuales accesibles (como las plataformas de hardware, desarrollo y/o servicios). Estos recursos se pueden reconfigurar dinámicamente para ajustar a una carga variable (escala), lo que permite también una utilización óptima de los recursos. Este conjunto de recursos normalmente es explotado por un modelo de pago por uso en el que garantías son ofrecidas por el proveedor de infraestructura mediante personalizado Acuerdos de Nivel de Servicio”*

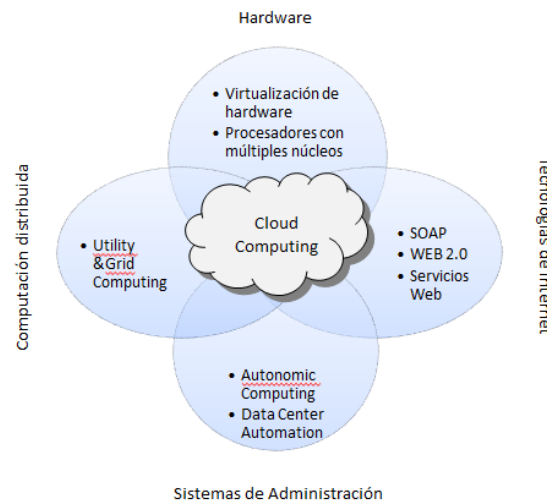
En forma general de las definiciones encontradas de *cloud computing* [17][18][19][20][21] es conveniente establecer que dentro de ellas existen puntos en los cuales la mayoría de ellas convergen estos en particular se podrían establecer de la siguiente forma:

- La computación en la nube es un nuevo paradigma computacional que centra su comercialización dentro de un sistema de pago por demanda o consumo de servicios.
- Cuenta con la capacidad de tener recursos ilimitados (elástica y escalable).
- Permite el acceso a múltiples usuarios.
- Ofrece acceso independiente del dispositivo y la capacidad de manejar autoservicio.
- Tiene la capacidad de abstracción de servicios, virtualización de hardware y capacidades de gran almacenamiento.
- Ofrece la posibilidad de tener acceso a software que permite explotar las diferentes ventajas ofrecidas por los distintos proveedores.

De esta forma es posible establecer que el modelo de computación en la nube, es un paradigma computacional que se enfoca en el aprovechamiento de todos los recursos disponibles en internet. Adicionalmente esta soportado bajo un modelo económico de consumo por demanda de las necesidades del cliente final.

En la Figura 8 se presenta el esquema general de computación en la nube, que describe los elementos asociados al modelo, categorizados en cuatro componentes: (1) Hardware, (2) Tecnologías de internet, (3) Sistemas de administración y (4) Computación Distribuida.





**Figura 8. Esquema General Computación en la nube**  
Tomado de [22]

## 1.2.1 CLOUD COMPUTING MODELOS DE SERVICIOS Y MECANISMOS DE IMPLEMENTACION

Dentro de la computación en la nube se reconocen tres modelos de servicios que brindan características diferentes según las necesidades de los usuarios. Estos modelos se describen a continuación.

### 1.2.1.1 Software as a service (Saas)

El software como servicio es un modelo de distribución de software en la cual el almacenamiento, mantenimiento y operación de las aplicaciones software corresponden al proveedor del servicio y que se brinda en su mayoría a través de internet. Dentro del *cloud computing* este modelo ofrece a los usuarios la posibilidad de adquirir aplicaciones software como un servicio según sea las necesidades. Este modelo está enfocado hacia el cliente final quien solo necesita consumir software para determinados propósitos, bajo demanda específica y permite igualmente a usuario final tener la posibilidad de acceder al software sin realizar grandes inversiones de capital.

Este modelo de computación abre un nuevo campo de posibilidades para la construcción de software capaz de cumplir con las necesidades crecientes del mercado, adaptable y distribuido bajo demanda. Entre sus principales ventajas se pueden distinguir que el modelo Saas permite: menor inversión inicial por parte del cliente para la utilización de software, dado que el cliente no se preocupa por la adquisición de licencias ya que el software se distribuye por la red y se cobra por el servicio (no por el software), no existe necesidad de la inversión en infraestructura para la adquisición de nuevo software, los costes de infraestructura y mantenimiento corren por parte del proveedor del servicio; también permite la escalabilidad dependiendo de las necesidades y la creación de software fácilmente adaptable para diferentes clientes. [12][13][14]

### 1.2.1.2 Platform as a Service (Paas)

Es un modelo de distribución en la cloud computing que se centra en ofrecer al usuario una solución completa para el desarrollo, pruebas, puesta en marcha y mantenimiento de aplicaciones propias, que estarán disponibles a través de internet. El modelo PaaS facilita el despliegue de las aplicaciones construidas librando de los costos derivados de la compra y mantenimiento del hardware necesario, y permite crear aplicaciones completamente accesibles desde internet. Entre los ejemplos principales de este modelo se encuentran Windows Azure, Google App Engine, Amazon SimpleDB, etc. [12][13][14]

### 1.2.1.3 Infrastructure as a service (IaaS)

La infraestructura como servicio es un modelo de cloud computing en la cual el objetivo principal es entregar toda la infraestructura informática (capacidad de computación, espacio de disco, bases de datos, etc.) necesaria para construir entornos informáticos, como un servicio, usualmente mediante sistemas virtualizados. Dentro de este modelo se ofrece a los clientes la posibilidad de adquisición de recursos virtualizados para poder satisfacer sus necesidades. Dichos recursos son fácilmente escalables según la demanda, y el costo de estos servicios es dependiente del proveedor y aumenta por lo general en la medida que el uso de los recursos aumenta. [12][13][14]

Dentro del modelo de Cloud Computing se encuentran diferentes mecanismos para poder hacer la implementación de los diferentes modelos entre los que se pueden distinguir:

### 1.2.1.4 Cloud Público

Es una de las formas de implementación de cloud computing, en el cual un proveedor de servicio se encarga de realizar el despliegue, la administración de la nube y ofrece al público los diferentes modelos de servicios (SaaS, PaaS, IaaS) bajo un esquema de comercialización y accesibles desde internet. Esta forma de implementación supone una serie de características como: [14]

- Los usuarios son visto como un cliente, dado que pagan por el uso de los diferentes servicios para un tiempo determinado.
- Resulta ser una de las mejores opciones dado que no requiere inversiones grandes de capital y no suponen gastos de mantenimiento, además permite un mejor aprovechamiento de la infraestructura del proveedor de servicio permitiendo consumir recursos a la medida de las necesidades, lo que se traduce en un alto grado de escalabilidad y flexibilidad.
- Dado su esquema de comercialización, las nubes públicas son compartidas por muchos clientes dentro de los *data center* de los proveedores de servicio.

### **1.2.1.5 Cloud Privado**

Es una forma de implementación que es realizada por una organización, institución o empresa. Dicha organización se encarga de realizar el despliegue de la nube sobre la infraestructura hardware propia, permitiendo que la solución cloud implementada pueda ser administrada directamente, prestando de forma similar los diferentes modelos de servicio (Saas, Paas, IaaS) pero de forma tal que solo es accesible para quienes hacen parte de la red privada en la cual se realizó el despliegue. Este tipo de implementación se caracteriza por: requerir capital económico para implementar la solución cloud deseada. Por lo general es asociada con sistemas y bases de datos propios, permite el aprovechamiento de infraestructura ya existente, y permite tener un control total sobre la infraestructura y la información, así como de los sistemas asociados a la misma y facilita la adopción de medidas de seguridad acordes a las necesidades propias de la empresa.[14]

### **1.2.1.6 Cloud Híbrido**

Se le denomina al tipo de implementación en la cual se tienen asociadas tipos de implementación como cloud pública y cloud privada, siendo estas entidades únicas pero que están interconectadas por tecnologías que permiten la portabilidad de aplicaciones e información entre ellas, logrando un balance entre las cargas de usuarios y el tráfico que manejan. Dentro de los aspectos más relevantes de este tipo de implementación se pueden mencionar: permite una mayor flexibilidad de los servicios prestados, permite un mayor control sobre la infraestructura y el manejo de datos, implica un mayor esfuerzo al momento de la implementación asociado con la interoperabilidad entre las nubes pertenecientes, y permite explotar de forma más efectiva las diferentes características de cada tipo de implementación.[14]

## **1.3 PLATAFORMAS DE APRENDIZAJE EN LÍNEA**

En la literatura actual es posible encontrar diversas fuentes en las que se proponen diferentes definiciones y conceptos relacionados con plataformas de aprendizaje en línea. Gracias al crecimiento exponencial que en la actualidad tienen las tecnologías de la información, los ambientes virtuales de aprendizaje (Learning Management Systems, LMS) y los Entornos Virtuales de Aprendizaje (EVA) han sido objeto de diversos estudios que establecen diferentes definiciones tanto para LMS como para EVA, a continuación exponemos distintos conceptos encontrados en la literatura.

### **1.3.1 Entornos Virtuales de Aprendizaje**

Según [23] un entorno virtual de aprendizaje “es una aplicación informática diseñada para facilitar la comunicación pedagógica entre los participantes en un proceso educativo, sea éste completamente a distancia, presencial o de naturaleza mixta que combine ambas modalidades en diversas proporciones”

Un EVA también es definido como un sistema de gestión de documentos educativos que incluye diferentes aplicativos o herramientas diseñadas con el fin de permitir la comunicación de forma fácil entre estudiantes-profesores y estudiante-estudiantes, donde además se busca incentivar el desarrollo de actividades colaborativas y brindar a los docentes mecanismos para la evaluación de conocimiento de los estudiantes [24].

Otra definición de EVA lo presenta como una plataforma web que permite acceder en cualquier momento y lugar a toda la información relacionada con un curso, también ofrece diferentes mecanismos de participación a sus usuarios tales como foros, chats, wikis, etc [25].

Para [26] un EVA es un espacio virtual donde estudiantes y docentes pueden realizar las actividades más comunes en la educación presencial, es posible realizar debates, consultar documentos, realizar ejercicios, trabajar en equipo, etc. Según el autor la característica más relevante de un EVA es la fuerte relación con el modelo pedagógico tradicional presencial.

Otro autor define EVA como un espacio de herramientas que integradas permiten la gestión del e-learning, el seguimiento a los estudiantes y el acceso continuo de los estudiantes a los recursos dados por los tutores, los cuales deben ser establecidos cuidadosamente por el tutor de tal manera que sean útiles para el estudiante [27].

Al analizar las definiciones se puede concluir que un Entorno Virtual de Aprendizaje es una plataforma web que permite la comunicación y colaboración de estudiantes y tutores en torno al proceso de aprendizaje, el cual puede ser totalmente virtual, semi-presencial o presencial. De esta manera, un EVA a pesar de surgir y enfocarse en satisfacer la necesidad de educación a distancia, proporciona todas las herramientas necesarias para incluirse en el proceso de aprendizaje presencial como apoyo a las actividades tradicionales entre docentes y estudiantes.

### **1.3.2 Ambientes Virtuales de Aprendizaje - LMS**

Un ambiente virtual de aprendizaje generalmente es una aplicación Web que ofrece a sus usuarios diferentes módulos enfocados a cubrir procesos administrativos y de seguimiento necesarios en un sistema enseñanza-aprendizaje, sus característica principal es servir como un contenedor de cursos y además ofrecer herramientas (tales como agendas, glosarios, foros, chats, videoconferencia, audioconferencia, streaming, etc) que faciliten la comunicación y el trabajo colaborativo logrando de esta manera en el elemento central de comunicación en los sistemas de e-learning [28][29].

Según [28] en los sistemas de gestión de aprendizaje se distinguen los siguientes grupos de funcionalidades: gestión de cursos, gestión de clases, herramientas de comunicación, herramientas para los estudiantes, gestión del contenido, herramientas de evaluación y gestión de la institución educativa.

La principal diferencia entre un LMS y un EVA radica en que el primero solo se enfoca en ofrecer las funcionalidades necesarias para realizar la gestión de los cursos, mientras el

segundo adicionalmente se interesa en la gestión correcta de los recursos que genera el tutor para sus estudiantes y en contribuir con la gestión de diferentes modelos pedagógicos.

En la actualidad existe gran cantidad de sistemas de gestión del aprendizaje, cada uno con diferentes características y funcionalidades, A continuación se muestran dos tablas con información sobre los 5 LMS más populares y usados según la información encontrada en [30], la Tabla 1 busca mostrar los datos de uso de cada una de las plataformas, donde es importante resaltar que Moodle se encuentra ponderada como la plataforma más usada. La Tabla 2 busca describir las características respecto a la gestión de archivos, incluyendo los avances o funciones respecto a la sincronización en cada una de estas plataformas.

**Tabla 1. Comparativa LMS en cuanto a Usuarios.**

	Moodle	Edmodo	BlackBoard	SkillSoft	Schoology
<b>Número Total de Clientes</b>	87.084	120.000	20.000	6.000	10.000
<b>Número Total de Usuarios</b>	73'753.035	20'000.000	20'000.000	19'000.000	7'000.000
<b>Número Total de seguidores en Redes Sociales</b>	37.497	128.537	60.731	30.227	19.498

**Tabla 2. Comparativa LMS respecto a Gestión y/o sincronización de Archivos y notificaciones por actualización.**

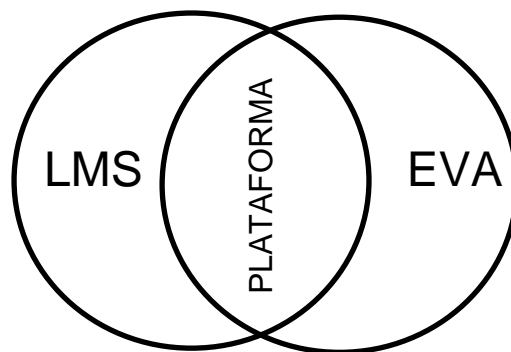
<b>Gestión y/o sincronización de Archivos</b>	
<b>Moodle [31]</b>	<p>A partir de la versión 2.0 es posible usar archivos desde repositorios como Dropbox, Google Drive, etc, pero se limita solo a importar el archivo desde dicho repositorio y la gestión de acceso a los archivos por configuración de acceso en cada repositorio puede ser difícil de lograr.</p> <p>Cuando se produce un cambio en un curso, es posible informar a los participantes del curso por medio de correo electrónico.</p> <p>Desde la versión 2.0 Moodle plantea la sincronización de archivos como trabajo futuro, pero no muestra en sus versiones posteriores ningún avance significativo relacionado.</p>

<b>Gestión y/o sincronización de Archivos</b>	
<b>Edmodo</b> [32][33][34]	La gestión de archivos de esta plataforma consiste en una serie de bibliotecas públicas y privadas en las cuales los usuarios pueden subir archivos o enlaces de archivo, Actualmente solo se soporta la importación de archivos desde el repositorio de Google Drive. No se encontró información relevante respecto a procesos de sincronización.
<b>BlackBoard</b> [35][36][37]	Según la información encontrada de esta plataforma, la gestión de archivos permite subir archivos y gestionarlos en distintas áreas de archivo, tales como, áreas de archivo de un curso, o personales, etc. Permitiendo subir archivos desde el computador desde el que se está accediendo, hasta el momento no se evidencia soporte para importación de archivos de repositorios externos como Dropbox, Google Drive, ni ningún otro, ni se evidencia avances en cuanto al proceso de sincronización.
<b>SkillSoft</b> [38][39]	SkillSoft es una plataforma de educación en línea fundamentada en el modelo de computación en la nube y ofrecida bajo el modelo de software como servicio, que ofrece a sus usuarios la oportunidad de adquirir distintos módulos y funcionalidades según sus necesidades. Este producto tiene como factor diferenciador la posibilidad de ser usado de una manera amigable y cómoda desde dispositivos móviles como tabletas y smartphones, lastimosamente al ser una aplicación comercial no ofrecen información sobre características técnicas y soporte sin tener una cuenta, por lo que no es posible identificar que se ha realizado en torno a la gestión de archivos, integración con repositorios externos o procesos de sincronización de archivos.
<b>Schoology</b> [40][41][42]	<p>LMS basado en el modelo de computación en la nube con grandes aportes en el área de integración con plataformas y repositorios externos, esta plataforma incluye la integración con repositorios tales como Dropbox, Google Drive, Turnintin y LTI content para el acceso a recursos externos.</p> <p>Adicionalmente cuenta con una integración total a la plataforma como servicio de Google, incluyendo mejoras en cuanto al trabajo colaborativo, ya que ofrece a sus usuarios la posibilidad de compartir las actividades a través de Google Calendar y realizar la edición en línea de archivos por varios usuarios al mismo tiempo por medio de las aplicaciones de Google Drive.</p> <p>Esta herramienta cuenta también con opciones enfocadas al aprendizaje móvil, que empieza a ser una nueva tendencia en educación, ofreciendo para esto aplicaciones móviles robustas que permiten al usuario estar al tanto de lo que sucede en la plataforma desde su dispositivo móvil, recibiendo notificaciones cuando se producen actualizaciones en los cursos seguidos por el usuario. No se encuentra información en cuanto a avances en procesos de sincronización de archivos.</p>

### 1.3.3 Plataforma de aprendizaje en línea

Debido a que el termino plataformas de aprendizaje en línea es usado frecuentemente y de manera indistinta en el contexto de diferentes herramientas de e-learning tales como LMS y EVA, es necesario determinar el concepto que se usará en adelante.

Se entenderá como plataforma de aprendizaje en línea el conjunto de módulos y/o funcionalidades ofrecidas en un sistema de e-learning, convirtiéndose de esta manera en el elemento central alrededor del cual es posible establecer un ambiente virtual de aprendizaje o un entorno virtual de aprendizaje, como se muestra en la Figura 9.



**Figura 9. Plataforma de Educación en Línea como intersección de los módulos y/o funcionalidades ofrecidas en sistemas e-learning**

### 1.3.4 HERRAMIENTAS Y RECURSOS DE LAS PLATAFORMAS DE APRENDIZAJE EN LINEA

En las diferentes plataformas existentes es posible encontrar gran variedad de herramientas y/o recursos diseñados para apoyar el proceso de aprendizaje en línea. En [43] se encuentra una clasificación de las diferentes de herramientas que actualmente son ofrecidas por los LMS y CMS, con base en dicha clasificación, a continuación se presenta en la Tabla 3 un resumen de las herramientas y recursos más comunes en la mayoría de plataformas de plataformas de educación en línea.

**Tabla 3. Herramientas y recursos comunes en plataformas de aprendizaje en línea.**

Adaptado de [44]

Categoría	Recurso	Descripción Recurso
Herramientas de Comunicación	Foros de Discusión	Herramienta que permite a los participantes de un curso intercambiar mensajes, en torno a un tema en particular de interés para los participantes.
	Administradores de discusión	Esta herramienta incluida en varias plataformas permite a los docentes otorgar permisos a los participantes de su curso para que creen foros de discusión en distintos momentos.
	Herramientas de carga de archivos	Permite a los participantes de un curso compartir archivos con el resto de participantes del curso, actualmente varias plataformas han incluido funcionalidades que permiten incluir archivos no solo del equipo local sino también desde diferentes repositorios de almacenamiento tales como Dropbox, Google Drive, etc.
	Correo electrónico interno	Esta herramienta permite a los usuarios enviar mensajes por medio de correo electrónico de manera interna, bien sea entre los participantes de un curso, o entre los usuarios de la plataforma
	Notas (diario)	Permite a los usuarios de las plataformas gestionar un diario de notas personales o privadas, de esta manera el usuario puede registrar notas sobre las experiencias del curso y compartirlas si lo desea con otros usuarios.
	Chat en tiempo real	Con esta herramienta los usuarios tienen a su disposición un recurso por medio del cual pueden comunicarse en tiempo real e intercambiar mensajes e información.



Categoría	Recurso	Descripción Recurso
	Pizarra compartida	Esta herramienta permite la realización de trabajo colaborativo entre participantes de un curso, en la cual cuentan con un tablero compartido que puede ser accedido de manera remota para modificar el contenido del mismo.
<b>Herramientas de Productividad</b>	Marcadores	Esta herramienta permite al usuario mejorar su navegabilidad por la plataforma, ya que el usuario puede marcar ciertas páginas para acceder más fácil a ellas.
	Herramientas de gestión de actividades por medio de calendario	Permite a los usuarios gestionar y planificar la realización de tareas asociadas a los cursos de la plataforma, en algunos casos el usuario puede ver su progreso en una actividad específica.
	Orientación o ayuda	Herramienta que aporta información sobre cómo usar la plataforma de aprendizaje
	Buscador dentro de cursos	Permite al usuario encontrar de manera más fácil los materiales que necesitan para la realización de una actividad por medio de la búsqueda por palabras claves.
	Trabajo fuera de línea	Algunas plataformas han integrado herramientas que permiten a los usuarios trabajar sobre las actividades de su curso aun estando fuera de línea, dichas herramientas se encargan de sincronizar el trabajo realizado por los usuarios una vez se retorna la conexión.
<b>Herramientas para Alumnos</b>	Grupos de trabajo	Permite a los usuarios crear grupos de trabajo para la realización de distintas actividades o tareas dentro de un curso.
	Herramientas de auto-evaluación	Permite a los usuarios llevar seguimiento de sus actividades dentro de un curso y obtener retroalimentación sobre su rendimiento.

Categoría	Recurso	Descripción Recurso
	Carpeta de alumnos	Este recurso permite a los alumnos de un curso mantener información y archivos que pueden ser compartidos con otros usuarios o en un curso en específico
<b>Herramientas del Curso</b>	Herramientas de evaluación y puntuación automatizada	Permite a los docentes crear evaluaciones y obtener la calificación de las mismas de manera automática.
	Herramientas de gestión del curso	Son el conjunto de herramientas y recursos en general que permiten a los usuarios docentes o tutores gestionar los diferentes materiales, recursos, actividades, etc. Que son mostrados a los usuarios.
	Información de contacto instructor	Permite a los instructores agregar una serie de información de contacto que pueda ser compartida con los demás usuarios de la plataforma.
	Herramientas de calificación	Estas herramientas asociadas a la calificación automática en algunos casos, permiten a los docentes gestionar las calificaciones que obtienen sus estudiantes en cada curso, algunas plataformas incluyen servicios que permite delegar la gestión de calificaciones de un curso a otro usuario de la plataforma.
	Seguimiento al alumno	Permite a los usuarios docentes vigilar o dar seguimiento al uso que los estudiantes dan a los diferentes materiales de un curso.

Es importante resaltar que el prototipo de sincronización presentado en esta propuesta solo abarca el uso de recursos tipo archivos, debido a las limitaciones de tiempo para el desarrollo del presente proyecto.

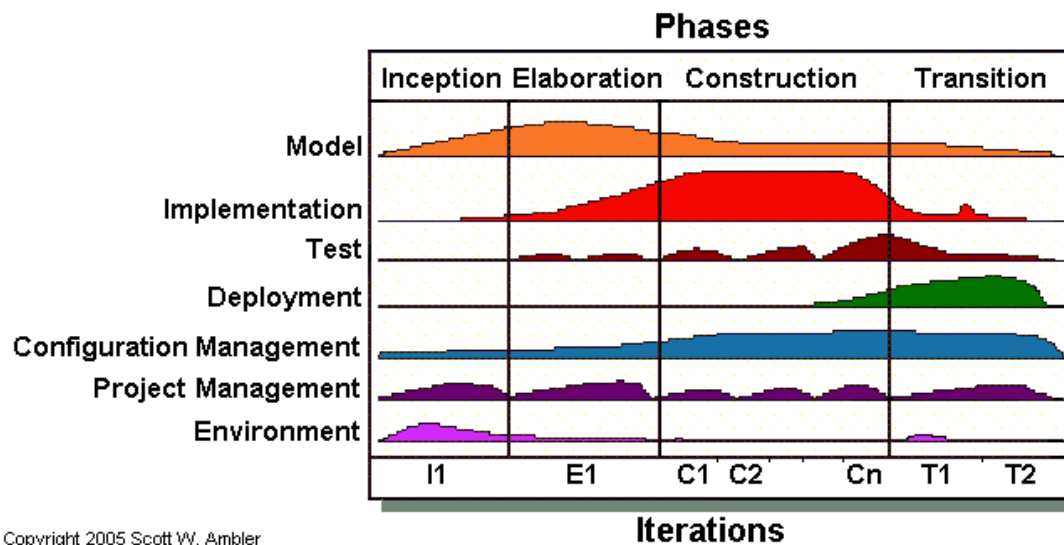
## CAPITULO II. ADAPTACION DE LA METODOLOGÍA AGIL UP

Dentro del proceso de realización del prototipo para la sincronización por demanda de actividades asociadas a actividades y/o preferencias de los usuarios en plataformas de aprendizaje en línea, se adoptó una adaptación de AUP (Proceso Unificado Ágil de Desarrollo de Software) como metodología que regiría el diseño y desarrollo del prototipo mencionado anteriormente.

### 2.1 METODOLOGIA AGIL UP

AUP es una metodología de desarrollo de software que representa un versión simplificada de RUP (Rational Unified Process) que implementa un enfoque de desarrollo de software utilizando técnicas y conceptos ágiles pero manteniendo su afinidad con RUP al mantener un esquema iterativo e incremental dentro del proceso de desarrollo de software.

En la Figura 10 se muestra de forma general las fases, iteraciones y disciplinas planteadas por AUP para el desarrollo de un producto software. En particular para el desarrollo de este documento se tendrán en cuenta las fases de AUP Inicio, Elaboración, Construcción y transición como mecanismo para la descripción del proceso de implementación del prototipo propuesto. [45]



Copyright 2005 Scott W. Ambler

Figura 10. ciclo de vida de AUP

Tomado de [45]

### **2.1.1 Disciplinas de AUP**

Las disciplinas contenidas dentro del proceso de AUP son un marco que permite agrupar una serie de actividades necesarias dentro del proceso de ejecución de un proyecto software; la utilización de estas disciplinas permite lograr una fácil comprensión del proceso de diseño y desarrollo de un producto software. Las disciplinas comprendidas dentro de AUP son las siguientes: [46]

#### ***Modelado***

La disciplina de modelado comprende las actividades necesarias dentro del proyecto software que permiten en primera instancia comprender el entorno de negocio de la organización a la cual está enfocado el producto, la delimitación del dominio del problema que se pretende abordar y la determinación de la solución viable dentro del dominio del problema encontrado.

#### ***Implementación***

Esta disciplina comprende el proceso de llevar a cabo la codificación de los modelos que se han diseñado pensando en dar solución a la necesidad planteada, con el objetivo de obtener un producto funcional. Además en esta disciplina se consideran también la realización de pruebas básicas de unidad frente al código realizado.

#### ***Pruebas***

En esta disciplina se realiza la evaluación del producto obtenido buscando validar el nivel de calidad de la solución, la detección de defectos, la verificación del cumplimiento con los requerimientos establecidos y la correcta operación del producto en funcionamiento.

#### ***Despliegue***

En esta disciplina se establece la planificación de los mecanismos necesarios para la entrega del proyecto finalizado y la puesta en marcha del producto software, que permita dejarlo disponible para el usuario final.

#### ***Administración de la configuración***

Disciplina transversal a todas las fases de desarrollo del proyecto software, y tiene como objetivo el manejo del acceso a todos los productos de trabajo generados (artefactos) en el transcurso del proyecto. Incluyendo el seguimiento de las versiones de cada artefacto, controlando y gestionando el manejo de cambios.

#### ***Administración del proyecto***

Disciplina que tiene como objetivo gestionar las actividades propuestas a lo largo del proyecto; donde se incluye la administración de los riesgos presentes, el manejo del

personal (tareas, roles, cumplimiento de metas, etc.), el manejo y coordinación de los elementos necesarios para obtener el producto software en los tiempo especificados, el presupuesto establecido y la puesta en marcha.

### ***Ambiente***

Es la disciplina en la que se determina como se va a soportar el proyecto, enmarcando las normas y directrices a seguir en el transcurso del desarrollo, como también la elección y disponibilidad de las herramientas software y hardware que sean requeridas por el equipo de trabajo en cada fase.

## **2.1.2 Fases de AUP**

Las fases de AUP corresponden a una división del ciclo de vida del proceso de diseño y desarrollo de un producto software. Cada fase hace énfasis en una o varias de las disciplinas anteriormente expuestas, con el objetivo de lograr los hitos comprendidos dentro del proceso de ejecución del proyecto software en dónde el alcance satisfactorio de cada hito representa la madurez del desarrollo del proyecto software. [47]

### ***Inicio***

Como su nombre lo indica es la fase en la cual se da inicio con la estructuración, establecimiento de los objetivos y la viabilidad del proyecto software. Dentro de esta fase se genera una serie de artefactos como casos de uso, definición de riesgos del proyecto, estimación de costos, cronograma y la preparación del entorno del proyecto. La fase de inicio cuenta con una serie de actividades principales: [48]

- Definir el alcance del proyecto.
- Estimación de costos y calendario.
- Definición de riesgos.
- Determinar la factibilidad del proyecto.
- Preparación del entorno.

### ***Elaboración***

La fase de elaboración tiene como objetivo principal la elaboración de la arquitectura del sistema a desarrollar, partiendo del análisis de los casos de uso obtenidos en la fase anterior y de la contextualización del dominio del problema; permitiendo que se pueda desarrollar un sistema que satisfaga de la mejor forma las necesidades y requerimientos establecidos. Durante esta fase se establecen los requisitos con el detalle suficiente para asegurar una comprensión del alcance de los mismos, y la determinación de los riesgos presentes en la ejecución del proyecto. Dentro de las actividades principales consideradas en esta fase se tiene: [49]

- Identificación de los riesgos.
- Modelado y validación de la Arquitectura.

- Prototipado de interfaces de usuario.
- Refinación del Modelo de pruebas.

### ***Construcción***

Es la fase que tiene como principal objetivo la implementación y desarrollo de las funcionalidades del producto software basado en la arquitectura obtenida en la fase anterior. La implementación del producto software se hace en forma iterativa logrando por cada iteración una nueva versión ejecutable del producto. Durante el transcurso de esta fase se llevan a cabo las siguientes actividades: [50]

- Refinamiento de la lógica del negocio, interfaces de usuario, esquemas de datos.
- Pruebas del producto software.
- Despliegue del productos en habientes de pruebas.
- Documentación del sistema.

### ***Transición***

La fase de transición está enfocada a la entrega del producto final, teniendo como principal objetivo liberar el sistema desarrollado a producción en el entorno del usuario final, lo que implica la realización del plan de pruebas exhaustivo que permita la detección de errores y verifiquen el correcto funcionamiento del producto obtenido. Es necesario durante esta fase llevar a cabo la capacitación del usuario final. Se deben tener en cuenta dentro de esta fase las siguientes actividades: [51]

- Proveer soporte técnico que permita la subsanación de errores encontrados, dentro del funcionamiento del producto software.
- Finalización de la documentación del proyecto.
- Realización del despliegue o liberación del producto software.

## **2.2 ADAPTACION DE LA METODOLOGIA AGIL UP**

En el siguiente apartado se establece la adaptación realizada a AUP para la realización del prototipo para la sincronización por demanda de actividades asociadas a actividades y/o preferencias de los usuarios en plataformas de aprendizaje en línea, propuesto en el presente trabajo de grado. Fue necesario adaptar la metodología para este proyecto en particular, debido a que el equipo de trabajo solo estaría compuesto por tres personas y a pesar de que la metodología es ágil, si se realizaran todas las actividades por cada disciplina durante cada fase se excedería los recursos existentes o el tiempo de realización del proyecto, por tanto la actual adaptación comprende los elementos esenciales adoptados para cada fase de AUP, los cuales se describen a continuación.

### 2.2.1 Fase de inicio

En la fase de inicio de la construcción del prototipo para el presente trabajo de grado se adoptaron las siguientes actividades consideradas relevantes, enmarcadas dentro de cada disciplina [48]. La adaptación realizada se describe en la Tabla 4.

**Tabla 4. Actividades seleccionadas en la adaptación de la fase de Inicio**

Disciplina	Actividades propuestas en AUP	Actividad Seleccionada	Descripción
<b>Modelado</b>	modelado de requerimientos de alto nivel	Si	Se realizara una abstracción de los requerimientos del sistema y basado en estos el diseño de casos de uso se realizara durante la fase de elaboración.
	modelado de la arquitectura de alto nivel	Si	Se realizara el estudio preliminar de las arquitecturas dentro del modelo de computación en la nube y se diseñara la arquitectura general del mecanismo.
<b>Implementación</b>	Prototipado técnico	No	La labor de prototipado se enfocara totalmente en el prototipo de interfaz de usuario
	Prototipado de interfaces de usuario	Si	Se diseñaran los prototipos de interfaz de usuario según el documento de especificación de requisitos pero esta actividad se realizara en fase de elaboración
<b>Pruebas</b>	Plan de pruebas inicial	No	Dado que la planificación de pruebas se llevara a cabo basado en la arquitectura obtenida en posteriores fases.

Disciplina	Actividades propuestas en AUP	Actividad Seleccionada	Descripción
	Revisión inicial de producto del proyecto	Si	Estas actividades serán realizadas pero no contarán con documentación formal del proceso por tratarse de revisiones enmarcadas dentro de la asesoría académica.
	Revisión inicial de modelos	Si	
Despliegue	Identificar la ventana potencial de liberación	No	Dado que en el transcurso del desarrollo de la presente propuesta se obtendrá un prototipo de la aplicación no será lanzado ni liberado.
	Iniciar el plan de despliegue de alto nivel	No	
Administración de la configuración	Establecer la configuración del entorno	Si	Se establecerá una carpeta disponible para cada miembro del equipo que le permita un fácil acceso a los artefactos desarrollados.  Se creará un repositorio que permita la integración de código de manera fácil para el equipo de desarrollo.
	Colocar todos los productos bajo el Control de la Configuración.	Si	
Administración del Proyecto	Inicia la creación del equipo.	No	No será necesario realizar la selección del equipo dado que el equipo de trabajo ya se encuentra determinado.  No se realizara dado que por tratarse de un prototipo de aplicación académico los interesados en el proyecto pertenecen al equipo de desarrollo.
	Crear relaciones con los interesados del proyecto.	No	



Disciplina	Actividades propuestas en AUP	Actividad Seleccionada	Descripción
	Determinar la factibilidad y alcancé del proyecto.	Si	Se delimitan las funcionalidades necesarias y de alta prioridad, que requiera el prototipo para satisfacer la pregunta de investigación y se adapte a los tiempos asignados para el desarrollo del proyecto.
	Determinar un cronograma de alto nivel para el proyecto	No	No será realizado ya que en la entrega preliminar del proyecto (anteproyecto) se determinó un cronograma general que incluye la realización del prototipo.
	Desarrollar un plan de iteraciones para las siguientes iteraciones	No	
	Administre el riesgo	Si	Esta actividad será realizada pero no contarán con documentación formal del proceso por tratarse de revisiones enmarcadas dentro de la asesoría académica.
	Obtenga el apoyo y financiamiento de los interesados.	No	Por enmarcarse el proyecto desde el ámbito académico no es necesario obtener el apoyo, ni financiamiento externo.
<b>Entorno</b>	Establecer el entorno de trabajo	Si	Se realizara un breve estudio de las herramientas, lenguajes y protocolos disponibles para la implementación del prototipo. Y se realizara la selección para la construcción del prototipo.

## 2.2.2 Fase de Elaboración

Para la fase de elaboración se establecieron las siguientes actividades consideradas relevantes enmarcadas dentro de cada disciplina [49]. Estas actividades se describen en la Tabla 5.

**Tabla 5. Actividades seleccionadas en la adaptación de la fase de elaboración**

Disciplina	Actividades propuestas en AUP	Actividad Seleccionada	Descripción
Modelado	Identificar los riesgos técnicos	Si	Esta actividad será realizada pero no contará con documentación formal del proceso, por tratarse de revisiones enmarcadas dentro de la asesoría académica.
	Modelado de la Arquitectura	Si	Se realizara la adaptación de una arquitectura basada en el modelo de computación en la nube que dé solución a la pregunta de investigación de la presente propuesta y que tenga en cuenta la limitación del alcance del prototipo.
	Prototipo de interfaces de usuario	Si	Se realizara la creación de las interfaces necesarias para la construcción del prototipo, dentro de un programa de modelado de interfaces de usuario.
	Probar la arquitectura	Si	Esta actividad será realizada pero no contará con documentación formal del proceso, por tratarse de revisiones enmarcadas dentro de la asesoría académica.

Disciplina	Actividades propuestas en AUP	Actividad Seleccionada	Descripción
<b>Pruebas</b>	Validar la arquitectura	Si	Esta actividad será realizada con un grupo de trabajo compuesto por personal con conocimiento en el área de validación de arquitectura por el método ATAM, pero dado que no hace parte de los objetivos del presente trabajo, esta labor se enfocará en establecer los escenarios y solo se realizara el análisis de uno de ellos.
	Evolucionar su modelo de pruebas	No	Como se indicó anteriormente el presente proyecto incluye la validación del prototipo desarrollado con usuarios, el modelo de pruebas será realizado en fases posteriores.
<b>Despliegue</b>	Actualizar su plan de desarrollo	No	En esta actividad no se realizará el modelo de despliegue dado que por tratarse de un prototipo no será desplegado.
<b>Administración de la configuración</b>	Poner todos los productos bajo la control de administración de la configuración	Si	Los diferentes artefactos que sean realizados estarán disponibles por herramientas como Dropbox para los miembros de equipo de trabajo bajo una carpeta única que contendrá todos los entregables del proyecto.
<b>Administración del Proyecto</b>	Construir el equipo	No	No será necesario realizar la selección del equipo dado que el equipo de trabajo ya se encuentra determinado.

Disciplina	Actividades propuestas en AUP	Actividad Seleccionada	Descripción
	Proteja el equipo	No	Esta actividad no se encuentra considerada.
	Obtenga los recursos	Si	Por enmarcarse el proyecto desde el ámbito académico no es necesario obtener el apoyo recursos diferentes a los que ya se encuentran disponibles, salas de computo, acceso a internet y computadores para el desarrollo.
	Maneje el riesgo	Si	Esta actividad será realizada pero no contará con documentación formal del proceso, por tratarse de revisiones enmarcadas dentro de la asesoría académica.
	Actualice su plan de proyecto	No	Esta actividad será realizada pero no contará con documentación formal del proceso, por tratarse de revisiones enmarcadas dentro de la asesoría académica.
Entorno	Evolucione el entorno de trabajo	Si	Esta actividad será realizada pero no contará con documentación formal del proceso.
	Ajuste los materiales de procesos	No	Esta actividad no contara con documentación formal del proceso por tratarse de revisiones enmarcadas dentro de la asesoría académica.

### 2.2.3 Fase de Construcción

Para la fase de construcción se establecieron las siguientes actividades consideradas relevantes enmarcadas dentro de cada disciplina [50] Estas actividades en la Tabla 6.

**Tabla 6. Actividades seleccionadas en la adaptación de la fase de construcción**

Disciplina	Actividades propuestas en AUP	Actividad Seleccionada	Descripción
<b>Modelado</b>	Abordaje del análisis del modelado	Si	Esta actividad será realizada pero no contará con documentación formal del proceso, por tratarse de revisiones enmarcadas dentro de la asesoría académica.
	Diseño por modelado de lluvia de ideas	Si	Esta actividad será realizada pero no contará con documentación formal del proceso, por tratarse de revisiones enmarcadas dentro de la asesoría académica.
	Documento	Si	La realización de esta actividad se verá enmarcada en la construcción de los manuales concernientes a cada funcionalidad dentro del prototipo.
<b>Implementación</b>	Primeras pruebas	Si	Las pruebas realizadas sobre cada funcionalidad realizada, se llevaran a cabo cuando la funcionalidad este lista y se verificara que esta cumpla con lo requerido.
	Crear constantemente	Si	Esta actividad será realizada dentro del marco del cronograma ya establecido y cada artefacto y funcionalidad nueva serán automáticamente reflejados en la carpeta compartida y en repositorio.

Disciplina	Actividades propuestas en AUP	Actividad Seleccionada	Descripción
	Evolución de la lógica de dominio	Si	Esta actividad será realizada pero no contará con documentación formal, pero será evidente con la creación de versiones nuevas tanto de las funcionalidades como de los modelos.
	Evolucionar las interfaces de usuario	Si	Se llevaran a cabo la construcción de las interfaces previamente diseñadas. Teniendo especial relevancia la construcción de interfaces lo más usables y sencillas como sea posible, en el caso de las aplicaciones web se buscara siempre que la interfaz sea responsiva <sup>8</sup> .
	Evolucionar el esquema de dato	Si	La evolución del esquema de datos se irá refinando conforme las iteraciones de desarrollo se cumplan para satisfacer los posibles requisitos que cambien durante la ejecución del proyecto.
	Desarrollo de interfaces de activos legados	Si	Dado que la aplicación considera el modelo de computación en la nube, se debe crear interfaces de acceso entre las diferentes aplicaciones que se generen en la construcción del prototipo.

<sup>8</sup> Una interfaz responsiva es aquella que se adapta al tipo de dispositivo desde el cual se esta visualizando.

Disciplina	Actividades propuestas en AUP	Actividad Seleccionada	Descripción
	Generar el script de conversión de datos	No	No se considera una actividad de relevancia en la realización del prototipo, dado que el manejo de datos se hará por medio de un repositorio que mantendrá las copias de los códigos actualizadas.
Pruebas	Pruebas de software	Si	Esta actividad se llevará a cabo periódicamente, y dentro de los entornos de desarrollo creados para el prototipo.
	Evolucionar su modelo de pruebas	No	Como se indicó anteriormente el plan de pruebas será diseñado de forma posterior a la construcción del prototipo y será coherente con el tercer objetivo de este proyecto.
Despliegue	Desplegar el script de instalación	No	Esta actividad no se llevará a cabo debido a que el desarrollo del prototipo se hará directamente sobre el entorno de pruebas y no será necesario pasarlo a otro entorno de producción.
	Desplegar notas publicadas	Si	Esta actividad será reflejada en el presente documento en el capítulo de construcción y se describirán los avances del prototipo de sincronización por cada iteración realizada.
	Desplegar documentación inicial	Si	Esta documentación estará comprendida dentro de los manuales de usuario desarrollados para el prototipo de sincronización.

Disciplina	Actividades propuestas en AUP	Actividad Seleccionada	Descripción
	Actualizar su plan	No	En esta actividad no se realizará el modelo de despliegue dado que la por tratarse de un prototipo no será desplegado.
	Desplegar el sistema en un ambiente de pre-producción	No	Dado que el prototipo de sincronización se realizará por completo en el ambiente de prueba, no se requiere un despliegue de pre producción.
<b>Administración de la configuración</b>	Poner todos los productos bajo la control de administración de la configuración	Si	<p>Los diferentes artefactos que sean realizados estarán disponibles por herramientas como Dropbox para los miembros de equipo de trabajo bajo una carpeta única que contendrá todos los entregables del proyecto.</p> <p>Se establecerá el acceso al repositorio para el código creado. Este repositorio será creado dentro de google code y será accedido por medio de TortoiseSVN. El repositorio solo podrá ser accedido por los participantes del proyecto.</p>
<b>Administración del Proyecto</b>	Administre el equipo del proyecto	No	Esta actividad no se considera relevante dado que el mismo equipo de trabajo desarrollara el prototipo de sincronización por lo cual siempre habrá autogestión.



Disciplina	Actividades propuestas en AUP	Actividad Seleccionada	Descripción
	Manejo del riesgo	Si	Esta actividad será realizada pero no contará con documentación formal del proceso, por tratarse de revisiones enmarcadas dentro de la asesoría académica.
	Actualizar su plan de proyecto	No	No será realizado ya que en la entrega preliminar del proyecto (anteproyecto) se determinó un cronograma general que incluye la realización del prototipo.
Entorno	Apoyar al equipo	Si	Esta actividad será realizada pero no contará con documentación formal del proceso, por tratarse de revisiones enmarcadas dentro de la asesoría académica.
	Evolucionar el entorno de trabajo	Si	Esta actividad será realizada pero no contará con documentación formal del proceso, por tratarse de revisiones enmarcadas dentro de la asesoría académica.
	Establecer el ambiente de capacitación	No	Esta actividad no se realizará dado que por tratarse de un prototipo no será desplegado y por tanto no se requiere un habiente de capacitación.

## 2.2.4 Fase de Transición

Para la fase de transición se establecieron las actividades consideradas relevantes enmarcadas dentro de cada disciplina, que se presentan en la Tabla 7 [51].

**Tabla 7. Actividades seleccionadas en la adaptación de la fase de transición**

Disciplina	Actividades propuestas en AUP	Actividad Seleccionada	Descripción
Modelado	Modelado por Lluvia de Ideas	No	No se realizarán nuevos modelos debido a que sale del alcance inicial de la presente propuesta.
	Finalice la documentación general del sistema.	Si	Se realizará un compendio de todos los artefactos generados y serán presentados como anexos a la presente propuesta.
Implementación	Corrija defectos	Si	Se corregirán los defectos más relevantes encontrados después de la realización del focus group como mecanismo de validación del prototipo.
Pruebas	Validación del sistema	Si	Dado que el prototipo no será desplegado, la validación del sistema dependerá del resultado arrojado en la fase de pruebas.
	Validación la documentación	Si	Esta actividad será realizada pero no contará con documentación formal del proceso, por tratarse de revisiones enmarcadas dentro de la asesoría académica.
	Analice su modelo de pruebas	No	Dado el alcance de la presente propuesta, no se contará con nuevas modificaciones una vez se entregado el prototipo del sincronizador de archivos.

Disciplina	Actividades propuestas en AUP	Actividad Seleccionada	Descripción
<b>Despliegue</b>	Finalice el paquete de entrega o liberación	No	Dado que el prototipo de sincronización no será puesto en producción no es necesario generar un paquete de entrega y liberación. Se pondrán a disposición los códigos fuente y artefactos generados durante la realización del proyecto.
	Finalice la documentación	Si	Toda la documentación referente a la realización del prototipo de sincronización quedara adjunta a la presente propuesta.
	Anuncie el despliegue o liberación.	No	Dado que el prototipo de sincronización no será puesto en producción no es necesario llevar a cabo esta actividad.
	Capacite el personal	No	Dado que el prototipo de sincronización no será puesto en producción no es necesario llevar a cabo esta actividad.
	Libere el sistema en producción.	No	Dado que el prototipo de sincronización no será puesto en producción no es necesario llevar a cabo esta actividad.
<b>Administración de la configuración</b>	Poner todos los productos bajo el control de la administración de configuración.	Si	Los diferentes artefactos que sean realizados estarán disponibles por herramientas como Dropbox para los miembros de equipo de trabajo bajo una carpeta única que contendrá todos los entregables del proyecto.

Disciplina	Actividades propuestas en AUP	Actividad Seleccionada	Descripción
			Se establecerá el acceso al repositorio para el código creado. Este repositorio será creado dentro de Google Code y será accedido por medio de TortoiseSVN. El repositorio solo podrá ser accedido por los participantes del proyecto.
Administración del Proyecto	Administre el equipo de proyecto	No	Esta actividad no se considera relevante dado que el mismo equipo de trabajo desarrollara el prototipo de sincronización por lo cual siempre habrá autogestión.
Entorno	Establezca las operaciones y / o el ambiente de soporte	No	Dado que el prototipo de sincronización no será puesto en producción no es necesario llevar a cabo esta actividad.
	Recupere las licencias del software	No	Dado que el proyecto será desarrollado bajo el marco académico no es necesario llevar a cabo esta actividad.

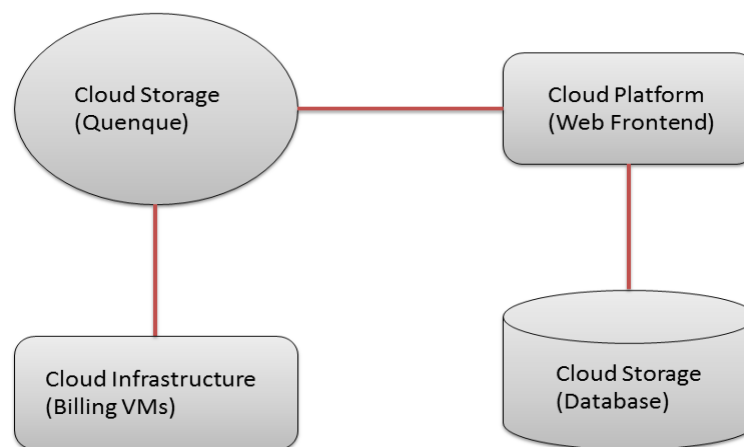
## CAPITULO III. FASE DE INICIO

El presente capítulo contiene la información relacionada con el trabajo realizado durante la primer fase del proceso de desarrollo actual. Este capítulo comprende el estudio de la arquitectura cloud computing necesaria para definir los requerimientos del mecanismo propuesto, posteriormente se presenta la propuesta de arquitectura para el mecanismo, la limitación del alcance del proyecto y la selección del conjunto de herramientas para el desarrollo del prototipo.

### 3.1 ARQUITECTURA CLOUD COMPUTING

La arquitectura en general para el modelo de cloud computing presenta en general cuatro elementos principales (infraestructura, almacenamiento, plataforma y servicios). El conjunto de estos elementos conforma un *modelo de servicio*, que de acuerdo a su mecanismo de interacción permiten ofrecer un servicio cloud.

La Figura 11 muestra los elementos principales del modelo de cloud computing.



**Figura 11. Arquitectura General Cloud Computing**  
Tomado de [52]

#### 3.1.1 Componentes de Arquitectura del Modelo Cloud Computing

##### ***Cloud Infrastructure***

La infraestructura hace referencia a la capa física del modelo de computación en la nube, debido que en este componente se encuentra ubicada toda la infraestructura hardware que permite soportar los servicios ofrecidos. Este componente permite la que las empresas puedan ofrecer infraestructuras virtualizadas, para que los clientes puedan obtener su propia infraestructura sin la necesidad de preocuparse por la necesidad del

mantenimiento y actualización dado que la infraestructura virtualizadas es fácilmente escalable y su mantenimiento corre por parte del proveedor de servicios [22][52].

### ***Cloud Storage***

Es la capa encargada de realizar el almacenamiento de los datos. Dicho almacenamiento se encuentra por lo general distribuido por diferentes lugares en grandes centros de datos. Estos centros de datos permiten la administración por medio de interfaces de usuario que permiten la comunicación para tener acceso a la información almacenada. El acceso a los datos depende en gran medida de cómo se estructure la nube de servicios, permitiendo que exista almacenamiento de carácter público, privado o mixto. Dentro de los protocolos de comunicación que permiten la comunicación y recuperación eficiente de los datos de estos centros de almacenamiento se tiene los siguientes:

- **SOAP (Simple Object Access Protocol)**

Es un protocolo estándar para servicios web basado en XML, el cual define como objetos en diferentes sistemas pueden comunicarse entre sí por medio de intercambio de datos XML. Este protocolo permite una mayor simplicidad, extensibilidad e independencia de cualquier plataforma, lenguaje, implementación o modelo de programación, por lo que SOAP puede funcionar bajo cualquier protocolo de transporte como HTTP, SMTP, TCP, etc [22][52][53].

- **REST (Representation State Transfer)**

Es un protocolo que consiste en directrices y mejores prácticas para la implementación de servicios web. “*Basado en principios o reglas de arquitectura de red, los estados y la funcionalidad de la aplicación se representan mediante recursos, utiliza HTTP para transferencia de información*”[54], por lo general su implementación se encuentra sujeta a las operaciones que se usan en el protocolo HTTP (GET, POST, PUT, DELETE) [55].

### ***Cloud Platform***

Es la capa en la cual se despliega el software necesario que solicita cada usuario a su proveedor de servicio. Dicho software es instalado sobre sistemas virtualizadas, lo que se traduce en que el proveedor de servicio es dueño del software y hardware subyacente a la nube y los clientes se convierten en propietarios de los recursos virtualizados. Por lo que la plataforma en la nube ofrece un entorno en el que se le permite a los desarrolladores crear y desplegar aplicaciones sin preocuparse de los recursos computacionales (memoria, procesadores, espacio en disco, etc.) consumidos en este proceso, ya que por el hecho de encontrarse en entornos virtualizados estos pueden ser escalados fácilmente [22] [52].

## **Cloud Service**

La nube de servicios representa la capa externa en la computación en la nube. Es la capa en la que se encuentran todos los servicios puestos a disposición de los usuarios finales a través de internet y quienes los consumirán por demanda desde los servidores del proveedor de cloud computing. En particular, la nube de servicios permite que los servicios ofrecidos dentro de esta sean escalables con facilidad para satisfacer las necesidades cambiantes de los usuarios y dado que esta capa esta soportada por la plataforma en la nube, las empresas, no se deben preocupar por los cambios en el hardware y software necesario para el funcionamiento adecuado de los servicios. Dentro de estos servicios ofrecidos se encuentra comúnmente: aplicaciones de correo electrónico, documentos en línea, almacenamiento de información, entre otros [22] [52].

### **3.2 DEFINICION DE REQUERIMIENTOS GENERALES**

Después de haber identificado las características generales del modelo de cloud computing, las propuestas existentes sobre el proceso de sincronización de archivos y las características de las plataformas de educación en línea, se procedió a establecer los requerimientos generales que debería cumplir el mecanismo de sincronización por demanda de archivos asociados a actividades de usuarios en plataformas de aprendizaje en línea.

Para esta labor se tuvo en cuenta las siguientes características generales:

- El mecanismo de sincronización debe ser diseñado para funcionar con distintas plataformas de aprendizaje en línea.
- Para reducir la carga de almacenamiento de archivos, el mecanismo debe tener la capacidad para utilizar servidores de almacenamiento tanto locales como remotos.
- Las acciones realizadas a través del mecanismo desarrollado deben ser transparentes para el resto de usuarios de la plataforma, es decir, si un usuario usa el mecanismo dentro de una plataforma y sube archivos a un curso, los archivos e información subida por medio del mecanismo deben ser visibles para los demás usuarios de la plataforma, aunque estos no empleen el mecanismo desarrollado.
- El mecanismo debe ser diseñado para facilitar el acceso de los usuarios hacia las actividades y los archivos relacionados a las mismas.
- Se debe tener en cuenta la seguridad de la información registrada en el mecanismo. Dado que para lograr comunicarse con las plataformas es necesario almacenar información de las cuentas de usuario en las plataformas.
- El mecanismo deberá ser diseñado para funcionar sobre el modelo de software como servicio de cloud computing.
- El mecanismo debe considerar el proceso de sincronización de archivos basado en detección de actualizaciones, reconciliación y manejo de cambios.
- Según el estudio realizado existen diferentes recursos (incluyendo los archivos) que apoyan las actividades de aprendizaje dentro de las plataformas de aprendizaje en línea. Dado que el mecanismo propuesto busca lograr la sincronización por demanda de archivos asociados a actividades de usuario, fue

necesario ampliar el esquema tradicional de sincronización (visto en el capítulo anterior), el cual se enfoca solo en archivos y no considera su manejo desde la perspectiva de las actividades educativas que los enmarcan cuando son usados dentro de una plataforma de aprendizaje en línea.

Una vez establecidas las características descritas anteriormente, se identificó que el mecanismo integraría como mínimo:

- Un servidor principal encargado de gestionar el proceso de sincronización y la información relacionada con los usuarios, plataformas, actividades, recursos, archivos, etc.
- Aplicaciones cliente que permitan al usuario interactuar con el servidor para gestionar las actividades y los archivos asociados a las mismas.
- Servidores de almacenamiento que podrán ser locales o remotos, estos últimos con el fin de reducir la carga de almacenamiento del servidor principal.
- componentes de integración como servicios web, plugins, etc, desarrolladas para las plataformas específicas asociadas al mecanismo que permitan comunicar el servidor con cada plataforma asociada.

Luego se definió usar, una priorización de requerimientos basada en los siguientes rangos:

- ALTA: sincronización por demanda de archivos asociados a actividades.
- MEDIA: no necesaria para el proceso básico de sincronización pero representa una funcionalidad básica que se debe tener en la aplicación.
- BAJA: si la funcionalidad no es considerada como necesaria para el proceso básico de sincronización y tampoco es considerada como funcionalidad básica que debe tener la aplicación.

A continuación se muestra la Tabla 8 que contiene el resumen de requerimientos, los cuales son un total de 48, el detalle de los requerimientos se encuentra en el Anexo 1. Documento de Especificación de Requisitos Mecanismo.

**Tabla 8. Resumen de Requisitos**

Aplicación	Modulo	Código. Nombre Requerimiento	Prioridad
Servidor Principal	Gestión de usuarios	MOD_GEST_USU_REQ01. Registrar Nuevo Usuario	ALTA



Aplicación	Modulo	Código. Nombre Requerimiento	Prioridad
		MOD_GEST_USU_REQ02. Editar Información de un usuario	ALTA
		MOD_GEST_USU_REQ03. Borrar la Información de un usuario	MEDIA
		MOD_GEST_USU_REQ04. Listar Usuarios del Mecanismo	MEDIA
		MOD_GEST_USU_REQ05. Buscar usuarios	BAJA
		MOD_GEST_USU_REQ06. Ver los detalles de la información de un usuario.	BAJA
		MOD_GEST_USU_REQ07. Registrar cuenta de plataforma	ALTA
		MOD_GEST_USU_REQ08. Editar Información de cuenta de plataforma	ALTA
		MOD_GEST_USU_REQ09. Borrar la Información de cuenta de plataforma	MEDIA
		MOD_GEST_USU_REQ10. Listar Cuentas de un usuario	MEDIA
		MOD_GEST_USU_REQ11. Buscar cuenta asociada	BAJA

Aplicación	Modulo	Código. Nombre Requerimiento	Prioridad
		MOD_GEST_USU_REQ12. Gestionar el proceso de autenticación de usuarios y manejo de sesiones	ALTA
		MOD_GEST_USU_REQ13. Gestionar el proceso de cierre de sesión.	ALTA
		MOD_GEST_USU_REQ14. Gestionar el control de acceso a las funciones	ALTA
	Gestión de Plantillas de Actividad	MOD_GEST_PLA_REQ01. Registrar Plantilla de Actividad	ALTA
		MOD_GEST_PLA_REQ02. Editar Plantilla de Actividad	ALTA
		MOD_GEST_PLA_REQ03. Borrar Plantilla de Actividad	MEDIA
		MOD_GEST_PLA_REQ04. Listar Plantillas de Actividad creadas por un usuario.	MEDIA
		MOD_GEST_PLA_REQ05. Listar Plantillas de Actividad creadas por otros usuarios.	MEDIA
		MOD_GEST_PLA_REQ06. Listar Plantillas de Actividad de todos los usuarios	MEDIA
		MOD_GEST_PLA_REQ07. Buscar plantillas	BAJA

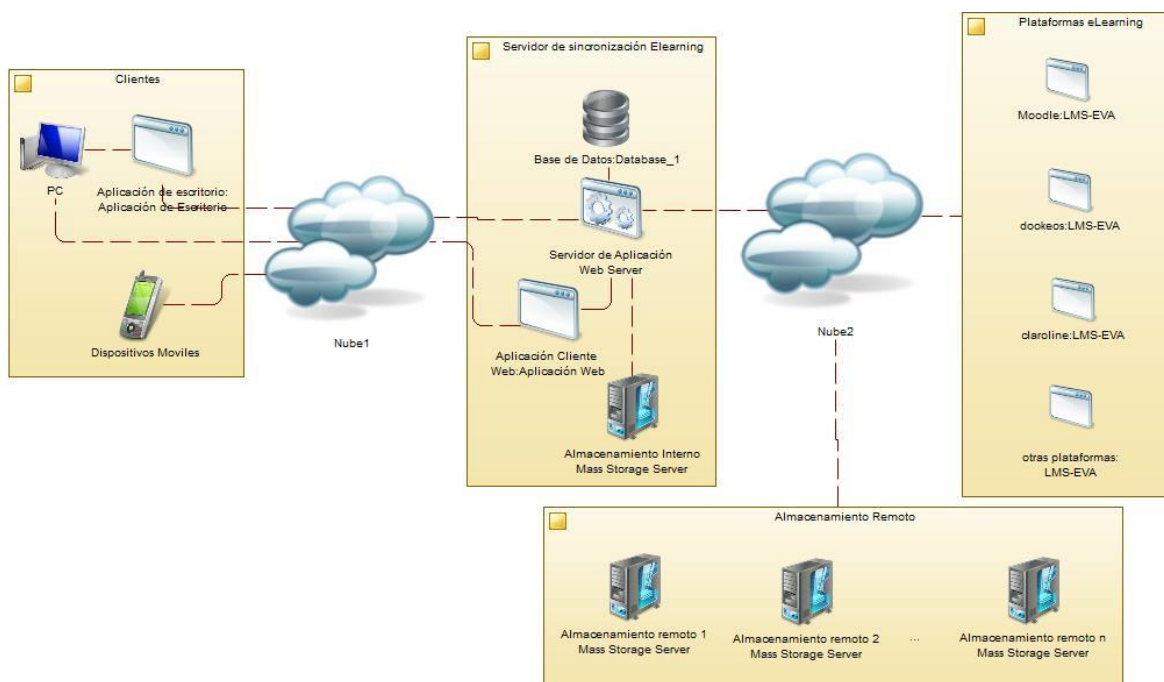
Aplicación	Modulo	Código. Nombre Requerimiento	Prioridad
		MOD_GEST_PLA_REQ08. Calificar plantilla	BAJA
		MOD_GEST_PLA_REQ09. Gestionar Comentarios sobre plantillas	BAJA
	Gestión de Actividades	MOD_GEST_ACT_REQ01. Registrar Actividad	ALTA
		MOD_GEST_ACT_REQ02. Editar Actividad	ALTA
		MOD_GEST_ACT_REQ03. Listar Actividades	ALTA
		MOD_GEST_ACT_REQ04. Gestionar la Sincronización de Archivos	ALTA
	Gestión de Notificaciones	MOD_GEST_NOT_REQ01. Registrar Notificaciones Automáticas para usuarios	ALTA
		MOD_GEST_NOT_REQ02. Listar Ultimas Notificaciones.	ALTA
		MOD_GEST_NOT_REQ03. Listar Historial de Notificaciones.	ALTA
	Gestión de y Plataformas Recursos	MOD_GEST_PYR_REQ01. Registrar Plataformas	ALTA
MOD_GEST_PYR_REQ02. Actualizar Plataformas		ALTA	

Aplicación	Modulo	Código. Nombre Requerimiento	Prioridad
		MOD_GEST_PYR_REQ03. Borrar Plataforma	MEDIA
		MOD_GEST_PYR_REQ04. Listar Plataformas	MEDIA
		MOD_GEST_PYR_REQ05. Registrar Recurso	ALTA
		MOD_GEST_PYR_REQ06. Actualizar Recurso	ALTA
		MOD_GEST_PYR_REQ07. Borrar Recurso	MEDIA
		MOD_GEST_PYR_REQ08. Listar Recursos existentes	MEDIA
		MOD_GEST_PYR_REQ09. Asociar Recurso con Plataforma	ALTA
		MOD_GEST_PYR_REQ10. Desasociar Recurso con Plataforma	ALTA
Cliente de Escritorio		APP_CLI_REQ_01. Detectar modificaciones de los archivos locales	ALTA
		APP_CLI_REQ02. Actualizar Actividades que han sufrido cambios	ALTA

Aplicación	Modulo	Código. Nombre Requerimiento	Prioridad
Servicio Web Moodle		PLA_MOO_SER_REQ_01. Obtener el listado de cursos en los que un usuario tiene rol de profesor editor	ALTA
		PLA_MOO_SER_REQ_02. Subir archivos dentro de un curso	ALTA
		PLA_MOO_SER_REQ_03. Obtener las secciones dentro del curso	ALTA
		PLA_MOO_SER_REQ_04 Fijar información básica de la sección	ALTA
		PLA_MOO_SER_REQ_05. Obtener el listado de usuarios participantes de un curso	ALTA
		PLA_MOO_SER_REQ_06. Borrar registro de módulo de recurso dentro de un curso en Moodle	ALTA

### 3.3 PROPUESTA DE ARQUITECTURA DE ALTO NIVEL

En el ítem anterior se expusieron los requerimientos que se deben considerar para proponer el mecanismo de sincronización por demanda de archivos asociados a actividades de usuario en plataformas de aprendizaje en línea. A partir del análisis de estos aspectos, se propone la siguiente arquitectura para el mecanismo de sincronización de archivos, basada en el modelo software como servicio de Cloud Computing. Esta arquitectura se presenta en la Figura 12 y tiene los siguientes componentes:



**Figura 12. Arquitectura General del Mecanismo de Sincronización.**

**Aplicación cliente:** Este componente agrupa todas las aplicaciones que funcionaran como interfaz de usuario. Por medio de dichas aplicaciones los usuarios podrán hacer uso de los servicios ofrecidos por el servidor de aplicaciones del componente Servidor de Sincronización. De esta manera, los usuarios podrán realizar la sincronización de archivos asociados a actividades de aprendizaje de manera transparente y recibir notificaciones de los cambios realizados en las diferentes actividades. Es importante resaltar que el proceso de sincronización se realizará *por demanda*, es decir, los usuarios podrán seleccionar las actividades de aprendizaje que desean sincronizar. Las aplicaciones cliente propuestas dentro la arquitectura tienen la responsabilidad de realizar la detección de cambios.

Uno de los objetivos del mecanismo propuesto es lograr que el usuario pueda hacer uso del sincronizador desde diferentes dispositivos por lo que se deberá contar con dos implementaciones independientes, la primera es una aplicación de escritorio que podrá ser instalada en los equipos de uso común de los usuarios y la segunda una aplicación web que permita al usuario acceder a los servicios desde cualquier dispositivo.

**Servidor de Sincronización:** este componente representa el eje principal de esta propuesta y tiene entre sus funciones:

- Interactuar con las aplicaciones que hacen parte del componente de aplicaciones clientes.
- Interactuar con las plataformas elearning asociadas.

- Manejo de información en base de datos de usuarios, actividades de aprendizaje, plataformas asociadas y los recursos y herramientas disponibles en ellas.
- Almacenamiento y gestión de archivos asociados a las actividades de aprendizaje de los usuarios, para esto se podrá contar con servidores de almacenamiento local y/o remoto.
- Realizar los procesos de reconciliación y resolución de conflictos concernientes a la sincronización por demanda de los archivos asociados a actividades de aprendizaje en las plataformas elearning.

**Plataformas Elearning:** la arquitectura planteada se diseñó con el objetivo de permitir al servidor de sincronización interactuar con distintas plataformas elearning. En el momento de desarrollo se deberá evaluar las plataformas elearning con las que se trabajará y para cada una de ellas será necesario establecer un mecanismo de comunicación que permita que los cambios hechos por los usuarios sean reflejados en la plataforma. El mecanismo de comunicación deberá además permitir la detección de cambios en las actividades de aprendizaje y replicar dichos cambios al servidor de sincronización para que este realice el proceso de sincronización y notifique a los usuarios.

### 3.4 REQUERIMIENTOS DEL PROTOTIPO

Una vez diseñada la arquitectura general del mecanismo de sincronización por demanda de archivos asociados a actividades de usuario en plataformas de educación en línea y de analizar el tiempo y recursos necesarios para desarrollar el mismo, se evidenció la necesidad de limitar el alcance de desarrollo del prototipo, por esta razón los requerimientos que habían sido considerados con prioridad alta y media para el proceso de sincronización serán los requerimientos que se desarrollaron y suman un total de 42. A continuación se muestran los requerimientos que no serán tenidos en cuenta:

**Tabla 9. Requerimientos que no serán incluidos en el alcance del prototipo**

Código. Nombre Requerimiento	Prioridad
MOD_GEST_USU_REQ05. Buscar usuarios	BAJA
MOD_GEST_USU_REQ06. Ver los detalles de la información de un usuario	BAJA
MOD_GEST_USU_REQ11. Buscar cuenta asociada	BAJA
MOD_GEST_PLA_REQ07. Buscar plantillas	BAJA
MOD_GEST_PLA_REQ08. Calificar plantilla	BAJA
MOD_GEST_PLA_REQ09. Gestionar Comentarios sobre plantillas	BAJA

Otros requerimientos que no se tendrán en cuenta hacen parte de requerimientos suplementarios, es decir, el prototipo a desarrollar tendrá también las siguientes limitantes:

- No se implementará el uso de servidores de almacenamiento.
- El algoritmo de sincronización estará enfocado en el proceso básico de sincronización y no maneja la resolución de conflictos.
- La única plataforma que se podrá usar es la plataforma Moodle, específicamente el desarrollo está enfocado en la última versión hasta el momento de inicio de codificación (versión 2.7 de Moodle).
- Para dicha plataforma solo se desarrollaran funciones del servicio web, dado que: 1) el plugin fue descartado después del estudio de la plataforma pues afecta negativamente la transparencia del mecanismo, ya que las actividades serian vistas como un nuevo recurso de la plataforma e implicaría que los usuarios tendrán que familiarizarse con un nuevo tipo de actividad, 2) el uso del API de Eventos ampliaría aún más el alcance de este proyecto.

### 3.5 SELECCIÓN DE HERRAMIENTAS DE DESARROLLO

Para el desarrollo del prototipo de sincronización por demanda de archivos asociados a actividades de usuario en plataformas de educación en línea es necesario establecer las herramientas de desarrollo que se requieren para la implementación del prototipo.

A continuación se presenta en la Tabla 10 las herramientas seleccionadas para el desarrollo de la presente propuesta, exponiendo los motivos por los cuales fueron seleccionadas.

**Tabla 10. Herramientas seleccionadas para el desarrollo del prototipo**

	Nombre de la Herramienta	Criterios de selección
Lenguajes de Programación	C#	<p>Se decidió el uso de c# como lenguaje para la creación de la aplicación de escritorio por existir un conocimiento previo por parte del equipo de desarrollo que permitiría mayor agilidad en el proceso de desarrollo.</p> <p>La elección de este lenguaje también cuenta con la posibilidad de trabajar en un entorno que facilita la creación de interfaces de usuario como lo es Visual Studio.</p>



	Nombre de la Herramienta	Criterios de selección
	PHP	<p>La selección del lenguaje de programación para la aplicaciones web se realizó teniendo en cuenta que php es uno de los lenguajes más usados en la actualidad, adicionalmente se cuenta con un conocimiento previo que permitiría que el proceso de desarrollo se realice con mayor rapidez evitando la demora en el aprendizaje de una nueva tecnología.</p> <p>Otro de los aspectos importantes que ayudaron a la escogencia de este lenguaje de programación es que cuenta con un gran número de frameworks que permiten aún más agilidad en el desarrollo de una aplicación, en el caso en particular para esta propuesta se escogió el framework llamado cakephp [56].</p> <p>Por otra parte la plataforma Moodle también ha sido, que ha sido desarrollada con el lenguaje PHP, por lo que se evita el aprendizaje de otra tecnología.</p>
Herramientas de Scripting	CSS	Css es el estándar actual para la creación de interfaces de usuario para la web, su utilización permite el desarrollo de interfaces agradables, complementado con la utilización de bootstrap que es una serie de framework que permite la creación de interfaces responsivas adaptables a los diferentes tipos de pantallas existentes en la actualidad.
	Jquery	Su utilización fue determinada debido a que jQuery es una biblioteca de funcionalidades de JavaScript que permite una utilización más fácil y rápida de funciones que comúnmente se utilizan en el desarrollo de aplicaciones web basadas en el lenguaje de scripting javascript..
Motor de base de datos	MySql	MySql es uno de los motores de base de datos libre más usado, y dado las características de desarrollo del prototipo se adaptaba con facilidad a lo requerido en el desarrollo de la presente propuesta.

	Nombre de la Herramienta	Criterios de selección
Protocolo de comunicación para los servicios web	REST	Es uno de los mecanismos de comunicación preferidos para la creación de aplicaciones basadas en el modelo de computación en la nube. Su escogencia se determinó tras un análisis previo en las asesorías académicas sostenidas en la cual se evidencio el potencial que tenía el uso de este protocolo de comunicación en el desarrollo de la presente propuesta. La principal ventaja de Rest respecto a Soap es que este funciona directamente sobre el protocolo http, por lo que hace un mejor uso de los recursos de la red, mientras SOAP requiere de intérpretes del lenguaje de descripción de servicios soap (wsdl), por lo que su consumo de los recursos de red es ligeramente superior. Igualmente, la mayoría de los servicios de almacenamiento en la nube hacen uso de REST.

---

## **CAPITULO IV. FASE ELABORACION**

El presente capítulo contiene la información relacionada con el trabajo realizado durante la segunda fase del proceso de desarrollo actual. Como ya se ha especificado anteriormente, la metodología de desarrollo escogida es AUP, según la cual, el objetivo principal de la fase de elaboración es definir la arquitectura del sistema que se va desarrollar.

Por lo tanto el presente capítulo se centrará en la arquitectura adaptada para la construcción del prototipo referente a el segundo objetivo específico del proyecto. Por otra parte, en este capítulo se incluye información relacionada con el modelo relacional, el diseño y plantillas de casos de uso, los prototipos de interfaz gráfica y un acercamiento inicial hacia la validación de la arquitectura general del mecanismo por medio del método ATAM [57].

### **4.1 ACERCAMIENTO INICIAL DE VALIDACION DE ARQUITECTURA PROPUESTA CON EL METODO ATAM**

El trabajo mostrado en esta sección del documento corresponde a un primer acercamiento hacia la validación de la arquitectura general del mecanismo propuesta en la sección 2.5 de este documento por medio del método ATAM, esta labor fue realizada de manera adicional a lo establecido entre los alcances del actual proyecto, dado que el planteamiento y validación de arquitecturas corresponde regularmente a trabajos de grado de maestría y doctorado. La sesión realizada para este acercamiento fue realizada de forma posterior al diseño, desarrollo y validación del prototipo, sin embargo es mostrada en este capítulo dado que éste está enfocado en la arquitectura.

Para la realización de este acercamiento se buscó el apoyo y la asesoría del Doctor en ciencias de la computación Julio Ariel Hurtado profesor del Departamento de Sistemas de la Universidad del Cauca, quien conformo un equipo de personas capacitadas en el método ATAM para la realización de una sesión que permitiera establecer los escenarios problemáticos que podrían presentarse en la arquitectura propuesta y realizar el análisis de por lo menos uno de los escenarios más críticos encontrados.

ATAM es un método para la evaluación de arquitecturas con el cual es posible determinar las consecuencias de las decisiones arquitectónicas tomadas en relación con los atributos de calidad establecidos como requisitos del sistema, con el uso de este método es posible detectar e identificar áreas de riesgo potencial dentro de la arquitectura de un sistema complejo. [57]

La sesión de validación de la arquitectura fue organizada y dirigida por el equipo de evaluación ATAM compuesto por el Doctor Julio Ariel Hurtado y los estudiantes de ingeniería de sistemas en trabajo de grado Edwar Alejandro Giraldo y Yuli Andrea Ordoñez Guzman. La agenda establecida por el equipo ATAM fue la siguiente:

1. Presentación de ATAM

2. Presentación de impulsores de negocio actuales
3. Presentación de la arquitectura a evaluar.
4. Identificación de los enfoques arquitectónicos.
5. Generar árbol utilidad de atributo de calidad
6. Analizar los enfoques arquitectónicos.
7. Lluvia de Ideas y dar prioridad a los escenarios
8. Analizar enfoques arquitectónicos.
9. Presentar los resultados.

Durante la realización de la sesión se ejecutó la actividad 1 de la agenda por parte del estudiante Edwar Giraldo, quien explicó los conceptos básicos sobre la metodología ATAM, tales como los enfoques arquitectónicos, el árbol de utilidad, la identificación de los escenarios y priorización de los mismos y como se haría el análisis de los enfoques arquitectónicos.

Posteriormente el equipo de diseño de la arquitectura se encargó de realizar las actividades 2 y 3, mostrando los impulsores de negocio actuales y la arquitectura propuesta resolviendo las dudas presentadas por el equipo de validación ATAM, una vez realizadas estas actividades, se identificaron los siguientes enfoques arquitectónicos: Escalabilidad, Modificabilidad, Desempeño y Confiabilidad.

Posteriormente se procedió a generar el árbol de atributos de calidad con el cual se detectaron los atributos que podrían estar afectados por las decisiones arquitectónicas tomados y se priorizaron dichos atributos utilizando los criterios de Importancia y Dificultad de Implementación. Estos dos criterios fueron evaluados en un rango de tres valores de la siguiente manera; Alto (H), Medio (M) y Bajo (L). En la Tabla 11 se muestra el árbol de atributos de calidad generado durante la realización de la sesión.

**Tabla 11. Árbol de Atributos de Calidad**

ARQUITECTURA	Enfoques Arquitectónicos	Atributos de Calidad	Prioridad (Importancia, Dificultad)
<b>Mecanismo para sincronización por demanda de archivos asociados a actividades de usuario en plataformas de aprendizaje en línea.</b>	Escalabilidad	Almacenamiento	(H, M)
		Crecimiento de usuarios	(H, L)
	Modificabilidad	Integrabilidad	(H, H)
		Evolución del Servidor de aplicaciones ElearningSync	(H,H)

ARQUITECTURA	Enfoques Arquitectónicos	Atributos de Calidad	Prioridad (Importancia, Dificultad)
	Desempeño	Tiempos de respuesta	(H, L)
	Confiabilidad	Fallos en el proceso de resolución de conflictos	(H, H)
		Fallos en el servidor de aplicaciones ElearningSync	(H, L)
		Fallos en el servidor de Almacenamiento	(H, L)

La siguiente actividad consistió en seleccionar el escenario para analizar el creado por el atributo de integrabilidad, dado que uno de los objetivos del mecanismo es operar con distintas plataformas de aprendizaje en línea por lo que este escenario es de suma importancia y tiene una complejidad de desarrollo alta.

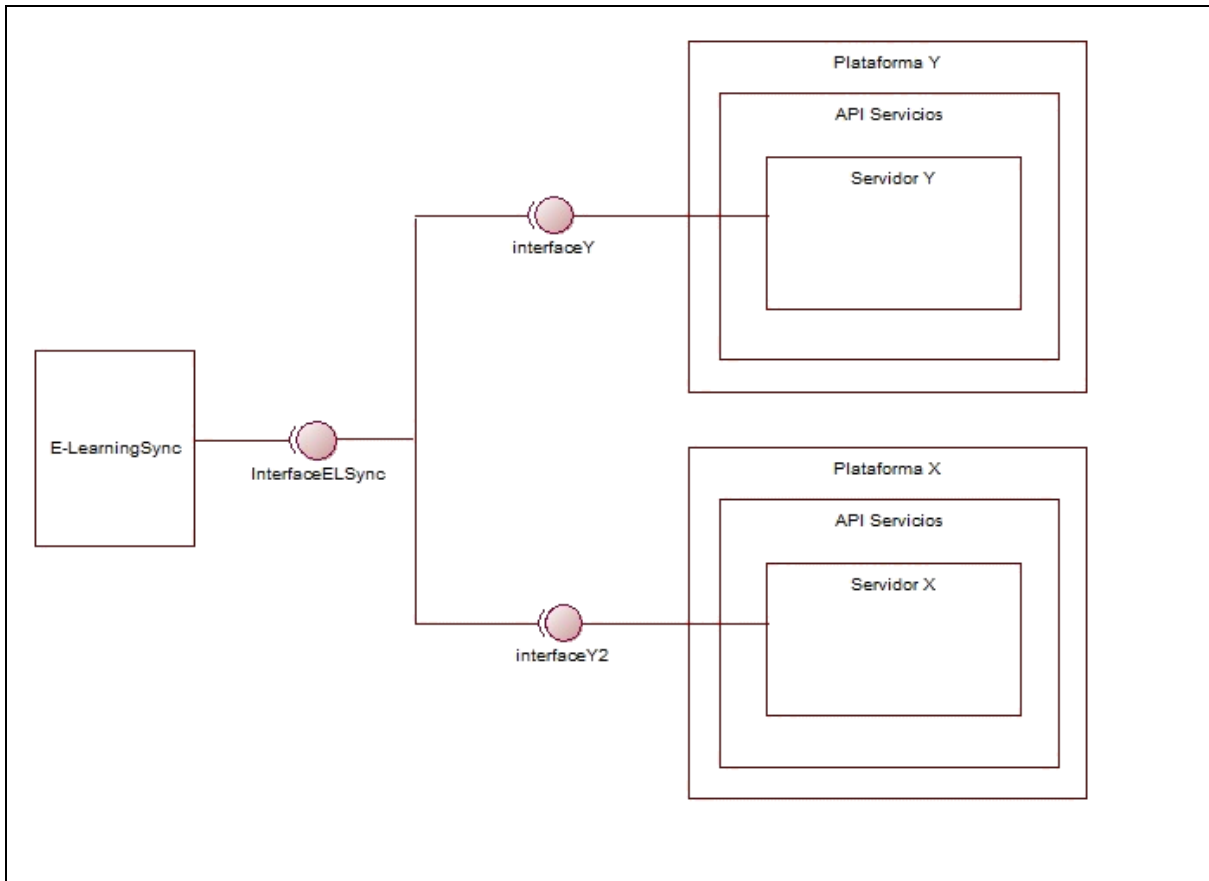
El análisis fue realizado siguiendo la estructura definida por el equipo de evaluación ATAM, para lo cual se realizó una descripción del escenario, se definió cual sería el estímulo para enfrentar o no dicho escenario, la respuesta que está establecida como el valor tendría atacar el escenario, y una tabla compuesta por las decisiones arquitecturales tomadas, los puntos sensibles, las compensaciones, los riesgos y los no riesgos, además de la definición de los *Rationale* (las razones por las cuales se tomaron las decisiones arquitecturales).

A continuación se presenta la abstracción del planteamiento del escenario en la Tabla 12.

**Tabla 12. Descripción de Escenario #1 ATAM**

Descripción del Escenario	Mejorar la integridad del mecanismo con las plataformas de aprendizaje en línea de tal manera que sea posible detectar la creación y actualización de recursos dentro de un curso e informar al servidor de sincronización para que este actualice los cambios si es necesario hacia los usuarios
Atributo	Integrabilidad

Estimulo	Lograr que los recursos creados desde las plataformas de aprendizaje puedan ser compartidos hacia los usuarios del mecanismo		
Respuesta	Para lograr esto se requiere:  3.5 personas/ mes para realizar las funciones para una plataforma 0 personas/mes para modificar el servidor.		
Decisiones Arquitecturales	Puntos Sensibles	Trade Off	Riesgos
Abstraer Servicios Web con el protocolo Rest	Servidor Web ElearningSync	Portabilidad	
Abstraer Servicios para cada plataforma de aprendizaje en línea	Servidor Web en la plataforma	Portabilidad	Algunas plataformas pueden restringir por seguridad la realización de algunas acciones por medio de servidores web
La comunicación entre componentes está enfocada en conexiones cliente / servidor ó P2P	Conexión	Disponibilidad	Posibles pérdidas de información
Rationale	<ul style="list-style-type: none"> <li>- Transparencia en el uso de las plataformas, dado que si el docente usa el mecanismo, los recursos pueden verse desde la plataforma como si hubiese sido creados desde ella misma.</li> <li>- Inter-operabilidad con distintas plataformas</li> </ul>		
El diagrama de componentes para el atributo de interoperabilidad se muestra en la siguiente pagina.			



Dentro de la evaluación realizada con el método ATAM, el equipo encargado de la evaluación, presento los siguientes resultados:

- La arquitectura presenta riesgos en el atributo de confiabilidad, debido al uso de servidores de almacenamiento remoto. Esto significa que la disponibilidad de los archivos no depende únicamente del servidor desarrollado, si no del funcionamiento de los servidores de almacenamiento remoto, lo que puede ocasionar que los archivos no se encuentren disponibles el 100% del tiempo.
- Se presenta un riesgo en cuanto a la integración del sistema con nuevos servicios de almacenamiento remoto.
- El uso de un almacenamiento local presenta un punto de Trade-Off entre el desempeño y la disponibilidad del sistema, ya que el tener respaldo de los archivos en un servidor local, permite mejorar la disponibilidad de los archivos, pero afecta el desempeño dado que es necesario involucrar un procedimiento adicional para cuando el servicio de almacenamiento remoto no esté disponible.
- Dentro del sistema no hay un método de recuperar información en caso de pérdidas de conexión.

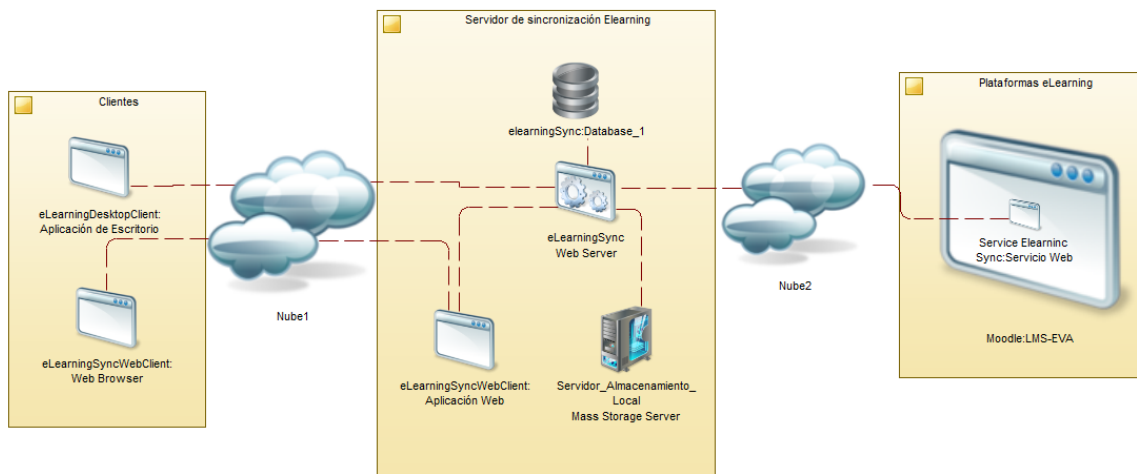
- La arquitectura presenta un punto de riesgo en la evolución del sistema al momento de integrar nuevas plataformas en el caso de que estas puedan restringir las acciones a realizarse por medio de los servidores web.
- El uso del protocolo Rest presenta un punto de Trade –Off, ya que se sacrifica la portabilidad con el rendimiento, debido a que el uso de servicios web con el protocolo Rest, limita a que las plataformas que se desean usar, deben hacer uso de este protocolo.

Se dificulta la lectura de la arquitectura debido a que en la representación que se hizo faltó el uso de convenciones, por lo cual se recomienda el empleo de las convenciones o buscar una mejor representación de la arquitectura. Además hizo falta la visualización de otros diagramas como el diagrama de componentes, el diagrama de paquetes y el diagrama de despliegue, que pudieran haber hecho más entendible la arquitectura.

## 4.2 ARQUITECTURA ADAPTADA PARA EL PROTOTIPO

Teniendo en cuenta que el desarrollo del mecanismo completo excede el tiempo y los recursos disponibles para la realización del trabajo de grado. La arquitectura tuvo que ser adaptada según los requerimientos fijados en el alcance descrito en el capítulo anterior. Esta Arquitectura se presenta en la Figura 13.

**Figura 13. Arquitectura adaptada para el desarrollo del prototipo**



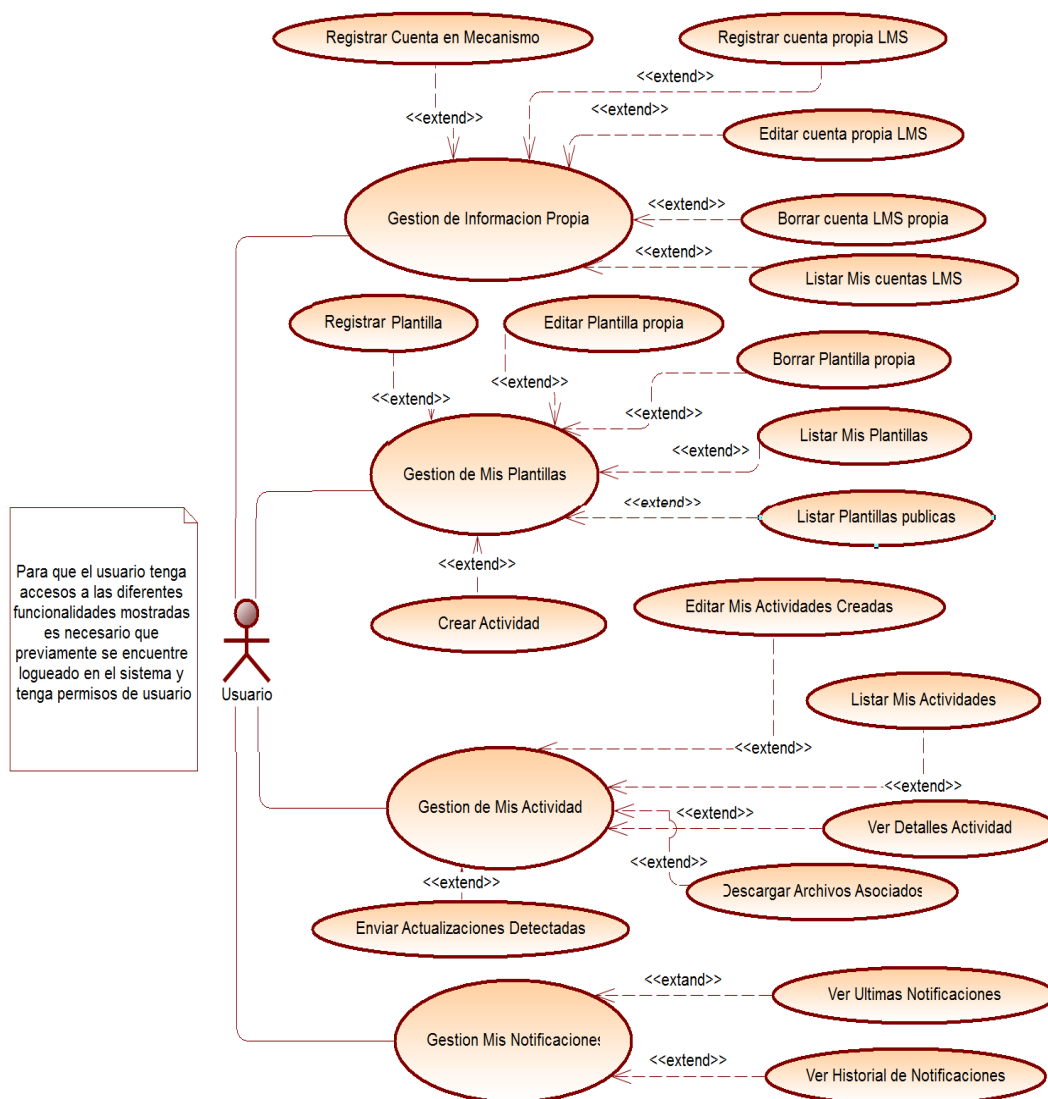
Una vez definida la arquitectura necesaria para el prototipo y definidas las aplicaciones a desarrollar en conjunto con requerimientos de las mismas especificados en el capítulo II. Fase de Inicio, fue posible proceder con el diseño de los casos de uso y la descripción de los mismos, diseñar los prototipos de las interfaces de usuario que serán usadas en las aplicaciones clientes y el diseño del modelo entidad-relación inicial, el modelo relacional posterior y por último el script de base de datos a usar.



### 4.2.1 Descripción de los casos de uso

Con el listado de requerimientos definidos en el Anexo 1. Documento de especificación de requerimientos mecanismo, se procedió a diseñar los diagramas de casos de uso por medio de la herramienta de diseño UML. Los artefactos generados pueden ser encontrados en formato nativo en el cd, en la carpeta Anexos/Modelos en PowerDesigner, A continuación se presenta en la Figura 14 el diagrama de casos de uso para usuarios registrados.

**Figura 14. Diagrama de casos de uso usuarios registrados**



## 4.2.2 Diseño de prototipos de interfaz de usuario

Dentro del proceso de desarrollo seguido se decidió diseñar prototipos de interfaz de usuario con el fin de visualizar de forma más rápida como sería el funcionamiento del prototipo. A continuación se muestra un collage con las imágenes de algunos prototipos diseñados. Para el diseño de estos prototipos se utilizó la herramienta Balsamiq Mockups. Los demás prototipos pueden ser revisados en formato original en el cd en la carpeta Anexos/Prototipos.

Figura 15. Collage de Prototipos de Interfaz



## 4.2.3 Diagrama entidad - relación, diagrama relacional, script de base de datos.

Por medio de la herramienta de diseño se generó el primer modelo entidad relación basado en el listado de requerimientos, este modelo fue usado como base principal del modelado de base de datos. Posteriormente se generó el modelo relacional y el script de base de datos. En la fase de construcción se requirió realizar ajustes al modelo entidad - relación por lo que al finalizar esta fase se terminó con dos versiones de este modelo.



## CAPITULO V. FASE DE CONTRUCCIÓN Y TRANSICIÓN

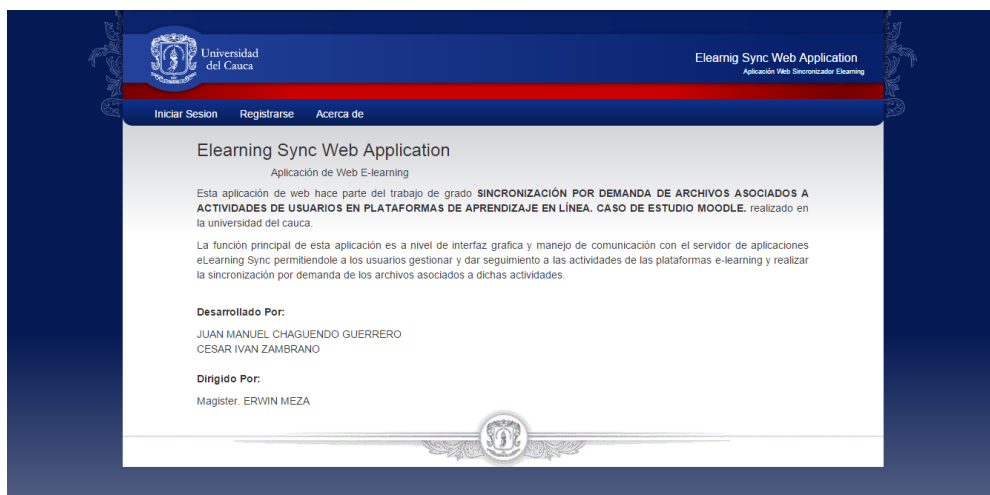
En este capítulo se aborda la construcción del prototipo de sincronización por demanda de archivos asociados a actividades de usuario en plataformas de educación en línea. Dentro del proceso de construcción del prototipo se llevaron a cabo una serie de iteraciones para conseguir el prototipo inicial que diera respuesta a la pregunta de investigación planteada en la presente propuesta y que sirviera de base para realizar una validación sobre todas las posibilidades que se podrían generar posteriormente. Adicionalmente se incluye en este capítulo un apartado que aborda la fase de transición y las actividades desarrolladas para esta fase.

### 5.1 Fase de Construcción

Dentro de la fase de construcción se llevó a cabo la implementación de la arquitectura adaptada definida en etapas anteriores, para la construcción del prototipo especificado fue necesario desarrollar cuatro aplicaciones diferentes que se describen brevemente a continuación:

#### ***Aplicación cliente web (ElearningSyncWebClient)***

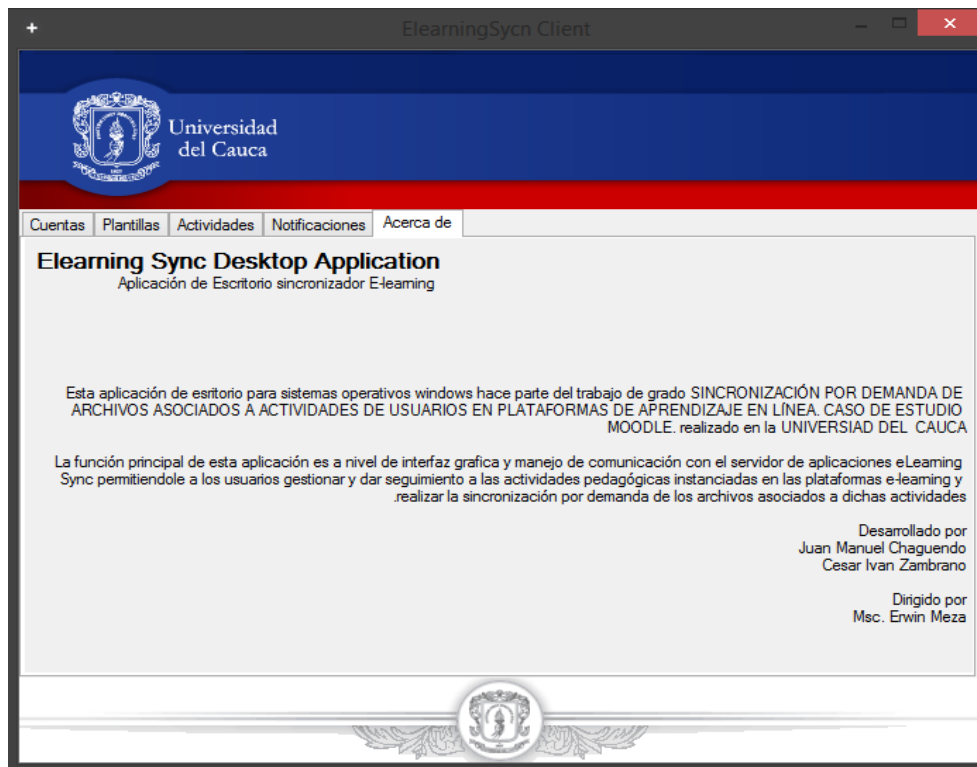
La aplicación cliente web tiene como objetivo permitir la comunicación del cliente con el servidor de sincronización, de tal forma que el usuario pueda interactuar con los recursos disponibles en el servidor desde un navegador convencional. Adicionalmente esta aplicación cuenta con una base de datos, que le permite hacer control de acceso sobre las diferentes funcionalidades. Una imagen inicial de esta aplicación puede ser vista en la Figura 17.



**Figura 17. ElearningSyncWebClient**

### ***Aplicación cliente Escritorio (ElearningSyncDesktopClient)***

La aplicación cliente escritorio tiene como objetivo permitir la comunicación con el servidor de sincronización para permitir que el usuario pueda interactuar con los recursos disponibles desde el servidor desde un computador personal. Esta aplicación también cuenta adicionalmente con un mecanismo de sincronización que permite mantener copias actualizadas de las actividades dentro del equipo del usuario y a su vez es la encargada de manejar la detección de cambios para que sean replicados hacia el servidor, una imagen inicial de esta aplicación puede ser vista en la Figura 18.



**Figura 18. ElearningSyncDesktopClient**

### ***Aplicación Servidor de Sincronización (ElearningSync)***

La aplicación servidor de sincronización tiene como objetivo la gestión de todos los requerimientos identificados, es el encargado de manejar todas las solicitudes, permitiendo la interacción entre las aplicaciones cliente, Moodle, el sistema de almacenamiento de datos y la base de datos.

### ***Módulo de servicios de comunicación con plataforma Moodle (Service ElearningSync)***

El módulo de comunicación con la plataforma Moodle cuenta con un paquete de servicios web que podrán ser consumidos desde el servidor de sincronización y le permitirán mantener la información consistente dentro de la plataforma, para que esta pueda ser accedida por los usuarios directamente en la plataforma Moodle. Este paquete de servicios web fue realizado siguiendo los lineamientos de Moodle para la creación de servicios web.

Por la complejidad del desarrollo que conlleva la realización del prototipo de sincronización por demanda de archivos asociados a actividades de usuario en plataformas de educación en línea, se abordó su desarrollo en diferentes iteraciones las cuales se exponen a continuación:

#### **5.1.1 PRIMERA ITERACIÓN**

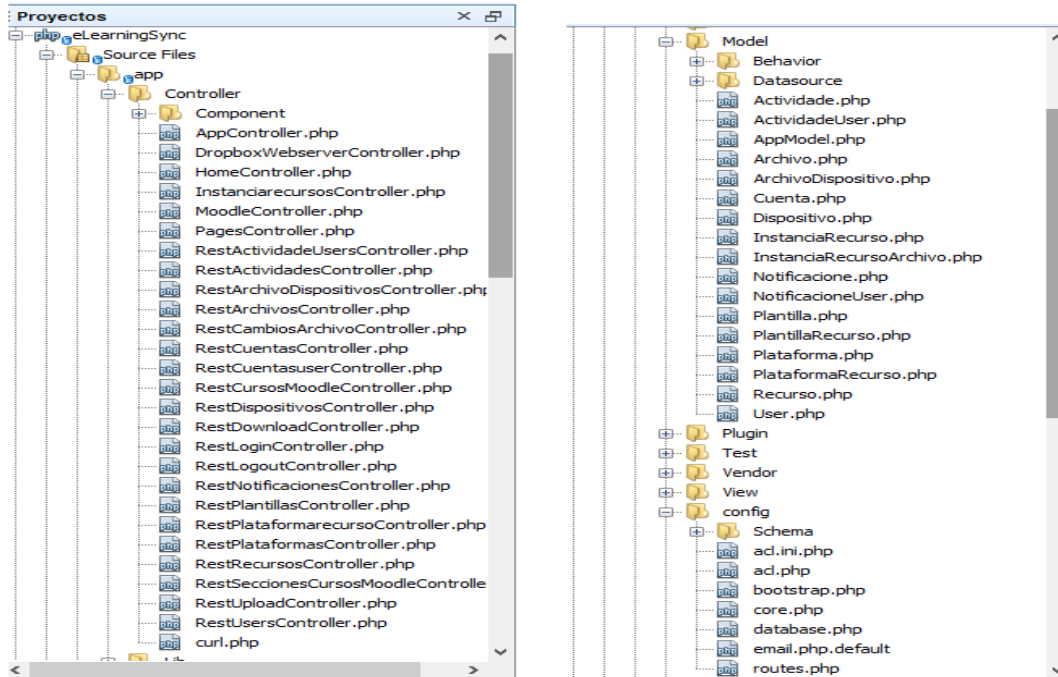
La primera iteración para la fase de construcción comprende el desarrollo de por lo menos el 60% de las funcionalidades del servidor de sincronización (ElearningSync), omitiendo las funciones de consumo de los servicios web de la plataforma Moodle. Para cumplir con el objetivo de esta iteración se realizó el desarrollo bajo el framework CakePHP[56], el cual está estructurado bajo el patrón MVC (Modelo, Vista controlador), que se encarga de separar los datos, la lógica de negocio y la interfaz de usuario. Para el servidor de sincronización la implementación del patrón de arquitectura solo se tuvo en consideración los siguientes componentes de MVC:

- **Modelos:** que representan los datos de operación del sistema.
- **Controladores:** que son los encargados de responder a las solicitudes invocadas comunicándose con los modelos para obtener la información necesaria y responder a la solicitud realizada.

Debido a que se usó el protocolo REST[58] para la definición de los servicios web dentro del servidor de sincronización, no fue necesaria la utilización de la capa de **“Vistas”**, dado que el servidor será accedido únicamente por servicios web y no es necesario crear ninguna interfaz de usuario para esta aplicación.

Partiendo de la base de que el servidor de sincronización está enmarcado dentro de framework de CakePHP se realizaron el desarrollo de las funcionalidades definidas en los casos de uso expuestos con anterioridad, para tal motivo y siguiendo las indicaciones de implementación del REST sobre CakePHP[58], para lo cual se desarrollaron en primera instancia todos los modelos con sus respectivas funciones de acceso a datos, luego se desarrollaron los controladores que contendrían todas las funcionalidades necesarias para dar respuesta a los requerimientos establecidos y por cada controlador creado se realizó la modificación del archivo **“routes.php”** en el cual se adiciona la definición de cada controlador para que las funciones contenidas dentro de este pudieran ser consumidas como un servicio web desde las aplicaciones cliente.

A continuación se muestra en la Figura 19 el esquema de archivos del servidor de sincronización haciendo.



**Figura 19. Esquema de archivos Servidor de sincronización**

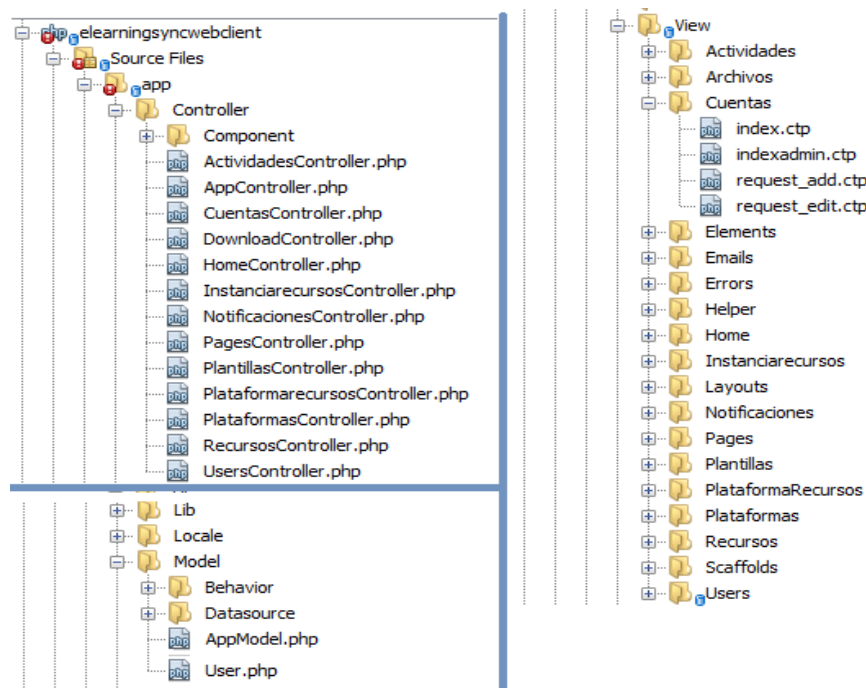
En la figura anterior se muestra la estructura de archivos realizada dentro del proceso de construcción del servidor de sincronización. En ella se pueden observar en la división de las capas de implementación. En la parte izquierda se pueden observar los controladores necesarios para cada funcionalidad definida dentro de los requerimientos estos se encuentran contenidos dentro de la carpeta **Controller**. En la parte derecha se pueden observar la capa de acceso a datos o modelos contenidos dentro de la carpeta **Model** que reflejan para este caso las tablas creadas dentro de la base de datos para el manejo de la información persistente que requiere el prototipo de sincronización. Por ultimo en la parte izquierda también se puede observar dentro de la carpeta **Config** el archivo *routes.php* en el cual como se mencionó anteriormente contiene las definiciones para lograr que las funciones contenidas en los controladores REST puedan ser consumidas por otra aplicación.

En paralelo a la construcción del servidor de sincronización también se llevó a cabo dentro de esta iteración la construcción de la aplicación cliente web (*ElearningSyncWebClient*). Esta fue con soporte del framework Cakephp[56] como una aplicación independiente que se conecta al servidor de sincronización para consumir los servicios ofrecidos y facilitar el acceso a las diferentes funcionalidades disponibles. En el desarrollo de la aplicación cliente web se configura para que sirva de enlace al servidor de

sincronización. La aplicación de cliente web, por estar implementada de forma independiente, debe contar con una capa de persistencia para el manejo de cuentas de usuario. Por esta razón dentro del modelo que tiene la aplicación cliente, en su implementación se establece la necesidad de adoptar el patrón de arquitectura MVC en su totalidad donde se tiene:

- **Modelos:** representan los datos de operación del sistema
- **Controladores:** son los encargados tomar las peticiones, ejecutar el consumo de los servicios web del servidor de sincronización y responder a las solicitudes invocadas por parte de los usuarios.
- **Vistas:** son la se encargadas de tomar la información obtenida por los controladores y reflejarla hacia el usuario.

A continuación se muestra en la Figura 20 el esquema de archivos para la aplicación cliente web desarrollada para el prototipo.



**Figura 20. Esquema de archivos Aplicación Cliente Web**

En la figura anterior se muestra la estructura de archivos realizada dentro del proceso de construcción del cliente web, en donde está reflejado en su totalidad el modelo MVC que se adoptó para el desarrollo. Se puede observar la capa de acceso a datos o modelos contenidos dentro de la carpeta **Model** que contiene solo un modelo dentro de ella el cual es llamado *users.php*. Este modelo hace referencia a una única tabla en la base que es



necesaria para el manejo de la autenticación dentro de la aplicación cliente web y que permite hacer uso del framework para poder hacer uso de las sesiones de usuario. [59]

La carpeta denominada **controller** contiene los archivos con la lógica necesaria para el consumo de los servicios ofrecidos por el servidor de sincronización. A diferencia del servidor de sincronización, en la aplicación de web se puede notar una diferencia substancial en el número de controladores necesarios. Esto se debe a que en el servidor de sincronización se debe crear un controlador REST por cada funcionalidad necesaria debido a la implementación del protocolo [58], lo que en la aplicación web no se da debido a que la implementación de cada controlador está orientada al consumo de servicios por lo cual se pueden agrupar las funcionalidades dependiendo la necesidad que satisfaga cada controlador.

Dentro de la carpeta **View** se encuentra la implementación de cada una de las vistas necesarias que desplegar la información obtenida desde el servidor de sincronización. Por las convenciones de Cakephp[56] es necesario crear por cada controlador desarrollado una subcarpeta dentro de la carpeta *view*. Por cada funcionalidad declarada dentro de un controlador se creó una vista con el mismo nombre de la funcionalidad, dichas vistas se encargan de mostrar al usuario las repuestas obtenidas al ejecutar una función específica. Para la codificación de las vistas se utilizó el framework Bootstrap [60] que permite que las vistas se puedan adaptar a diferentes dispositivos.

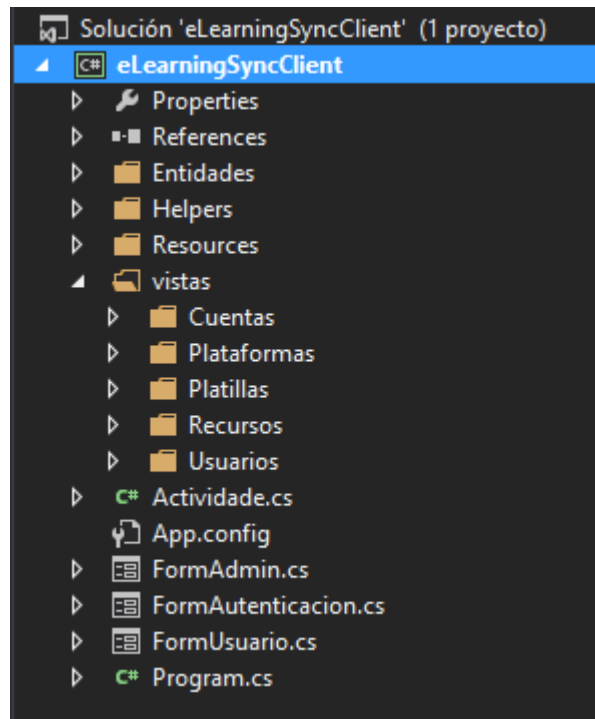
Por ultimo dentro de esta iteración se estableció que el servidor al ser una aplicación orientada a servicios basada en REST no almacenaría el estado de sesión de ningún usuario, se planteó como una alternativa para manejar la seguridad, desarrollar un inicio de sesión dentro de la aplicaciones cliente, que al ser usado genera un token con tiempo de expiración dentro del servidor de sincronización, este token debe ser enviado en todas las peticiones posteriores realizadas al servidor, quien se encarga de validar la legitimidad del token y dar el acceso o negarlo dependiendo del resultado de dicha validación.

El prototipo obtenido en la primera iteración: permitía la comunicación y la realización de las funcionalidades necesarias para la gestión de la información requerida dentro del mecanismo de sincronización y la aplicación cliente web que contenía las funcionalidades necesarias para lograr una conexión y consumo de servicios del servidor de sincronización.

### 5.1.2 SEGUNDA ITERACIÓN

En la segunda iteración se abordó la construcción de la aplicación de escritorio para hacer uso de las funcionalidades ofrecidas a través de servicios web por el servidor de sincronización. La aplicación escritorio construida en esta iteración solo considera su uso bajo plataformas Windows por lo que fue realizada bajo el lenguaje C#.

En la Figura 21 se presenta el esquema de archivos usados para el desarrollo de la aplicación de escritorio *ElearningSyncDesktopClient*.



**Figura 21. Esquema de archivos Aplicación Cliente Escritorio**

En la figura anterior se muestra la estructura de archivos definida dentro del proceso de construcción del cliente escritorio, en donde los archivos contenidos en la carpeta entidades hacen referencia a los objetos que debían ser creados para el manejo de la decodificación de las repuestas enviadas por el servidor. Las vistas son los diferentes formularios creados para el manejo de funcionalidades por parte del usuario.

La implementación de la aplicación de escritorio requirió implementar un mecanismo para la sincronización de archivos, por tanto en el desarrollo de la aplicación de escritorio fue necesario crear funcionalidades para poder descargar los archivos a sincronizar desde el servidor de sincronización en el computador del usuario. Para este propósito, al ejecutar la aplicación de escritorio dentro del sistema se genera una estructura de directorios para realizar la sincronización y el almacenamiento de los archivos de las diferentes actividades. Esta estructura consideraba la creación de una carpeta base que contendría carpetas adicionales, cada una llamada con el nombre de la actividad a la cual estaban asociadas. Con este modelo se pudo establecer el mecanismo de detección de cambios que consistiría en mantener una supervisión constante sobre las carpetas creadas para determinar si se había realizado cambios sobre alguno de estos archivos, y enviarlos al servidor de sincronización. Por último se creó dentro de la aplicación un hilo de ejecución que se encargaría de verificar con el servidor de forma periódica si existían cambios para realizar el proceso de sincronización.

La iteración termina obteniendo la aplicación de escritorio que se comunica con el servidor de sincronización y con las funcionalidades necesarias para el consumo de los servicios ofrecidos.

### 5.1.3 TERCERA ITERACIÓN

La tercera iteración consistió en lograr el proceso de comunicación desde el servidor ElearningSync hacia la plataforma Moodle y la implementación de los algoritmos de sincronización básicos para el proceso de sincronización automática, sincronización por demanda y sincronización de actualizaciones de actividades.

Para lograr la comunicación con Moodle se decidió realizar la implementación de un servicio web que integrase las funciones necesarias para el funcionamiento adecuado del prototipo. Se sabía de antemano que la mayoría de esas funciones deberían ser desarrolladas puesto que las funciones ofrecidas por el API de servicios de Moodle [61] no eran suficientes o útiles para las necesidades del prototipo, por lo tanto fue necesario investigar sobre el funcionamiento en general del API de Servicios de Moodle [62], como desarrollar nuevos servicios para la plataforma y crear clientes que consuman dichos servicios [63] y como configurar Moodle para activar el uso de servicios web [64]. También fue necesario estudiar las convenciones de desarrollo de Moodle [65], las convenciones de Sql [66] y el funcionamiento del API de Manipulación de Datos [67].

A continuación se presenta el esquema general de desarrollo cuando se emplea el uso de servicios web sobre la plataforma Moodle.

1. El cliente envía al script de autenticación de web service un nombre de usuario y contraseña.
2. El servidor devuelve una llave token para esa cuenta de usuario
3. El cliente invoca a la función particular del servidor web que desea consumir sobre el protocolo de servicios seleccionado, enviando el token obtenido en el paso anterior.
4. El servidor verifica por medio del token que el usuario tenga los permisos suficientes para ejecutar la función web solicitada.
5. El servidor ejecuta a la función externa solicitada dentro del módulo que se está solicitando usar.
6. La función externa verifica que el usuario que realiza la petición tenga las capacidades<sup>9</sup> necesarias para realizar esa operación.
7. Si se tiene los permisos, la función externa invoca a la función del *core* de Moodle<sup>10</sup> que es compatible con la función solicitada.
8. La función del core retorna los resultados a la función externa.

---

<sup>9</sup> En el léxico de Moodle, una capacidad se refiere a un permiso sobre un recurso o una funcionalidad.

<sup>10</sup> Función privada que realiza una acción como crear un usuario, crear un curso etc.

9. La función externa envía los resultados hacia el servidor.
10. El servidor envía la respuesta y/o resultado hacia el cliente.

Después de entender la lógica de funcionamiento de las API de servicios Web y Manipulación de datos, las necesidades de configuración de Moodle para habilitar los servicios web, y las convenciones de programación. Se procedió a realizar una prueba piloto consistente en utilizar las plantillas de cliente web de servicios y la plantilla de creación de servidores que incluye la función por defecto Hello\_World. Usando estas dos plantillas fue posible probar la comunicación básica entre el servidor ElearningSync y la plataforma de aprendizaje en línea. En la Figura 22 se muestra la imagen del archivo services.php con el código final de la declaración del servicio web desarrollado.

```
// We define the services to install as pre-build services. A pre-build service is not
$services = array(
    'Service Elearning Sync' => array(
        'functions' => array (
            'local_wselearningsync_hello_world',
            'local_wselearningsync_get_courses_role',
            'local_wselearningsync_upload_course_files',
            'local_wselearningsync_get_course_sections',
            'local_wselearningsync_set_info_sections',
            'core_enrol_get_enrolled_users',
            'local_wselearningsync_delete_course_modules'
        ),
        'restrictedusers' => 0,
        'enabled'=>1,
    )
);
```

**Figura 22. Código final de declaración del servicio Web y funciones asociadas.**

En la imagen anterior se muestra el código final del archivo de declaración de servicios el cual contara con siete servicios, donde uno de ellos pertenece a las funciones del core de Moodle y los otros seis hacen parte del desarrollo realizado. Como resultado de esta iteración se obtuvo el prototipo que permitió hacer la validación que sobre la solución planteada a la pregunta de investigación consignada en la presente propuesta.

## **5.2 Limitaciones Del Prototipo**

Dentro del desarrollo del prototipo se encontró con una serie de limitantes que imposibilitaron llevar a cabo algunos pasos en el proceso sugerido por AUP. Entre las limitantes encontradas se tiene:

- Por limitaciones del alcance del proyecto no fue posible implementar un mecanismo de resolución de conflictos dentro del prototipo realizado. Se encontró que solo el proceso de resolución de conflictos ha sido abordado en tesis de maestría [4][5].
- El proceso de sincronización implementado cuenta con la desventaja de no estar optimizado ya que ejecuta los algoritmos básicos para el manejo de sincronización de archivos.
- La comunicación desde el servidor de sincronización y la plataforma Moodle se hace de forma unidireccional ya que no fue posible la adaptación del api de eventos de Moodle para lograr una comunicación bidireccional.
- Por cambios del api de Dropbox en etapas avanzadas del desarrollo del proyecto no fue posible implementar el almacenamiento remoto de archivos y se optó por usar almacenamientos de archivos localmente dentro del servidor de sincronización.
- Por limitaciones de alcance del proyecto algunos de los requerimientos propuestos no fueron desarrolladas dentro de la presente propuesta del prototipo. Dichos requerimientos están descritos en la Tabla 7.
- Por limitaciones de alcance del proyecto la aplicación del escritorio solo funciona con conexión a internet el modo offline no está implementado.

### **5.3 Pruebas**

Durante cada una de las iteraciones ejecutadas en la fase de desarrollo se realizó de manera informal pruebas de caja negra para cada una de las funcionalidades implementadas, para esto se utilizó el documento de especificación de requerimientos, logrando establecer por cada requerimiento los prerrequisitos, entradas y pasos a realizar para llevar a cabo la prueba. Posteriormente se ejecutaba cada prueba y se contrastaba si el resultado obtenido era el esperado y en caso contrario se refinó las funcionalidades para lograr el resultado esperado por ellas.

Posteriormente, al finalizar la construcción de prototipo y la validación del mismo por medio del Focus Group se procedió a la formalización de dichas pruebas logrando identificar que las funcionalidades desarrolladas arrojaban los resultados esperados para cada uno de los requerimientos definidos, esta información puede ser revisada en profundidad en el Anexo 5. Plan de Pruebas Para el Mecanismo.

### **5.4 Fase de Transición**

Dentro de la fase de transición se llevó a cabo las inicialmente las actividades necesarias para que el prototipo pudiese ser evaluado. Estas actividades comprenden la puesta en marcha del prototipo desarrollado bajo un ambiente en el cual se pudiera tener contacto con usuarios reales que tuvieran la posibilidad de interactuar con la solución propuesta y pudieran brindar retroalimentación sobre el prototipo desarrollado. Para lograr este propósito se realizó una instalación dentro de un ambiente el cual contaba con un número determinado de computadores conectados al servidor de sincronización. El proceso de

validación y los resultados obtenidos se expresan con mayor detalle en el siguiente capítulo.

Dentro de las actividades adaptadas para esta fase se considera la conclusión en el proceso de documentación lo cual se llevó a cabo, generando los manuales de usuario y los diagramas y artefactos correspondientes los cuales se encuentran dentro de los anexos realizados para la presente propuesta.

---

## CAPITULO VI. VALIDACIÓN

Para la validación del prototipo de sincronización por demanda, se utilizó el método de grupos focalizados (Focus Group) con un grupo de discusión compuesto por expertos con dos perfiles de usuarios existentes en las actividades de aprendizaje (docentes y estudiantes).

### 6.1 Focus Group

El método Focus Group surge del entorno de las ciencias de la sociología y la psicología y consiste en debates focalizados que cuentan con una cuidadosa estructura y planificación que permita a los participantes exponer sus percepciones personales con la mayor espontaneidad posible en torno al tema objetivo de investigación y discusión. [68][69]

Según [69] la aplicación de Focus Group es adecuada, para: “realizar la evaluación inicial de potenciales soluciones, basado en los practicantes o usuarios”, lo que se asemeja al objetivo de validación de este trabajo, ya que hasta el momento se ha desarrollado un prototipo inicial de la arquitectura planteada, compuesto por las funcionalidades de mayor prioridad y se considera como principal criterio de validación la utilidad del mecanismo definida por la información entregada por los principales usuarios.

A continuación se presentan los pasos realizados en la planificación y ejecución de la sesión de Focus Group, siguiendo la estructura teórica y los procesos establecidos en [69]. Dichos procesos han sido establecidos con el objetivo de aplicar el Focus Group dentro de la ingeniería de software como un método útil para evaluar y validar propuestas teóricas y potenciales soluciones a partir del juicio de expertos.

#### 6.1.1 Fase de planeamiento de la investigación

La fase de planeamiento de la investigación está compuesta por dos actividades: definición del problema de investigación (donde se clarifica el objetivo del debate) y la preparación de materiales y métodos (cuyo objetivo es generar todos los elementos necesarios para la realización del debate).

##### 6.1.1.1 Definición del problema de investigación

El método de Focus Group es útil para nuestro objetivo de validación, el cual consiste en realizar la validación inicial del prototipo desarrollado como mecanismo de sincronización por demanda de archivos asociados a actividades de usuario en plataformas de aprendizaje en línea. Como base para el planeamiento se deberá utilizar el prototipo desarrollado conjunto con los requisitos establecidos para el mismo, de tal forma que los materiales y métodos a preparar brinden la información necesaria para conocer las funciones del prototipo (evitando influir en la percepción de los participantes) y recolectar información útil para la validación del prototipo respectivamente.

### 6.1.1.2 Preparación de materiales y métodos.

El objetivo de esta actividad es generar todos los materiales necesarios para la realización del Focus Group, además de los procedimientos y técnicas para llevar a cabo el debate e incluyendo los elementos para obtener la de información generada durante la sesión, para lo cual se ejecutaron las siguientes tareas

➤ Definición de Estructura de Debate

Para la realización del debate se realizó una sesión con la participación de dos docentes del Departamento de Sistemas de la Universidad del Cauca que han utilizado la plataforma Moodle como apoyo a sus procesos de enseñanza en por lo menos un curso, y con ocho estudiantes que han participado en por lo menos un curso apoyado por la plataforma Moodle. A continuación se presenta la estructura de la sesión de Focus Group en la Tabla 13.

**Tabla 13. Estructura de la sesión de Focus Group**

Estructura Focus Group			
<b>Lugar:</b>	Universidad del Cauca – Facultad de Ingeniería Electrónica y Telecomunicaciones – Departamento de Sistemas – Sala 4		
<b>Fecha:</b>	Viernes 6 de Marzo de 2015		
<b>Hora de Inicio:</b>	11:00 A.M.		
<b>Duración Estimada para La sesión:</b>	2 horas		
<b>Tema a Tratar:</b>	Evaluación y validación del mecanismo de sincronización por demanda de archivos asociado a actividades de usuario en plataformas de aprendizaje en línea, caso de estudio Moodle.		
<b>Moderador:</b>	Cesar Ivan Zambrano Espinosa		
<b>Relator:</b>	Juan Manuel Chagüendo Guerrero		
<b>Participantes:</b>	<b>DOCENTES</b>	<b>ESTUDIANTES</b>	
	1. Dr. Francisco Pino 2. Msc. Carlos Ardila	1. Ing Alexis Ante 3. Alexander Romero 5. Jhon Paz 7. Cristian Ledesma	2. Luis Ordoñez 4. Klauss Rendon 6. Juan Camilo Ceron 8. Deiber Gaviria



Estructura Focus Group	
<b>Objetivo General:</b>	Validar el prototipo desarrollado como mecanismo de sincronización por demanda de archivos asociados a actividades de usuario en plataformas de aprendizaje en línea.
<b>Objetivos Específicos:</b>	<ul style="list-style-type: none"> <li>➤ Presentar el prototipo desarrollado junto con los requerimientos asociados al mismo.</li> <li>➤ Conocer las apreciaciones acerca del prototipo desarrollado.</li> <li>➤ Validar si las funciones desarrolladas cumplen con los requerimientos planteados.</li> <li>➤ Validar la necesidad e importancia del mecanismo planteado desde la perspectiva de los usuarios.</li> <li>➤ Detectar posibles mejoras y trabajo futuro para el mecanismo planteado.</li> </ul>
<b>Protocolo:</b>	<p>El Moderador entregará a los participantes un documento inicial donde se encuentran las actividades que se van a realizar en la sesión. Este documento es Denominado “Guía de ejecución Focus Group para participantes”.</p> <ol style="list-style-type: none"> <li>1. La primera parte del documento y primera actividad de la sesión consisten en una contextualización del problema de investigación y la posible solución (prototipo desarrollado) por parte del moderador con la opción de intervención de los participantes para resolver los interrogantes que se generen.</li> <li>2. Posterior a esto se les solicita a los participantes registrarse en el servidor ElearningSync por medio de la aplicación de escritorio desarrollada ElearningSync Desktop Client y registrar la cuenta entregada a cada participante para la plataforma Moodle de pruebas, solicitando a un usuario docente y a tres estudiantes seleccionar el modo de sincronización automático, a un docente y tres estudiantes sincronización por demanda, y a dos estudiantes el modo de no sincronizar.</li> <li>3. El usuario Moderador con permisos de administrador tanto en el sitio Moodle como en el servidor ElearningSync se encarga de registrar a los docentes como profesores de un curso cada uno y de registrar a los usuarios estudiantes como estudiantes en los cursos usados para las pruebas.</li> <li>4. Se le solicita a los usuarios docentes que abran sus cursos por medio de la plataforma Moodle y creen en la primera sección una actividad básica compuesta por cinco archivos, además de modificar el título y descripción de la sección.</li> <li>5. Se le solicita a los usuarios estudiantes que abran por lo menos</li> </ol>

### Estructura Focus Group

- dos cursos y descarguen los archivos en el curso.
6. El Usuario Moderador explicara en esta actividad el concepto de plantilla de actividad y sus posibles usos a los usuarios, se les pedirá a los usuarios docentes que interactúen por medio de la aplicación cliente Web ElearnincSync Web Client, para crear una plantilla de actividad cada uno. Por otra parte se les solicita a los usuarios estudiantes que verifiquen por medio de la misma aplicación que las plantillas creadas como publicas aparezcan como creadas.
  7. Se le solicitara a los usuarios docentes que usen la aplicación cliente para instanciar una plantilla que cuente solo con recursos tipo archivo, como una actividad asociando archivos que sean de bajo tamaño en bytes dada las restricción de tamaño de Moodle, para lo cual agregan un nombre y descripción de la actividad y seleccionaran la sección del curso en la que cagaran la actividad, deberán definir además las fechas de realización de la actividad las cuales son usadas para mostrar a los estudiantes las actividades más próximas a realizar.
  8. Se verificara con los tres usuarios estudiantes y con el docente que habían definido sincronización automática, el funcionamiento del prototipo, de tal manera, que los archivos subidos por los docentes sean descargados de manera automática para estos usuarios.
  9. Se le solicita a los cinco usuarios estudiantes y al docente que no tenían definido la sincronización automática que accedan al módulo de actividades por medio de la aplicación cliente y verifiquen las actividades creadas por los profesores y seleccionen la opción de sincronizar en cualquiera de las actividades. De esta manera se validara el funcionamiento de la sincronización por demanda.
  10. Se le solicita a dos usuarios docentes que modifiquen una actividad creada por medio de la aplicación web y a los otros 2 que modifiquen archivos de una actividad por medio de la aplicación cliente.
  11. Se les solicita a todos los usuarios verificar que las actividades definidas para sincronizar se actualicen de forma correcta.
  12. Se les solicitara a los usuarios estudiantes que usen el módulo de notificaciones y verifiquen las notificaciones creadas de forma automática.
  13. Se les solicita a los usuarios llenar una encuesta inicial que está incluida en la guía entregada.

### Estructura Focus Group

	14. Se realiza el debate grupal donde el orden será inicialmente el establecido por las preguntas de la encuesta y posteriormente los temas o tópicos definidos por los participantes.
--	--

- Definición de instrumentos, materiales y métodos.

#### Materiales Utilizados

1. Plataforma Moodle Desplegada para la realización de esta prueba.
2. Servidor Web del Prototipo ElearningSync
3. Aplicaciones clientes: ElearningSync Desktop Client y ElearningSync Web Client.
4. Documento: “Anexo 4. Guía de Ejecución de Focus Group para participantes” que puede ser encontrado en el cd en la carpeta Anexos/Focus Group
5. Anexo 2. Ficha de Participantes del Focus Group, que puede ser encontrado en el cd en la carpeta Anexos/Focus Group
6. Formulario de encuesta del Focus Group, que se realizo de forma virtual por medio de la aplicación de formularios de Google Drive.
7. Un computador para cada participante del debate.
8. Un computador y videobeam para el moderador del debate.
9. Una grabadora de Audio para registrar el debate
10. Una Cámara Fotográfica para contar con evidencia visual de la realización del debate.

- Formalización de documentos a los participantes e instancias necesarias.

Dadas las particularidades de la sesión de Focus Group planteada fue necesario dividir la guía en diferentes instancias, en las cuales se entregaron a cada usuario datos de la cuenta usada en la plataforma Moodle y valores relacionados con el modo de sincronización que debería ingresar cada usuario.

Por otra parte se crearon tres documentos para enviar de manera anticipada a los participantes, el primer documento “Síntesis del problema de investigación y trabajo realizado” consistió en una síntesis del problema que se está abordando y una breve descripción del mecanismo propuesto conjunto con el alcance del prototipo. El segundo documento tenía el protocolo resumido que se seguiría en el debate y el tercero consistía en la ficha de participantes que debía ser diligenciada de manera previa al debate.

- Definición de Métodos de Captura y Registro de Información derivada del debate.

Debido a que los pasos definidos en el protocolo de realización de la sesión de debate eran numerosos y podrían extender la realización del debate por demasiado tiempo, se emplearan como mecanismos de recolección de datos los siguientes:

1. Ficha de Participantes diseñada con el objetivo de obtener información sobre el uso de las plataformas de aprendizaje en línea de cada participante.
  2. Documento de recopilación de apreciaciones generadas durante el debate realizado por el relator.
  3. Audio de la sesión de debate.
  4. Formulario en línea de la encuesta en la herramienta de Formularios de Google Drive, Las preguntas realizadas en la encuesta fueron diseñadas con el objetivo de obtener información sobre el funcionamiento del prototipo y las apreciaciones de los participantes sobre el prototipo y el mecanismo en general.
- Definición de métodos de análisis de información para generar el procesamiento de lo generado en el debate.

Posterior a la realización del debate, el grupo de investigación analiza la información obtenida por medio de las encuestas de manera cualitativa y cuantitativa; se debe contrastar el documento de recopilación de apreciaciones realizado por el relator con el audio grabado de la sesión de debate para lograr abstraer de la manera más exacta posible las percepciones de cada uno de los participantes y posteriormente se analiza dicha información con el objetivo de identificar las opiniones de los participantes en cuanto al mecanismo propuesto en general, el prototipo desarrollado y las posibles mejoras a aplicar tanto al mecanismo como al prototipo que serán incluidas de ser conveniente en el trabajo futuro de esta propuesta.

- Definición de estrategias de publicación.

Una vez obtenidos los resultados y las conclusiones del análisis estos serán reflejados en un documento que será entregado vía correo electrónico a los participantes de la sesión de Focus Group, esta información también se reflejará en esta monografía de trabajo de grado y si el grupo de investigación considera que la información obtenida durante esta sesión amerita la realización de un artículo científico que recopile la información más relevante capturada en la aplicación del método Focus Group.

## **6.1.2 Fase de definición de grupos de discusión**

Para la definición del grupo de participantes el grupo investigador realizó las siguientes actividades

### **6.1.2.1 Selección de participantes.**

Esta actividad fue realizada por el grupo investigador por medio de las siguientes tareas:

- Definición del perfil del participantes  
Dado que el prototipo desarrollado está enfocado en el caso de estudio Moodle, se estableció que los usuarios participantes sean decentes y estudiantes que hayan usado esta plataforma por lo menos un curso.
- Identificación de potenciales participantes.

Los participantes mostrados en la Tabla 14 son profesionales o estudiantes en el área de la ingeniería de sistemas, los estudiantes de la lista en su mayoría se encuentran en trabajo de grado, algunos a la espera de la ceremonia de grado y otros ya egresados. Todos los participantes han hecho uso de la plataforma de educación en línea Moodle.

**Tabla 14. Posibles Participantes de Focus Group**

DOCENTES	ESTUDIANTES	
<b>1. Dr. Francisco Pino</b>	3. Ing Carlos Suarez	2. Ing Alexis Ante
<b>2. Dra. Carolina Gonzales</b>	1. Ing Fabian Mondragon	4. Estudiante Alex Romero
<b>3. Msc Carlos Ardila</b>	5. Estudiante Luis Ordoñez	6. Estudiante Jhon Paz
<b>3. Ing. Jair Moreno</b>	7. Estudiante Klauss Rendon	8. Estudiante Camilo Ceron
<b>4. Ing Libardo Pantoja</b>	9. Estudiante Silvana Fernandez	10. Estudiante Edinson Castillo
<b>6. Dr Carlos Cobos</b>	11. Estudiante Wilmer Camacho	12. Estudiante Eduar Peña
<b>7. Ing Jimena Timana</b>	13. Estudiante Deiber Gaviria	14. Estudiante Cristian Ledesma

### 6.1.2.2 Segmentación de grupos.

- Definición de Criterios de Agrupación:

Dadas las características del prototipo desarrollado y la existencia de distintos modelos de sincronización en el mismo, se usa como criterio de agrupación de grupos los tres modelos de sincronización existentes en la plataforma.

➤ Aplicación de Criterios de Agrupamiento

Una vez aplicado el criterio de agrupación se definió que era necesarios los siguientes tres grupos:

1. Grupo de Participantes con Modo de Sincronización Automático
2. Grupo de Participantes con Modo de Sincronización Por Demanda
3. Grupo de Participantes con Modo No Sincronizar.

### **6.1.3 Fase de conducción de la sesión de debate**

La ejecución del debate se realizó con el grupo de participantes que aceptaron la invitación y que se encuentran registrados en el protocolo en la sección 6.1.1. Durante la sesión se siguió el protocolo establecido y se hizo uso de todos los materiales y artefactos resultantes de la primera fase de aplicación del Focus Group.

#### **6.1.3.1 Captura de Información**

El relator Juan Manuel Chagüendo designado para la sesión se encargó de registrar las observaciones y comentarios realizados por los participantes en la actividad 14 del protocolo correspondiente al debate grupal. Además se empleó el uso de la grabadora de audio y cámara fotográfica como técnicas de recolección de información.

#### **6.1.3.2 Rol del Moderador.**

Este rol fue realizado por Cesar Ivan Zambrano quien se encargó de dar cumplimiento a la agenda propuesta, presentando el mecanismo diseñado y los conceptos de plantillas en los momentos definidos en la agenda. Luego de probar el prototipo con los participantes del Focus Group se realizó la actividad 14 debate grupal, en la que todos los participantes pudieron aportar sus observaciones de forma crítica y constructiva.

#### **6.1.3.3 Productos de trabajo**

En las sesiones anteriores se ha informado de la realización de distintos productos de trabajo que se encuentran disponibles en el cd en la carpeta Anexos/Focus Group, entre los cuales se encuentran:

- Recursos previos enviados a los participantes de manera previa entre los que se encuentra el protocolo, el documento síntesis del problema de investigación y la ficha de participantes.
- Recursos entregados al inicio de la sesión, entrega de la Guía de ejecución del Focus Group.
- Documentos de relatoría y audio de la sesión, son los documentos que encapsulan las observaciones realizadas por los expertos. El primero fue realizado por el

relator durante la realización de la sesión y el otro es obtenido luego del análisis del audio recopilado.

- Reporte de Participantes, este documento se realizó con las fichas de participantes diligenciadas y registra la información relevante relacionada con la experiencia de los participantes entorno a las plataformas de aprendizaje en línea.
- Reporte de Análisis de resultados que se presenta en este documento en la próxima sección.

## **6.1.4 Fase de Análisis de Información y Reporte de Resultados**

### **6.1.4.1 Análisis de Información**

Para el análisis de la información inicialmente se contrastó la información reportada por el relator con el archivo de audio de la grabación, obteniendo un listado de posibles mejoras presentadas por los participantes. Posteriormente se realizó el análisis de la información obtenida con la encuesta virtual realizada.

### **6.1.4.2 Reporte de Resultados**

En la siguiente sección se mostrará un reporte de los resultados obtenidos con la sesión de Focus Group, inicialmente se mostrara un listado de los comentarios tanto positivos como negativos obtenidos en torno al mecanismo propuesto y al prototipo desarrollado. Posteriormente se mostrara un listado de las posibles mejoras a realizar tanto al prototipo como a nivel del mecanismo.

El siguiente listado es un grupo de comentarios tanto positivos como negativos realizados por parte de los participantes ante el mecanismo propuesto y el prototipo desarrollado, los comentarios considerados positivos serán marcados con (+) y los comentarios considerados como negativos serán marcados con (-) al inicio de cada uno:

- (+) El concepto de plantilla de actividad presentado es bastante útil debido a que agrupa los recursos de una manera donde cada uno aporta a un objetivo como tal y no como una serie de recursos dispuestos sin razón aparente.
- (-) La estructura de las plantillas debe ser refinada de tal manera que se pueda explicar cuál será el uso de cada recurso dentro de la actividad.
- (+) La posibilidad de compartir las plantillas de actividades entre distintos usuarios y poder comentarlas es importante porque permite obtener retroalimentación sobre los procesos de aprendizaje y evitar hacer reproceso.
- (+) la creación de plantillas y actividades por medio del mecanismo desarrollado es fácil de realizar y menos tediosa que crear una serie de recursos por Moodle.
- (+) Los procesos de sincronización automática y por demanda funcionaron de manera adecuada desde la percepción de los participantes.
- (+) El proceso de actualización de una actividad por medio del mecanismo es más directo y expedito del realizado por la plataforma Moodle al cambiar un recurso y la actualización de estos cambios hacia los usuarios se realizó de forma adecuada.

- (-) se debe mejorar las notificaciones y mensajes de la aplicación para que el usuario pueda reconocer que actividades y archivos se encuentran actualizados, cuales están en proceso de sincronización.
- (-) visualmente se debe usar iconos o herramientas visuales que permitan al usuario identificar el estado de las actividades y los archivos asociados en torno al proceso de sincronización.
- (+) según los comentarios el prototipo desarrollado es de utilidad.
- (+) es oportuno continuar con el desarrollo del mecanismo
- (+) la idea del mecanismo en general puede llegar a tener mercado.

A continuación se presenta una lista de las posibles mejoras propuestas por los participantes del Focus Group, se marcarán con una letra (f) las consideraciones que se incluirán como trabajo futuro y con un signo de asterisco (\*) aquellas mejoras que puedan ser solucionadas dentro del alcance del actual proyecto, tomando aquellas que representen una mejora significativa o importante en torno al eje principal del prototipo que es el funcionamiento de la sincronización de actividades y archivos asociados.

- (f) Se debe evitar tener que realizar el doble registro de los usuarios, ya que estos primero deben registrarse en el mecanismo y posteriormente enlazar su cuenta de la plataforma, se sugiere crear funcionalidad que permita registrarse con las credenciales de una cuenta de forma directa.
- (f) Registrarse en el sistema por medio de la autenticación con las credenciales de un LMS o con las redes sociales como Facebook.
- (f) Verificar en el registro de cuentas LMS que las credenciales ingresadas sean válidas.
- (f) Agregar nuevos criterios de ordenamiento de plantillas de actividad adicionales al criterio de calificación, para que el usuario pueda tener una mejor orientación de las plantillas creadas.
- (f) Cuando se asocian fijan los recursos en la creación de una plantilla de actividad sería importante agregar una descripción del objetivo de cada recurso para dar una mayor estructura a las plantillas.
- (f) Agregar funcionalidad que permita crear una plantilla desde otra ya creada.
- (f) Definir una estructura que permita obtener comentarios sobre las plantillas que aporten al mejoramiento y evaluación constructiva de las mismas.
- (f) Agregar una categorización o tipificación de las plantillas que permita saber cuál es su principal uso de manera más fácil, ejemplo: plantillas para preparación de evaluaciones.
- (\*) Solucionar el error existente que impide que las actividades sean creadas o modificadas de forma remota, ocurrido dentro de la sesión.
- (f) Mejorar iconografía sobre la carpeta en donde se encuentra los archivos y sobre las actividades que permita al usuario saber que los archivos están siendo sincronizados por la aplicación como lo hace Dropbox y Google Drive.
- (f) Mejorar información visual en la aplicación cliente que permita identificar si las actividades en los archivos están actualizadas, se están sincronizando, etc.
- (\*) Revisar si la sincronización se está realizando por cuentas y/o actividades y si es por cuentas modificar el algoritmo para que funcione por actividades.



- (\*) Mejorar la estructura de las notificaciones de tal manera que entregue información más detallada sobre los cambios producidos en las actividades en las que participan los usuarios.

### 6.1.4.3 Análisis de encuesta.

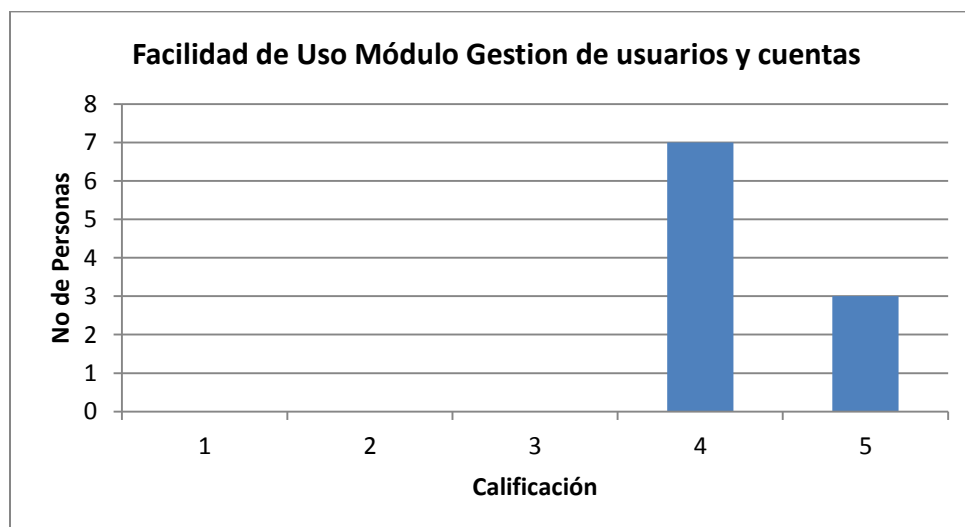
A continuación se presenta el análisis de la información obtenida por medio de la encuesta diligenciada por los docentes y los estudiantes que participaron en el Focus Group. La información obtenida por medio de la encuesta puede ser revisada en el cd en la carpeta Anexos/Focus Group/Anexo 7. Respuesta Encuesta Focus Group. También se incluye el documento Anexos/Focus Group/Anexo 8. Resumen de Respuestas de Google Drive que ofrece un análisis previo realizado de manera automática por la herramienta.

Los resultados de la encuesta han sido graficados y analizados de acuerdo al objetivo de la pregunta y/o el método a evaluar con cada una de ellas. A continuación se presentan las gráficas respectivas con el análisis de cada una de ellas.

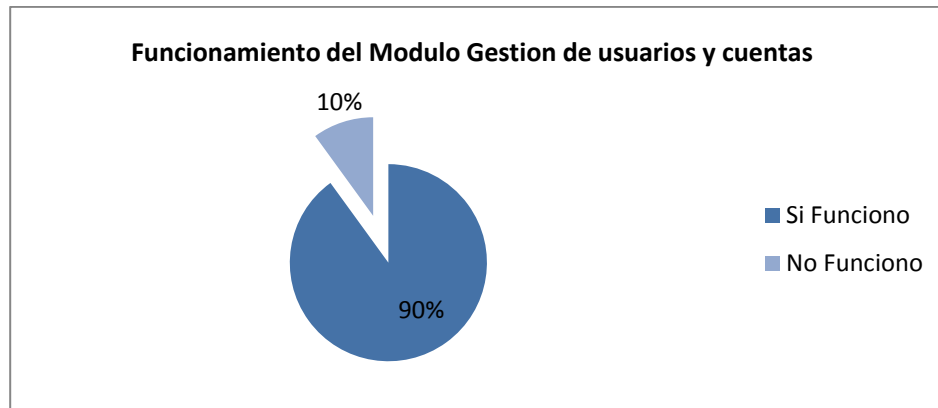
### Resultados del Módulo de Gestión de Usuarios y Cuentas.

Preguntas de la encuesta.

2. Califíquese de 1 a 5 la facilidad de uso del módulo de registro de usuarios y cuentas del prototipo.



3. El módulo de registro de usuarios y cuentas del prototipo funcionó de manera exitosa?

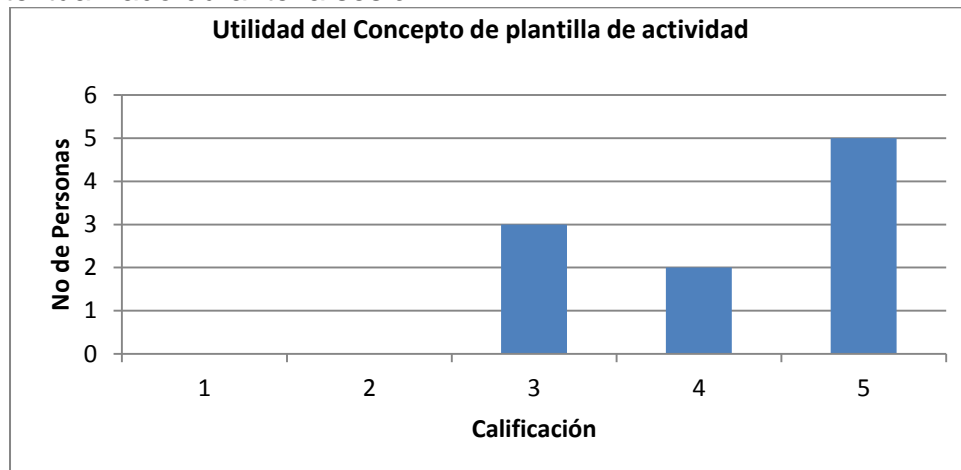


Con las respuestas obtenidas se puede concluir que el módulo de gestión de usuarios y cuentas funcionó de manera adecuada para el 90% de los participantes y su facilidad de uso es alta dado que el 70% de los participantes la califican como 4 y el 30% como 5.

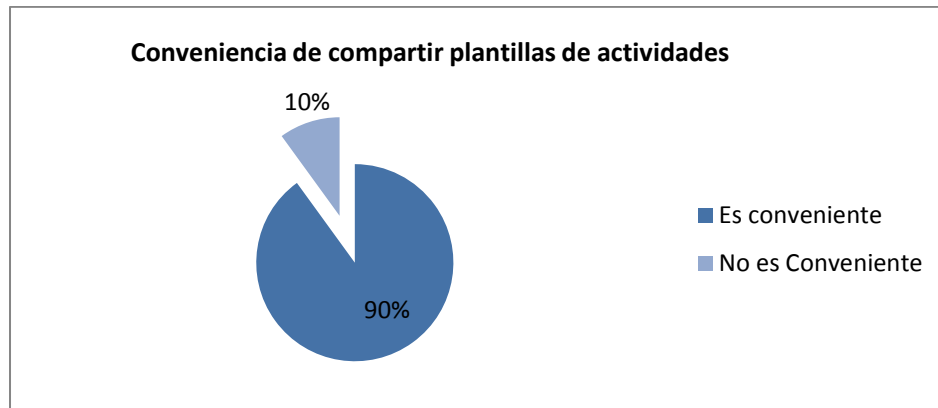
### Resultados Relacionados al Concepto de plantillas de Actividad

Preguntas de la encuesta.

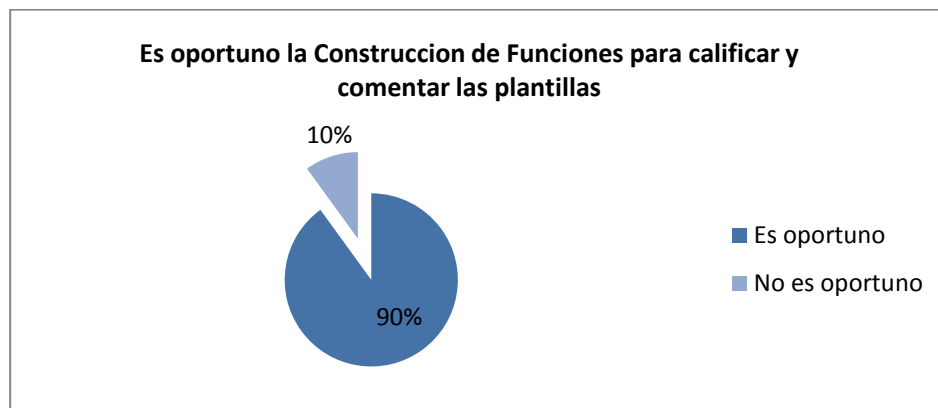
5. Califique de 1 a 5 la utilidad del concepto de plantilla de actividad contextualizado durante la sesión.



6. Considera usted conveniente que las plantillas de actividades puedan ser compartidas entre usuarios con el fin de apoyar los procesos de aprendizaje de los estudiantes por medio de las plataformas de aprendizaje en línea?



7. Considera usted oportuno desarrollar funcionalidades que permitan calificar y comentar las plantillas creadas para saber cuáles apoyan mejor el proceso de aprendizaje desde la perspectiva de los estudiantes y demás docentes?



Según las respuestas dadas por los expertos, el concepto de plantilla de actividad es útil, pero apoyándose en los comentarios brindados durante la realización del debate grupal se puede encontrar que es necesario estructurar mejor la asociación de los recursos de tal forma que se le pueda agregar a cada recurso una descripción que permita identificar cual es el objetivo que se busca cubrir con cada recurso y adicionalmente es necesario establecer una categorización de las plantillas que permita identificar de manera más fácil el objetivo de una plantilla, como por ejemplo: plantillas para preparación de evaluaciones.

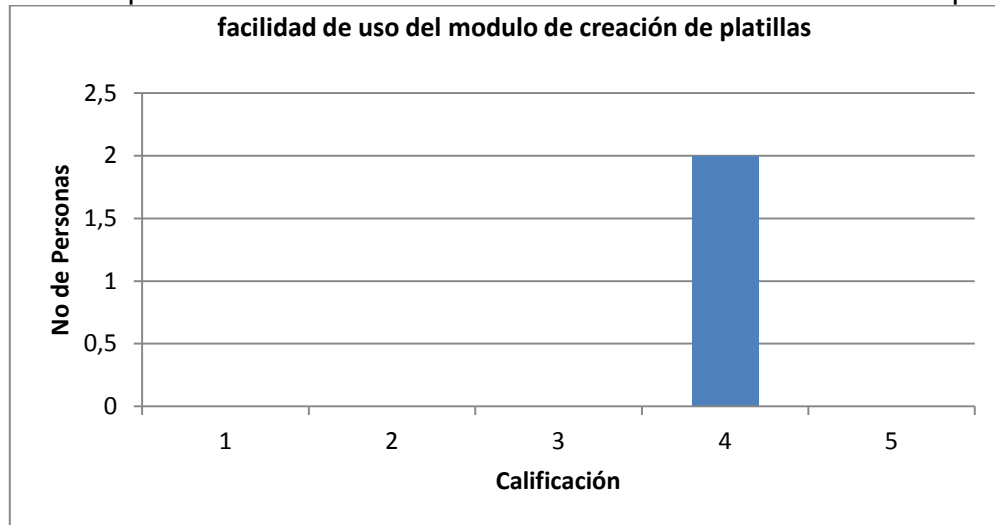
Por otra parte se encuentra que para el 90% de los participantes del Focus Group es conveniente la opción de compartir las plantillas de actividades y es oportuno el

desarrollo de funcionalidades que permitan calificar y registrar comentarios sobre las plantillas creadas, para obtener retroalimentación de las mismas.

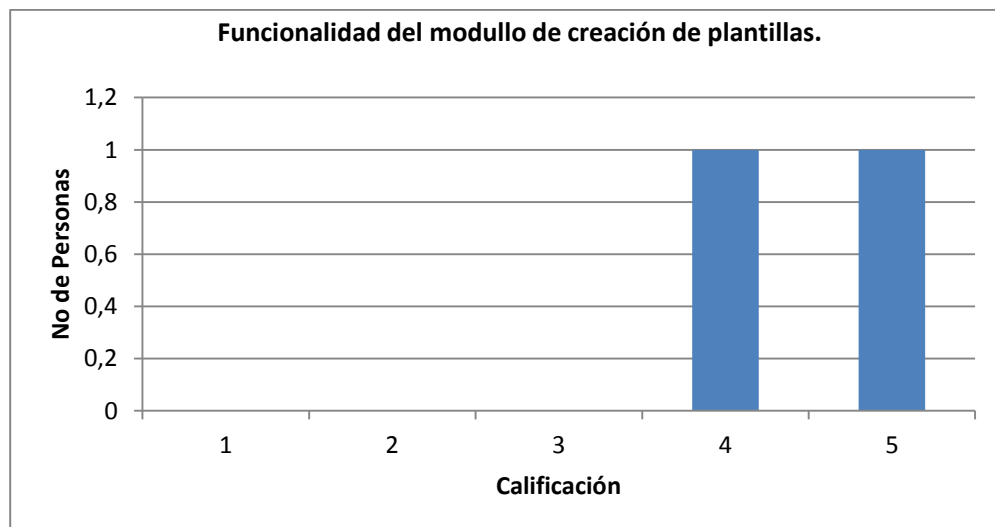
### Resultados del Módulo de Gestión de Usuarios y Cuentas.

Preguntas realizadas:

8. Califique de 1 a 5 la facilidad de uso del módulo de creación de plantillas



9. El módulo de creación de plantillas funciono de manera adecuada?



El módulo de creación de plantillas según la información entregada por los docentes participantes es fácil de usar y funciono de la manera adecuada, esta valoración se da a pesar de que durante la realización de la sesión ocurrió un error

de configuración del despliegue que impidió que las plantillas y las actividades pudieran ser creadas de manera remota, solo se pudo realizar desde el mismo equipo en el que estaba el servidor ElearningSync.

### Resultados relacionados al proceso de sincronización automática, sincronización por demanda, y posterior a las actualizaciones de las actividades por parte de los usuarios docentes

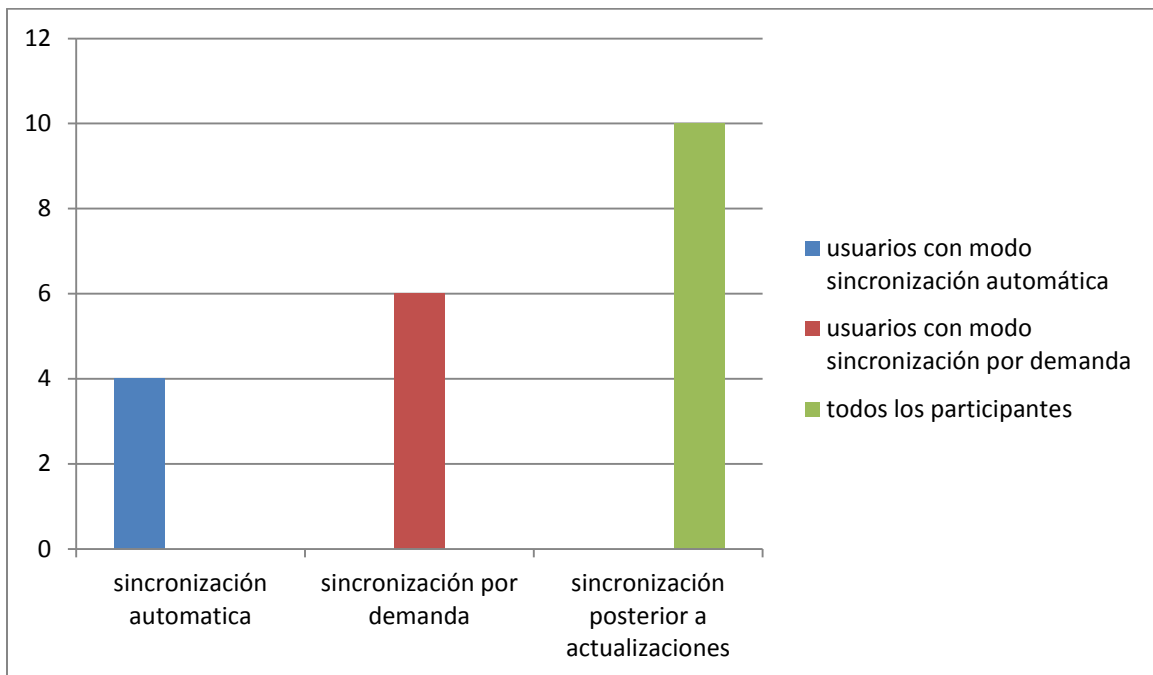
Preguntas realizadas:

11. El proceso de sincronización por demanda se ejecutó de manera exitosa?

11. El proceso de sincronización automática se ejecutó de manera exitosa?

13. El proceso de Sincronización posterior a una actualización se ejecutó de manera exitosa?

Vale la pena resaltar que el grupo de participantes fue segmentado de tal manera que cuatro usuarios tenían el modo de sincronización automática, cuatro usuarios el modo de sincronización por demanda y dos usuarios el modo no sincronizar.



Con la información obtenida por medio de las encuestas y contrastando con lo ocurrido durante la sesión de debate es totalmente válido decir que el proceso de sincronización implementado en el prototipo y el cual es el foco principal del

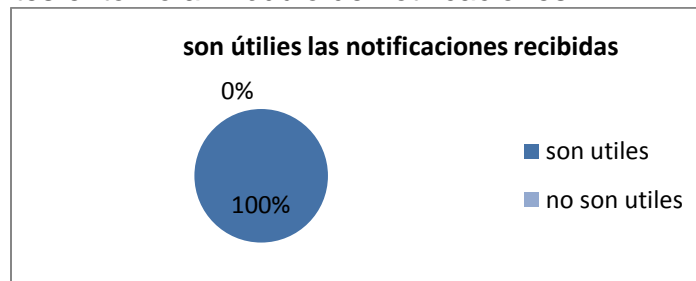
mismo, funciona en todos los casos de la manera adecuada ya que en los 3 escenarios de sincronización planteados el proceso funciona para 100% de los participantes.

### Resultados del Módulo de Notificaciones

Preguntas realizadas:

15. Le parece útil las notificaciones recibidas ante la creación de actividades y la modificación de las mismas?

16. si considera necesario, agregue las observaciones que crea convenientes entorno al módulo de notificaciones

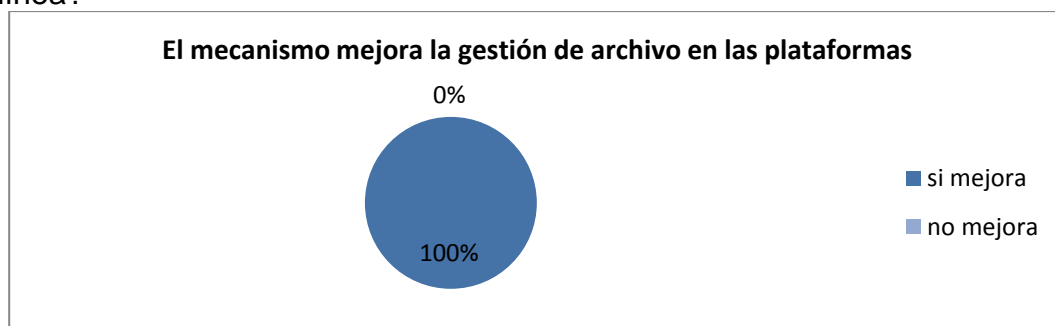


El 100% de los participantes del Focus Group considera útiles las notificaciones recibidas, pero por los comentarios realizados en la pregunta 16, y durante la realización del debate es posible establecer que es necesario establecer una estructura con mayor número de datos que permita a los usuarios reconocer con mayor facilidad que actividades y /o archivos han sido actualizados.

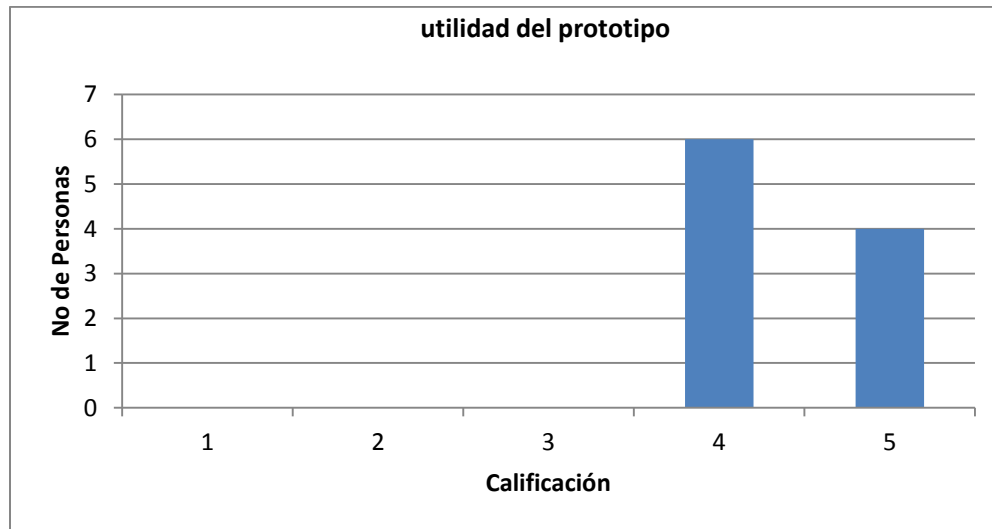
### Resultados sobre el mecanismo de sincronización y el prototipo en general

Preguntas realizadas:

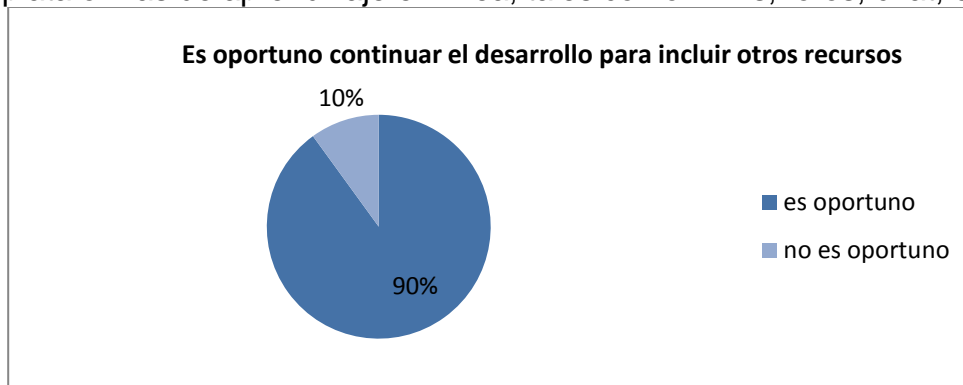
17. considera usted que el mecanismo planteado ofrece una mejor gestión de los archivos asociados a actividades en plataformas de aprendizaje en línea?



18. Califique de 1 a 5 la utilidad del prototipo desarrollado



19. Considera usted oportuno continuar el desarrollo del mecanismo y agregar funciones que permitan gestionar los demás recursos existentes en las plataformas de aprendizaje en línea, tales como: wikis, foros, chat, etc.



Según la información obtenida tanto en la encuesta como en el debate grupal de la sesión de Focus Group se puede concluir que el mecanismo propuesto mejora sustancialmente la gestión de archivos en plataformas de aprendizaje en línea, planteamiento con el cual el 100% de los participantes está de acuerdo, el prototipo desarrollado es útil pero se debe incluir algunas funcionalidades comentadas para lograr un mejor producto final, y según el 90% de los participantes es oportuno continuar el desarrollo del mecanismo de tal manera que se incluyan funcionalidades para la gestión de los demás recursos existentes tales como wikis, foros, chat, etc.

## CONCLUSIONES

Durante el desarrollo del trabajo se logra identificar que el proceso de sincronización de archivos no se encuentra desarrollado de forma totalmente robusta, es decir, a pesar de los avances obtenidos para este proceso, aún existe una brecha en el proceso de resolución de conflictos, dado que las estrategias diseñadas hasta el momento no han logrado satisfacer totalmente el problema generado cuando varios usuarios modifican de forma simultánea un mismo archivo. En la actualidad ni siquiera las herramientas más usadas tales como Dropbox, Google Drive, etc, han logrado solucionar este problema y aún existe lugar a pérdida de información y generación de réplicas conflictivas.

La arquitectura propuesta para el mecanismo es una primera aproximación para mejorar el proceso de gestión de archivos dentro de las plataformas de educación en línea, que debe ser mejorada y refinada por medio de la utilización de métodos de validación de arquitectura como ATAM u otros que permitan identificar los lugares más críticos de las misma y plantear soluciones que hagan esta arquitectura más robusta.

El prototipo desarrollado logra mejorar el proceso de gestión de archivos en las plataformas por medio de los procesos de sincronización automática y sincronización por demanda, pero es necesario continuar con el desarrollo del mismo para lograr mejorar otros atributos de calidad importantes tales como el desempeño, la usabilidad, la integración con distintas plataformas.

El concepto de plantillas de actividad generado representa una abstracción útil que ayuda a mejorar la realización de actividades de aprendizaje por medio de las plataformas de aprendizaje en línea, sin embargo, es necesario continuar trabajando en la estructuración del mismo, de tal manera que sea posible identificar por cada recurso el objetivo y características del mismo, además de construir una categorización de las plantillas de actividad, dado que dentro de la academia existe una gran variedad de actividades como actividades de preparación, de evaluación, de capacitación, etc.

El prototipo desarrollado logra cumplir hasta el momento con uno de los requisitos funcionales más importantes del mismo, la transparencia, dado que las acciones realizadas por medio del mismo son reflejadas dentro de la plataforma Moodle de forma transparente, un usuario que no utilice el prototipo desarrollado puede seguir accediendo a los recursos creados por medio de la plataforma.

El proceso de sincronización de archivos fue ampliado dentro del presente trabajo hacia un proceso de sincronización de actividades de usuario, es oportuno y necesario continuar con el proceso de implementación de funciones que permitan realizar la sincronización de los demás recursos existentes dentro de las plataformas de educación en línea tales como chats, foros, wikis, etc.

la validación realizada por medio del método de Focus Group permite determinar que las aplicaciones software desarrolladas dentro del presente trabajo logran realizar el proceso



de sincronización automática y sincronización por demanda de archivos de manera exitosa.

## **RECOMENDACIONES Y TRABAJO FUTURO**

Es importante continuar con el estudio de la propuesta de arquitectura presentada en el este trabajo y realizar procesos de la validación por medio de metodos como ATAM para lograr identificar y resolver los posibles problemas de la misma.

La solución implementada por medio del prototipo solo incluye una aplicación cliente que funciona sobre equipos con sistema operativos Windows, como trabajo futuro se propone implementar aplicaciones clientes compatibles con otros sistemas operativos tales como Linux, Mac, etc.

Es necesario buscar alternativas e implementar soluciones que mejoren el rendimiento del proceso de sincronización dado que hasta el momento solo se ha implementado el algoritmo más básico de este proceso lo que según el marco teórico puede ser poco eficiente.

El prototipo desarrollado fue diseñado con el objetivo de funcionar sobre la plataforma caso de estudio Moodle, sin embargo la arquitectura propuesta busca que el mecanismo pueda operar con otras plataformas de aprendizaje en línea, por lo que se define como trabajo futuro continuar el desarrollo de tal manera que sea posible lograr integrar otras plataformas de aprendizaje en línea.

Uno de los problemas identificados en el mecanismo es la creciente necesidad de espacio de almacenamiento por la gestión de archivos de varios usuarios, por lo que la arquitectura del mecanismo considera emplear servidores de almacenamiento remoto para reducir la carga de almacenamiento y procesamiento de los archivos, en el prototipo desarrollado no se realizó la implementación para el uso de dichos servidores, por lo que se presenta esta labor como trabajo futuro.

---

## REFERENCIAS

- [1] Deepak Gupta and Kalpana Sagar, "Remote File Synchronization Single-Round Algorithms." *International Journal of Computer Applications* (0975 – 8887) Volume 4 – No.1, Jul-2010.
- [2] Utku Irmak, Svilen Mihaylov, and Torsten Suel, "Improved Single-Round Protocols for Remote File Synchronization." .
- [3] S. Balasubramaniam and Benjamin C. Pierce, "What is a file synchronizer." .
- [4] Norman Ramsey and Elod Csirmaz, "An algebraic approach to file synchronization." .
- [5] A. Tridgell, "Efficient algorithms for sorting and synchronization," Australian National University Canberra, 1999.
- [6] Giovanni Tummarello, Christian Morbidoni, Reto Bachmann-Gmür, and Orri Erling, "RDFSyc: efficient remote synchronization of RDF models." .
- [7] Philipp C. Heckel, "Minimizing remote storage usage and synchronization time using deduplication and multichunking Syncany as an example," Manheim, 2012.
- [8] Torsten Suel and Nasir Memon, "Algorithms for Delta Compression and Remote File Synchronization." .
- [9] Benjamin C. Pierce and Jerome Vouillony, "What's in Unison A Formal Specification and Reference Implementation of a File Synchronizer." 2004.
- [10] Dropbox Inc., "Dropbox Website." [Online]. Available: <https://www.dropbox.com/>. [Accessed: 28-Dec-2015].
- [11] Dropbox Help, "¿Dropbox siempre carga y descarga todo el archivo cada vez que se implementa un cambio?" [Online]. Available: <https://www.dropbox.com/help/8/es>. [Accessed: 11-Jun-2014].
- [12] Mell, Pether and Grance, Timothy, "The NIST Definition of Cloud Computing Recommendations of the National Institute of Standards and Technology." .
- [13] Mesa Sectorial Cloud Computing, "Cloud Computing una perspectiva para Colombia." abril-2010.
- [14] Observatorio Nacional de las telecomunicaciones y de la SI, "Cloud Computing Retos Y Oportunidades." May-2012.
- [15] Rajkumar Buyyaa, Chee Shin Yeo, Srikumar Venugopala, James Broberg, and Ivona Brandic, "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility." .
- [16] Vaquero, Luis M, Merino Luis Rodero, Caceres Juan, and Maik Lindner, "A Break in the Clouds: Towards a Cloud Definition." .
- [17] "Cloud computing: Flexibility, Efficiency, Security." .
- [18] Rabindra K. Jena and P. K. Mahanti, "Computing in the Cloud: Concept and Trends." *International Review on Computers and Software (I.RE.CO.S.)*, Vol. 6, N. 1, Jan-2011.

- 
- [19] Safiriyu Eludiora, Olatunde Abiona, Ayodeji Oluwatope, Adeniran Oluwaranti, Clement Onime, and Lawrence Kehinde, "A User Identity Management Protocol for Cloud Computing Paradigm." *Int. J. Communications, Network and System Sciences*, 2011, 4, 152-163, Mar-2011.
- [20] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, Ion Stoica, and Matei Zaharia, "Above the Clouds: A Berkeley View of Cloud Computing." .
- [21] L. Wang, J. Tao, M. Kunze, A. C. Castellanos, D. Kramer, and W. Karl, "Scientific Cloud Computing: Early Definition and Experience.," in *HPCC*, 2008, vol. 8, pp. 825–830.
- [22] Rajkumar Buyya, James Broberg, and Andrzej Goscinski, "CLOUD COMPUTING Principles and Paradigms.pdf." John Wiley & Sons, Inc., 2011.
- [23] Jordi Adell, José Miguel Castellet, and José Pascual Gumbau, "Selección de un entorno virtual de enseñanza aprendizaje de código fuente abierto para la Universidad Jaume." .
- [24] O. U. Press, "Learn about Virtual Learning Environment/Course Management System content," *Online Resource Centres - Oxford University*. [Online]. Available: <http://global.oup.com/uk/orc/learnvle/#Whatis>. [Accessed: 28-Feb-2015].
- [25] Luis L. Martins and F. W. Kellermanns, "A Model of Business School Students' Acceptance of a Web-Based Course Management System," *Academy of Management Learning & Education*, Vol. 3, No. 1, pp. 7–26, Mar. 2004.
- [26] XarxaTIC, "Entornos virtuales de aprendizaje (EVAs)." [Online]. Available: <http://www.xarxatic.com/herramientas-2-0/entornos-virtuales-de-aprendizaje-evas/>. [Accessed: 02-Sep-2014].
- [27] "Effective use of virtual learning environments (VLEs)." [Online]. Available: <http://www.jiscinfonet.ac.uk/infokits/vle/>.
- [28] María Jesús García Alba, "Análisis Desarrollo De Extensiones Moodle," UNIVERSIDAD DE ALCALÁ, 2010.
- [29] Alejandro Palacios Provencio and Roberto Tornero Santamarina, "CS training." 2009.
- [30] Capterra, "Top LMS Software," 15-Oct-2014. [Online]. Available: Top LMS Software. [Accessed: 13-Apr-2015].
- [31] "Gestión de Archivos Moodle." [Online]. Available: <https://docs.moodle.org/all/es/coursefiles2>. [Accessed: 16-Apr-2015].
- [32] "Edmodo." [Online]. Available: <https://www.edmodo.com/>. [Accessed: 16-Apr-2015].
- [33] "Teacher's Guide to Edmodo." [Online]. Available: <https://support.edmodo.com/home#entries/55582854-teachers-guide-to-edmodo>. [Accessed: 18-Apr-2015].

- 
- [34] “Manual de Usuario Edmodo.” [Online]. Available: <http://craig.com.ar/biblioteca/Manual%20de%20Edmodo%20-%20Aula%20de%20Tecnolog%EDa.pdf>. [Accessed: 16-Apr-2015].
- [35] “Blackboard Learn.” [Online]. Available: <http://www.blackboard.com/Platforms/Learn/Products/Blackboard-Learn.aspx>. [Accessed: 17-Apr-2015].
- [36] “CourseFiles BlackBoard.” [Online]. Available: [http://help.blackboard.com/en-us/Learn/9.1\\_2014\\_04/Instructor/090\\_Course\\_Content/050\\_Course\\_Files](http://help.blackboard.com/en-us/Learn/9.1_2014_04/Instructor/090_Course_Content/050_Course_Files). [Accessed: 17-Apr-2015].
- [37] “Add Files to Course Files BlackBoard.” [Online]. Available: [http://help.blackboard.com/en-us/Learn/9.1\\_2014\\_04/Instructor/090\\_Course\\_Content/050\\_Course\\_Files/020\\_Add\\_Files\\_to\\_Course\\_Files](http://help.blackboard.com/en-us/Learn/9.1_2014_04/Instructor/090_Course_Content/050_Course_Files/020_Add_Files_to_Course_Files). [Accessed: 17-Apr-2015].
- [38] “SkillSoft.” [Online]. Available: <http://www.skillsoft.com/>. [Accessed: 18-Apr-2015].
- [39] “SkillSoft elearning-management-system.” [Online]. Available: <http://www.skillsoft.com/experience/learning-management-system.asp>. [Accessed: 18-Apr-2015].
- [40] “Schoolology.” [Online]. Available: <https://www.schoolology.com/home.php>. [Accessed: 19-Apr-2015].
- [41] “schoolology lms Integration.” [Online]. Available: <https://www.schoolology.com/lms-integration.php>.
- [42] “Schoolology LMS.” [Online]. Available: <https://www.schoolology.com/learning-management-system.php>. [Accessed: 19-Apr-2015].
- [43] “Matriz de Mejoras y Criterios LMS/CMS Edutools,” *EduTools Matrix*. [Online]. Available: <http://wcet.wiche.edu/learn/edutools>. [Accessed: 20-Aug-2014].
- [44] Oscar Comezaña Portilla and Francisco José García Peñalvo, “Plataformas para la educación basada en la web,” Departamento de Informática y Automática - Universidad de Salamanca, Informe Técnico, Enero de 2005.
- [45] Ambysoft Inc., “El Proceso Unificado Ágil (AUP).” [Online]. Available: <http://www.cc.una.ac.cr/AUP/>.
- [46] Ambysoft Inc., “El Proceso Unificado Ágil (AUP)-Disciplinas.” [Online]. Available: <http://www.cc.una.ac.cr/AUP/html/disciplines.html>.
- [47] Ambysoft Inc., “El Proceso Unificado Ágil (AUP)-Fases.” [Online]. Available: <http://www.cc.una.ac.cr/AUP/html/phases.html>.
- [48] Ambysoft Inc., “El Proceso Unificado Ágil (AUP)-Inicio.” [Online]. Available: <http://www.cc.una.ac.cr/AUP/html/inception.html>.
- [49] Ambysoft Inc., “El Proceso Unificado Ágil (AUP)-Elaboración.” [Online]. Available: <http://www.cc.una.ac.cr/AUP/html/elaboration.html>.

- 
- [50] Ambysoft Inc., “El Proceso Unificado Ágil (AUP)-Construcción.” [Online]. Available: <http://www.cc.una.ac.cr/AUP/html/construction.html>.
- [51] Ambysoft Inc., “El Proceso Unificado Ágil (AUP)-Transición.” [Online]. Available: <http://www.cc.una.ac.cr/AUP/html/transition.html>.
- [52] Carlos Mario Agredo Ordoñez and Carlos Javier Cepeda, “GESTIÓN DE ARCHIVOS MULTIMEDIA DE GRAN TAMAÑO BASADA EN EL MODELO CLOUD COMPUTING APLICADA A PLATAFORMAS DE EDUCACIÓN EN LÍNEA. CASO DE ESTUDIO MOODLE.pdf.” 2012.
- [53] W3C, “SOAP\_W3C,” SOAP. [Online]. Available: <http://www.w3.org/TR/soap12/>.
- [54] Luis Fernando Espino Barrios, “Cloud Computing como una Red de Servicios.pdf.” Nov-2009.
- [55] Leonard Richardson and Sam Ruby, “RESTful Web Services.pdf.” O’REILLI, May-2007.
- [56] Cake Software Foundation, Inc., “Cookbook 2.x.” [Online]. Available: <http://book.cakephp.org/2.0/en/contents.html>.
- [57] Rick Kazman, Mark Klein, and Paul Clements, “ATAM: Method for Architecture Evaluation.” Agosto de-2000.
- [58] Cake Software Foundation, Inc., “Cookbook 2.x - REST.” [Online]. Available: <http://book.cakephp.org/2.0/en/development/rest.html>.
- [59] Cake Software Foundation, Inc., “Cookbook 2.x - Authentication.” [Online]. Available: <http://book.cakephp.org/2.0/en/core-libraries/components/authentication.html>.
- [60] Twitter, “Bootstrap.” [Online]. Available: <http://getbootstrap.com/>.
- [61] Moodle, “Web services Roadmap.” [Online]. Available: [https://docs.moodle.org/dev/Web\\_services\\_Roadmap](https://docs.moodle.org/dev/Web_services_Roadmap). [Accessed: 11-Oct-2014].
- [62] Moodle, “Web services API.” [Online]. Available: [https://docs.moodle.org/dev/Web\\_services\\_API](https://docs.moodle.org/dev/Web_services_API). [Accessed: 12-Sep-2014].
- [63] Moodle, “Adding a Web Service to plugin.” [Online]. Available: [https://docs.moodle.org/dev/Adding\\_a\\_web\\_service\\_to\\_a\\_plugin](https://docs.moodle.org/dev/Adding_a_web_service_to_a_plugin). [Accessed: 13-Dec-2014].
- [64] Moodle, “Using Web Service.” [Online]. Available: [https://docs.moodle.org/28/en/Using\\_web\\_services](https://docs.moodle.org/28/en/Using_web_services). [Accessed: 13-Dec-2015].
- [65] Moodle, “Coding style.” .
- [66] Moodle, “SQL coding style.” [Online]. Available: [https://docs.moodle.org/dev/SQL\\_coding\\_style](https://docs.moodle.org/dev/SQL_coding_style). [Accessed: 17-Dec-2014].
- [67] Moodle, “Data Manipulation API.” [Online]. Available: [https://docs.moodle.org/dev/Data\\_manipulation\\_API](https://docs.moodle.org/dev/Data_manipulation_API). [Accessed: 15-Dec-2014].

- [68] Jyrki Kontio, Laura Lehtola, and Johanna Bragge, "Using the Focus Group Method in Software Engineering: Obtaining Practitioner and User Experiences." 2004.
- [69] MIGUEL MENDOZA-MORENO, CAROLINA GONZÁLEZ-SERRANO, and FRANCISCO J. PINO, "FOCUS GROUP COMO PROCESO EN INGENIERÍA DE SOFTWARE: UNA EXPERIENCIA DESDE LA PRÁCTICA." Agosto-2013.