



UNIVERSIDAD DE LA REPÚBLICA  
FACULTAD DE INGENIERÍA



# Interoperabilidad Matefun-Moodle

Ramiro Rodriguez

## TUTORES

Marcos Viera ..... Universidad de la República  
Gonzalo Tejera ..... Universidad de la República

# Resumen

Matefun es un lenguaje de programación de paradigma funcional desarrollado en Haskell por los tutores de este proyecto de grado con la intención de facilitar la enseñanza de las herramientas matemáticas y lenguajes de programación a estudiantes de enseñanza secundaria. Por otro lado dentro de un proyecto/producto de mayor escala, mediante distintos proyectos de grado Matefun se ha ido extendiendo logrando al día de hoy tener una interfaz web que maneja sus propios archivos y manejo de gráficas en dos y tres dimensiones además de manejar internacionalización (i18n).

En los últimos años la educación secundaria y superior ha experimentado un proceso profundo de cambio en su estructura y funcionamiento, instalar y utilizar un *sistema de gestión de aprendizaje* (LMS) hoy en día es cada vez más común y las plataformas de gestión del aprendizaje han ganado presencia e importancia en casi todas las instituciones a varios niveles incluso en nuestro país. Esto se ha incrementado de forma abrumadora luego de las restricciones a la movilidad impuestas por los distintos gobiernos a partir de inicios del 2020 en el marco de la pandemia ocasionada por el virus SARS-CoV-2.

En este marco se presenta este proyecto de grado para lograr una extensión a Matefun que logre conectarse a los sistemas de gestión de aprendizaje que pueden tener los centros educativos para lograr tener un mejor seguimiento de las actividades de los alumnos utilizando las plataformas de LMS pre instaladas por los centros.

Se intenta conectar Matefun a sistemas externos mediante protocolos utilizados en la industria para que nos permita realizar las tareas habituales del aula logrando respaldo en el sistema LMS del centro educativo. A modo de ejemplo, estas tareas pueden ser manejo de archivos privados, compartir archivos, entregar y corregir evaluaciones.

**Palabras clave:** Sistemas de gestión de aprendizaje, Moodle, Interoperabilidad, Web services, REST, HTTP, Angular, JavaEE

# Tabla de contenidos

<b>Resumen</b>	<b>2</b>
<b>1. Introducción</b>	<b>5</b>
1.1. Motivación y Objetivos . . . . .	5
1.2. Organización de este documento . . . . .	5
1.2.1. Glosario . . . . .	6
<b>2. Antecedentes</b>	<b>7</b>
2.1. Proyecto Matefun original . . . . .	7
2.2. Learning Management Systems . . . . .	7
2.2.1. Descripciones generales . . . . .	8
2.2.2. Edmodo . . . . .	9
2.2.3. Moodle . . . . .	9
2.2.4. Sakai Project . . . . .	9
2.2.5. Schoology . . . . .	10
2.2.6. Google Classroom . . . . .	10
2.3. Tecnologías y herramientas utilizadas . . . . .	11
2.3.1. Javascript . . . . .	11
2.3.2. TypeScript . . . . .	11
2.3.3. HTML5 & CSS3 . . . . .	12
2.3.4. Angular . . . . .	13
2.3.5. Java (EE) y JPA . . . . .	13
2.3.6. Git/GitLab . . . . .	14
2.4. Conceptos importantes . . . . .	15
2.4.1. Interoperabilidad . . . . .	15
2.4.2. Servicios web . . . . .	15
2.4.3. REST . . . . .	16
2.4.4. HTTP . . . . .	17
<b>3. Integración Matefun-Moodle</b>	<b>18</b>
3.1. Por qué Moodle? . . . . .	18
3.2. Arquitectura y diseño . . . . .	19
3.2.1. Cambios en arquitectura y diseño . . . . .	19
3.3. Análisis de requerimientos . . . . .	20
3.4. Proceso de desarrollo . . . . .	22
3.5. Implementación . . . . .	24
3.5.1. Existencia de usuario administrador . . . . .	24

## Tabla de contenidos

3.5.2. Gestión de centros educativos . . . . .	25
3.5.3. Mapeo de archivos de usuario . . . . .	26
3.5.4. Compartir archivos . . . . .	28
3.5.5. Manejo de archivos compartidos y bloqueos . . . . .	29
3.5.6. Corrección de evaluaciones . . . . .	31
3.5.7. Borrado de archivos . . . . .	32
3.5.8. Estado de subsistemas . . . . .	32
<b>4. Experimentación</b>	<b>37</b>
4.1. Ambiente de pruebas . . . . .	38
4.2. Deploy en AWS . . . . .	38
4.3. Casos de prueba . . . . .	39
4.3.1. Login . . . . .	40
4.3.2. Gestión de centros educativos . . . . .	40
4.3.3. Mapeo de archivos y carpetas . . . . .	41
4.3.4. Gestión de archivos compartidos . . . . .	42
4.3.5. Entrega de archivos para assignments . . . . .	43
4.3.6. Listado y corrección de assignments . . . . .	44
4.4. Pruebas de regresión . . . . .	44
<b>5. Conclusiones y trabajo futuro</b>	<b>51</b>
5.1. Conclusiones . . . . .	51
5.2. Trabajo Futuro . . . . .	52
5.2.1. Actualización de dependencias . . . . .	52
5.2.2. Movimientos/eliminaciones físicas . . . . .	53
5.2.3. Merge de cambios de archivos compartidos utilizando git . . . . .	53
5.2.4. Compartir archivos utilizando plugin/addon . . . . .	53
5.2.5. Soporte Schoology . . . . .	54
<b>6. Anexos</b>	<b>55</b>
6.1. Modelo Entidad-Relación . . . . .	55
6.2. Casos de prueba realizados . . . . .	56
6.2.1. Login . . . . .	56
6.2.2. Gestión de centros educativos . . . . .	57
6.2.3. Mapeo de archivos y carpetas . . . . .	58
6.2.4. Gestión de archivos compartidos . . . . .	59
6.2.5. Entrega de archivos para assignments . . . . .	61
6.2.6. Listado y corrección de assignments . . . . .	62
<b>Índice de tablas</b>	<b>65</b>
<b>Índice de figuras</b>	<b>66</b>
<b>Referencias</b>	<b>72</b>

# Capítulo 1

## Introducción

### 1.1. Motivación y Objetivos

La interfaz web de Matefun fue desarrollada para tener una forma más interactiva e intuitiva de manejar el lenguaje Matefun. El desarrollo de esta interfaz web y el de este lenguaje fue pensado para lograr un mayor acercamiento de los estudiantes de enseñanza secundaria a herramientas de matemática computacional y programación.

Por esta razón también es que el proyecto Matefun original desarrolló un plan piloto de utilización en el colegio José Pedro Varela y presentado en congresos de educación en el interior del país [28].

En el proyecto original de Matefun al momento de hacer login de un alumno/docente se verificaba la existencia de usuario y contraseña en una instancia de Moodle (en caso de haber sido configurado de esta manera) para poder iniciar sesión en Matefun, dado esto y también al haber sido plasmado como posible trabajo futuro es que surge la motivación de esta integración con algún Sistema de Gestión de Aprendizaje (LMS por su sigla en inglés por *Learning Management System*) del mercado luego de investigar la mejor opción para no solamente implementar un caso de uso de inicio de sesión sino también de todas las actividades relacionadas con el uso en el aula, tales como entregar/corregir evaluaciones, etc.

Queda el objetivo planteado que toda funcionalidad que previamente tenga Matefun se pueda realizar dejando respaldo de las actividades en el LMS a diferencia del proyecto original que utiliza su propia base de datos. También se han incluido nuevos requerimientos que no tienen relación con los sistemas LMS como es la existencia de una sección en la interfaz web para verificar el estado de servidores y del uso de la plataforma.

### 1.2. Organización de este documento

Inicialmente, en el *capítulo siguiente* se presentaran los antecedentes y conceptos necesarios que fue necesario obtener o reforzar para abordar este proyecto, tales como la introducción del proyecto original, los sistemas LMS evaluados y

## Capítulo 1. Introducción

las tecnologías/herramientas utilizadas durante la implementación.

Luego en el *capítulo 3*, la parte central, se presenta la justificación técnica del LMS elegido, detalles globales de la arquitectura y cambios en la misma así como también la explicación de la implementación de los distintos requerimientos. En la experimentación presentada en el *capítulo 4* se muestran algunos de las distintas pruebas realizadas para verificar la correctitud de los requerimientos.

Las conclusiones obtenidas al final se presentan en el *capítulo 5* junto a ideas de trabajo futuro que fueron surgiendo en la medida que este proyecto avanzaba, luego en el *capítulo 6* se presentan distintos anexos con diagramas y un resumen de los casos de prueba completados.

Se finaliza este documento con índices de tablas y figuras y las referencias recopiladas a lo largo del proyecto.

### 1.2.1. Glosario

A continuación se presentan siglas y términos utilizados a lo largo de este documento.

1. **AJAX**: acrónimo para **A**synchronous **J**avaScript **A**nd **X**ML.
2. **API**: acrónimo para **A**pplication **P**rogramming **I**nterface.
3. **Assignment**, **actividad** ó **activity**: hace referencia a un trabajo que un docente coloca en un curso de Moodle (*tarea* en español).
4. **CSS**: acrónimo para **C**ascading **S**tyle **S**heets.
5. **HTML**: acrónimo para **H**yper**T**ext **M**arkup **L**anguage.
6. **HTTP**: acrónimo para **H**ypertext **T**ransfer **P**rotocol.
7. **IP**: acrónimo para **I**nternet **P**rotocol, si bien el acrónimo hace referencia al protocolo de capa de red, en este documento se refiere simplemente a una dirección IP.
8. **JavaEE**: **J**ava **E**nterprise **E**dition.
9. **LMS**: acrónimo para **L**earning **M**anagement **S**ystem (o Sistema de Gestión de Aprendizaje).
10. **MOOC**: acrónimo para **M**assive **O**pen **O**nline **C**ourse, Curso en Línea Masivo y Abierto.
11. **Moodle**: acrónimo para **M**odular **O**bject-**O**riented **D**ynamic **L**earning **E**nvironment [38].
12. **REST**: acrónimo para **R**epresentational **S**tate **T**ransfer.
13. **SOAP**: acrónimo para **S**imple **O**bject **A**ccess **P**rotocol.
14. **Submission**: hace referencia a la entrega que hace un alumno a un assignment en Moodle.

## Capítulo 2

# Antecedentes

A continuación se describirán los conceptos y herramientas que fue necesario entender y aprender a utilizar para la realización satisfactoria de este proyecto.

### 2.1. Proyecto Matefun original

Originalmente el proyecto Matefun fue realizado como proyecto de grado para la carrera de Ingeniería en Computación de esta Facultad. Consiste en una interfaz web y servidor Backend que permite interactuar con un compilador de un lenguaje de programación previamente desarrollado por los tutores de este proyecto y del proyecto original.

Matefun es un lenguaje de paradigma funcional que permite el acercamiento a las herramientas matemáticas y a la programación a estudiantes de enseñanza secundaria.

La interfaz web permite ejecutar los archivos tipo *.mf* que contienen el código fuente Matefun escrito por los estudiantes y/o docentes, así como también compartir y corregir tareas, compartir archivos, gestionar archivos privados y compartidos.

Los centros educativos habitualmente pueden llegar a utilizar distintos tipos de software para gestionar cursos, trabajos y/o archivos de manera virtual desde los docentes a los alumnos, son conocidos como *Learning Management Systems*. La primera intención del actual proyecto es lograr la interoperabilidad de las funcionalidades preexistentes de Matefun con algún tipo de LMS que se seleccione. Información adicional sobre el proyecto Matefun original puede encontrarse en el documento de estado del arte del presente proyecto [64].

A continuación se presentan algunos de los evaluados para dicha tarea.

### 2.2. Learning Management Systems

*LMS* es el acrónimo de *Learning Management System* (en español, sistema de gestión de aprendizaje). Este tipo de software ayuda a la gestión, seguimiento y documentación de cursos cuando queremos verlo desde el punto de vista puramente educacional (centros educativos en general) o manejo de programas

## Capítulo 2. Antecedentes

de entrenamiento que podrían brindar las empresas a sus empleados a modo de ejemplo.

Ayudan al usuario que tiene rol de instructor a entregar material a los estudiantes, administrar exámenes, realizar un seguimiento del progreso del alumno y administrar el mantenimiento de registros.

Los LMS se centran en la entrega de aprendizaje en línea, pero admiten una variedad de usos, actuando como una plataforma para cursos totalmente en línea, así como varias formas híbridas, como el aprendizaje mixto y aulas semi-presenciales.

También se pueden complementar con otras tecnologías de aprendizaje, como un sistema de gestión de capacitación para administrar la capacitación dirigida por un instructor o una tienda de registros de aprendizaje (Learning Record Store (*LRS*)) para almacenar y rastrear datos de aprendizaje.

Se intenta llevar una similitud con el concepto de *aula virtual* con este se quiere decir que las tareas relevantes a un docente dentro del aula se emulan dentro del sistema, a modo de ejemplo sin tratar de ser exhaustivos se enumeran la creación y corrección de evaluaciones y cuestionarios, gestión de grupos de alumnos para trabajos en conjunto y publicación de materiales de estudio para estudiantes pertenecientes al sistema.

Este tipos de sistemas buscan la gestión del aprendizaje en línea aunque puede no ser de forma exclusiva, también son utilizados para brindar una experiencia educativa más adaptada a ambientes semi-presenciales. Información adicional sobre sistemas LMS puede encontrarse en el documento de estado del arte del presente proyecto [64].

### 2.2.1. Descripciones generales

Un sistema de gestión del aprendizaje es una aplicación de software para la administración, documentación, seguimiento, informes, automatización e impartición de cursos educativos, programas de formación o programas de aprendizaje y desarrollo.

El concepto de sistema de gestión del aprendizaje surgió directamente de e-Learning. Aunque el primer LMS apareció en el sector de la educación superior, la mayoría de los LMS de hoy se centran en el mercado corporativo. La primera introducción del LMS fue a fines de la década de 1990 [4].

Los sistemas de gestión del aprendizaje se diseñaron para identificar la formación y las brechas de aprendizaje, utilizando datos analíticos e informes. Los LMS se centran en la entrega de aprendizaje en línea, pero admiten una variedad de usos, actuando como una plataforma para el contenido en línea, incluidos los cursos, tanto de base asincrónica como sincrónica.

Un LMS puede ofrecer administración de aulas para capacitación dirigida por un instructor o un aula invertida, que se usa en la educación superior, pero no en el espacio corporativo. Los LMS modernos incluyen algoritmos inteligentes para realizar recomendaciones automatizadas de cursos basados en el perfil de habilidades del usuario, así como para extraer metadatos de los materiales de aprendizaje para que dichas recomendaciones sean aún más precisas.

A continuación se presentaran algunos ejemplos de este tipo de software, luego de los ejemplos se presenta la tabla 2.1 comparando dichas soluciones.

### 2.2.2. Edmodo

Creada en el año 2008 por Jeff O'Hara y Nic Borg [21], Edmodo es una plataforma educativa gratuita [24] digital que permite involucrar a los estudiantes, administrar el aula y compartir materiales, haciendo accesible el aprendizaje desde cualquier lugar [34].

También puede ser concebida como una red social educativa o herramienta de microblogging que conecta a los alumnos con las personas y recursos del aula, hacer asignaciones (tareas) y mantener una interacción activa y constante entre profesores, alumnos y padres.

En marzo de 2015, Noodle [50] [6] nombró a Edmodo como una de las *32 herramientas educativas en línea más innovadoras* [49].

### 2.2.3. Moodle

En 2002 Martín Dougiamas lanza Moodle (acrónimo de entorno de aprendizaje dinámico orientado a objetos modular), basado bajo la General Public License y escrito en PHP [44] [39].

Una de las fortalezas de Moodle es que es software libre, esto significa que su creador inicial, al momento de su publicación, decidió utilizar la Licencia Pública GNU (GPL) y por lo tanto puede ser utilizado sin pagar licencias. La institución que lo instale está autorizada a copiar, usar y modificar Moodle.

Moodle ofrece las herramientas necesarias para crear sitios web privados con cursos en línea para educadores utilizando características de administración personalizables, también cuenta con una API pública (REST/SOAP) [45] que permite realizar de forma remota la mayoría de las tareas que permite realizar desde la interfaz web directamente logrando así interoperabilidad con otros sistemas.

Puede ser instalado en cualquier sistema operativo que pueda ejecutar PHP y pueda soportar una base de datos tipo relacional (por ejemplo MySQL). Se ejecuta sin modificaciones en Unix, GNU/Linux, OpenSolaris, FreeBSD, Windows, Mac OS X, NetWare y otros sistemas que soportan PHP, incluyendo la mayoría de proveedores de hosting o platform as a service (*PaaS*) como Amazon Web Services.

### 2.2.4. Sakai Project

Inicialmente lanzado en marzo del 2005, Sakai es una plataforma multilingüe de gestión de aprendizaje, comunicaciones y colaboración de código abierto y forma parte de la fundación Apereo [2].

Técnicamente, Sakai es una aplicación web orientada a servicios y basada en Java que proporciona una variedad de capacidades que respaldan la enseñanza y el aprendizaje. Por lo general, Sakai se despliega utilizando Apache Tomcat como su contenedor de servlets y la escalabilidad se logra ejecutando varias instancias de Tomcat en un entorno agrupado en un cluster, cada una de las cuales implementa una copia de Sakai [57].

Se integra con una variedad de servicios de autenticación externos, incluidos CAS, Kerberos, LDAP, Shibboleth y WebAuth. Una sola base de datos,

## Capítulo 2. Antecedentes

generalmente MySQL u Oracle, proporciona un almacén transaccional de información, mientras que el almacenamiento de archivos generalmente se delega a las soluciones de tipo SAN (Storage Area Network) o NAS (Network Attached Storage) [57].

Sakai tiene la opción de descargar un paquete para instalación y obtener su propia instancia, así como también servicios de cloud-hosting que resuelven ese trabajo [70].

Esta licenciado bajo Educational Community License, v2.0 [57] [74].

### 2.2.5. Schoology

Schoology es un entorno de aprendizaje virtual y un servicio de redes sociales para escuelas e instituciones de educación superior que permite a los usuarios crear, administrar y compartir contenido académico. Ésta plataforma en línea proporciona las herramientas necesarias para gestionar una lección en el aula virtual.

Schoology permite a los maestros administrar las clases de forma remota mediante la publicación de recordatorios o actualizaciones diarias, enviar mensajes directamente a los estudiantes y administrar las tareas programadas.

Las asignaciones pueden incluir cuestionarios, formularios, enlaces a sitios web/videos y debates para que los estudiantes compartan ideas. Los estudiantes pueden unirse a un curso usando un código de acceso o ser agregados a una clase directamente por un profesor.

### 2.2.6. Google Classroom

Google Classroom es un servicio web desarrollado por Google para escuelas que tiene como objetivo simplificar la creación, distribución y calificación de tareas.

El objetivo principal de Google Classroom es agilizar el proceso de compartir archivos entre profesores y alumnos.

Google Classroom integra *Docs*, *Sheets*, *Slides*, *Gmail* y *Calendar* (todos servicios de Google) en una plataforma para administrar la comunicación entre estudiantes y profesores.

Se puede invitar a los estudiantes a unirse a una clase a través de un código privado o se pueden importar automáticamente desde un dominio de la escuela. Los profesores pueden crear, distribuir y marcar tareas, todo dentro del ecosistema de Google. Cada clase crea una carpeta separada en el Drive del usuario respectivo, donde el alumno puede enviar el trabajo para que lo califique un profesor.

Las tareas entregables y las fechas de vencimiento de entrega se agregan al calendario de Google, cada tarea puede pertenecer a una categoría (o tema). Los maestros pueden monitorear el progreso de cada estudiante revisando el historial de revisión de un documento, y después de ser calificados, los maestros pueden devolver el trabajo junto con comentarios y calificaciones.

## 2.3. Tecnologías y herramientas utilizadas

	<b>Edmodo</b>	<b>Moodle</b>	<b>Sakai</b>	<b>Schoology</b>	<b>Classroom</b>
Online	si	si [41]	si [70]	si	si
Descargable	no	si [43]	si [70]	no	no
Gratuita	si [24]	si	si [19]	no [46]	no [5]
Software libre	N/A	si [27]	si [56]	no [72]	no [30]
Licencia	N/A	gpl [39]	ECL2.0 [57]	derechos reservados [72]	no [30]
API	rest [18] [54]	rest/soap [45]	soap [53] [55]	rest [71]	rest [14]
Cuota de mercado	1.8 % [22]	28 % [22]	0.94 % [22]	1.94 % [22]	3 % [22]

Tabla 2.1: Tabla comparativa de sistemas LMS.

## 2.3. Tecnologías y herramientas utilizadas

En la presente sección se describirán las tecnologías y herramientas utilizadas tanto heredadas como introducidas para la elaboración de este proyecto.

### 2.3.1. Javascript

JavaScript (JS), es un lenguaje de programación que se ajusta a la especificación ECMAScript [37]. Es un lenguaje de alto nivel y de múltiple paradigma, admite estilos de programación imperativos, funcionales y controlados por eventos. Tiene API's para trabajar con texto, fechas, expresiones regulares, estructuras de datos estándar y el manejo del *Document Object Model* (DOM).

En conjunto con HTML y CSS, JavaScript es una de las tecnologías centrales de la World Wide Web. Más del 97% de los sitios web lo utilizan del lado del cliente para el comportamiento de las páginas web [75] y a menudo incorporan bibliotecas de terceros [76]. Todos los principales navegadores web tienen un motor de JavaScript dedicado para ejecutar el código en el dispositivo del usuario.

Los motores de JS se usaban originalmente solo en navegadores web, pero ahora son componentes centrales de otros sistemas de software, principalmente servidores y una variedad de aplicaciones, esto se logró más recientemente con la introducción al mercado de NodeJS.

En la elaboración de este proyecto Javascript es una de las tecnologías centrales, si bien obviamente se heredó del proyecto Matefun original también fue utilizado extensivamente por el autor de este proyecto de integración mediante la utilización de la plataforma Angular.

### 2.3.2. TypeScript

TypeScript (TS) es un lenguaje de programación desarrollado y mantenido por Microsoft. Es un *superset* sintáctico de JavaScript [80] y agrega tipado estático opcional al lenguaje.

TypeScript está diseñado para el desarrollo de aplicaciones web o Backend, se traduce a JavaScript utilizando herramientas como por ejemplo Babel [3]. Como TypeScript es un superconjunto de JavaScript, los programas JavaScript existentes también son programas TypeScript válidos.

## Capítulo 2. Antecedentes

TypeScript se puede utilizar para desarrollar aplicaciones JavaScript para la ejecución del lado del cliente y del lado del servidor (como con Node.js [48] o Deno [13]). Hay varias opciones disponibles para la transcompilación [3]. Se puede utilizar el comprobador de TypeScript predeterminado [25] [7], o se puede invocar el compilador de Babel para convertir TS a JS.

En este proyecto TypeScript se usa para definir tipos a los parámetros/-variables/constantes y así facilitar el proceso de desarrollo obteniendo errores en tiempo de traducción (a Javascript) ahorrándose estos errores en tiempo de ejecución facilitando así la depuración (debugging).

### 2.3.3. HTML5 & CSS3

Para finalizar las tecnologías básicas y más importantes utilizadas en este proyecto se hace una breve presentación de HTML es su versión 5 [32] y CSS en su última versión 3 [8].

#### HTML5

HTML5 es la última especificación del lenguaje HTML y representó una ruptura importante con las prácticas de marcado anteriores. El propósito de los cambios profundos en el lenguaje fue estandarizar las muchas formas nuevas en las que los desarrolladores lo estaban usando, así como fomentar un conjunto único de mejores prácticas con respecto al desarrollo web.

La mayoría de los cambios individuales son el resultado de objetivos más amplios en el diseño del lenguaje. Estos objetivos incluyen principalmente:

- Fomentar el *markup* semántico, es decir ahora las etiquetas tienen significado.
- Separa el diseño del contenido
- Promover la accesibilidad y la capacidad de respuesta del diseño.
- Reducir la superposición entre HTML, CSS y JS
- Admite experiencias de medios enriquecidos al tiempo que elimina la necesidad de complementos como Flash o Java.

#### CSS3

CSS es un lenguaje de hojas de estilo que se utiliza para describir la presentación de un documento escrito en un lenguaje de marcado como HTML. CSS es una tecnología fundamental de la World Wide Web junto con HTML y JavaScript.

CSS está diseñado para permitir la separación de la presentación y el contenido, incluido el diseño, los colores y las fuentes. Esta separación puede mejorar la accesibilidad del contenido, proporcionar más flexibilidad y control en la especificación de las características de la presentación, permitir que varias páginas web compartan el formato especificando el CSS relevante en un archivo *.css* separado, lo que reduce la complejidad y la repetición en el contenido estructural,

## 2.3. Tecnologías y herramientas utilizadas

además de habilitar el archivo `.css` se almacenará en caché para mejorar la velocidad de carga de la página entre las páginas que comparten el archivo y su formato.

CSS3 (su última versión) introduce nuevas funcionalidades que simplifican las tareas que en versiones anteriores se veían más engorrosas de realizar, a modo de ejemplo:

- Definir variables y utilizar sus valores en otras propiedades
- Restringir la definición de estilos a un ancho de pantalla (media queries)
- Introducción de transiciones, transformaciones, animaciones, sombras, gradientes
- Cálculo de valores (`calc(...)`)
- Nuevos selectores: `:is`, `:where`, `:not`, `:first-of-type`, etc.

### 2.3.4. Angular

Angular junto con Java es de las tecnologías más utilizadas en este proyecto, es desde donde se basa la completitud del desarrollo. Angular en este caso ha sido utilizado para el completo desarrollo de la interfaz de usuario que se carga desde el navegador web seleccionado. Angular ha sido heredado por este proyecto desde su implementación original de Matfeun en su versión 4, al momento de la edición de este documento Angular ya ha sido lanzado en su versión 12 [29].

Angular es un framework de código abierto mantenido por Google que permite crear aplicaciones frontend en HTML/CSS y JS. Con Angular se pueden crear aplicaciones del estilo *single page application*, aplicaciones que modifican dinámicamente la página visualizada, reescribiendo parte del HTML según la interacción del usuario utilizando request asíncronos con AJAX, en lugar de recargar todo el HTML como inicialmente se acostumbraba, esto aumenta la performance global además de reducir la carga del servidor.

### 2.3.5. Java (EE) y JPA

Java Platform Enterprise Edition es una colección de API's de Java propiedad actualmente de Oracle que se puede utilizar para escribir aplicaciones del lado del servidor (Backend), anteriormente era conocido como Java 2 Platform, Enterprise Edition o J2EE.

Al inicio Sun Microsystems desarrolló originalmente JavaEE para simplificar la creación de aplicaciones en un entorno de niveles de clientes ligeros. JavaEE simplifica el desarrollo de aplicaciones, reduce la necesidad de programación al crear componentes modulares estandarizados y reutilizables.

Las aplicaciones Java EE se pueden alojar en servidores de aplicaciones tales como *IBM WebSphere*, *GlassFish* o *Red Hat WildFly* (como es el caso de Matfeun), todos los cuales se ejecutan en la nube o dentro de un centro de datos corporativo.

## Capítulo 2. Antecedentes

Si bien las aplicaciones Java EE están alojadas en el lado del servidor, los ejemplos de clientes Java EE pueden ser dispositivos *IoT*, *smartphones* o *web-services RESTful* entre otros.

En el caso de Matefun se utiliza JavaEE para implementar y exponer una API rest que es utilizada por el cliente Angular de Matefun, así como mantener la comunicación con las instancias de Moodle que representan a los centros educativos que se ingresan al sistema (también desde una API REST pero que Moodle provee y Matefun consume). Esta aplicación corre en el servidor Wildfly versión 10 que se encuentra alojado en una instancia en *Amazon web services* (proveedor de infraestructura).

### JPA

La Java persistence API (*JPA*) se ocupa de la persistencia de los datos de las aplicaciones, que en términos generales significa cualquier mecanismo mediante el cual los objetos de Java sobreviven al proceso de aplicación que los creó. Habitualmente no es necesario conservar todos los objetos, pero en general las aplicaciones conservan los objetos clave. La especificación JPA le permite definir qué objetos se deben persistir y cómo se deben persistir esos objetos en la aplicación.

Por sí mismo, JPA no es una herramienta o framework; más bien, define un conjunto de conceptos que pueden ser implementados por cualquier herramienta o framework. Si bien el modelo de mapeo relacional de objetos (**Object-Relational Mapping**, **ORM**) de JPA se basó originalmente en Hibernate, desde entonces ha evolucionado. Del mismo modo, aunque JPA se diseñó originalmente para su uso con bases de datos relacionales basados en SQL, algunas implementaciones de JPA se han ampliado para su uso con bases de datos NoSQL. Un framework popular que admite JPA con NoSQL es EclipseLink, que es justamente la implementación utilizada en este proyecto para persistir centros educativos, registros, archivos de invitado, etc.

### 2.3.6. Git/GitLab

Git es un software para rastrear cambios en cualquier conjunto de archivos, generalmente utilizado para coordinar el trabajo entre programadores que desarrollan código fuente de manera colaborativa durante el desarrollo del software.

Sus objetivos incluyen la velocidad, la integridad de los datos y la compatibilidad con flujos de trabajo distribuidos y no lineales (ej: creación de ramas (*branches*) paralelas que se ejecutan en diferentes sistemas).

Git fue creado por Linus Torvalds en 2005 para el desarrollo del kernel de Linux, con otros desarrolladores de kernel contribuyendo a su desarrollo inicial. Desde 2005, Junio Hamano ha sido el mantenedor principal. Al igual que con la mayoría de los otros sistemas de control de versiones distribuidos, y a diferencia de la mayoría de los sistemas cliente-servidor, cada directorio Git en cada computadora es un repositorio completo con historial completo y capacidades completas de seguimiento de versiones, independientemente del acceso a la red o un servidor central.

## 2.4. Conceptos importantes

Git es un software gratuito y de código abierto distribuido bajo GNU General Public License Version 2.

GitLab es una herramienta de control de versiones y *DevOps* desarrollada como aplicación web que proporciona un administrador de repositorio Git junto a funciones wiki, seguimiento de problemas e integración continua y canalización de implementación, utilizando una licencia de código abierto, desarrollada por *GitLab Inc.* por los desarrolladores ucranianos Dmitriy Zaporozhets y Valery Sizov. En este proyecto fue utilizado como repositorio centralizado de código fuente y funcionalidad wiki provisto por la Facultad de Ingeniería.

## 2.4. Conceptos importantes

Luego de la tecnologías y software, a continuación se enumeran algunos conceptos importantes que si bien no son tecnologías, fueron necesarias para realizar este proyecto como ser protocolos, definiciones, estándares, etc.

### 2.4.1. Interoperabilidad

La interoperabilidad se define como la capacidad de los programas informáticos de comunicarse entre sí para el intercambio y el proceso de información efectivos (ya sean datos, documentos u otros objetos digitales, de manera uniforme y eficiente). El propósito de la interoperabilidad es lograr que los diferentes sistemas puedan comprender y procesar la información que se transmiten entre sí.

Para lograr el intercambio, deben contar con características técnicas y de estructura específicas que faciliten la comunicación; así como condiciones físicas y de software que les proporcionen estabilidad y adaptabilidad.

### 2.4.2. Servicios web

Como primera aproximación para lograr una definición de este concepto podemos citar al *World Wide Web Consortium* [9]:

*Un web service es un sistema de software diseñado para admitir la interacción interoperable de máquina a máquina a través de una red. Tiene una interfaz descrita en un formato procesable por máquina (específicamente WSDL). Otros sistemas interactúan con el servicio web de la manera prescrita por su descripción utilizando mensajes SOAP que normalmente se transmiten mediante HTTP con una serialización XML junto con otros estándares relacionados con la web* [10].

Inicialmente las comunicaciones se realizaban de esta manera, utilizando **SOAP**, una especificación de un protocolo de mensajería para intercambiar información estructurada en la implementación de servicios web, utiliza XML para su formato de mensaje y se basa en protocolos de capa de aplicación en su mayoría HTTP(S).

### 2.4.3. REST

Acrónimo de *REpresentational State Transfer*, REST es una abstracción de los elementos arquitectónicos dentro de un sistema hipermedia distribuido. REST ignora los detalles de la implementación del componente y la sintaxis del protocolo para enfocarse en los roles de los componentes, las restricciones sobre su interacción con otros componentes y su interpretación de elementos de datos importantes. Abarca las limitaciones fundamentales de los componentes, conectores y datos que definen la base de la arquitectura web y, por lo tanto, la esencia de su comportamiento como aplicación basada en red.

Como cualquier otro estilo arquitectónico, REST también tiene sus propias restricciones de guía que deben cumplirse si una interfaz necesita ser referida como *RESTful*. Estos principios se enumeran a continuación.

#### Principios Rectores de REST

**Cliente-servidor:** al separar las preocupaciones de la interfaz de usuario de las preocupaciones de almacenamiento de datos, mejoramos la portabilidad de la interfaz de usuario en múltiples plataformas y mejoramos la escalabilidad al simplificar los componentes del servidor.

**Sin estado (*stateless*):** cada solicitud del cliente al servidor debe contener toda la información necesaria para comprender la solicitud y no puede aprovechar ningún contexto almacenado en el servidor. Por lo tanto, el estado de la sesión se mantiene completamente en el cliente.

**Almacenable en caché (*cacheable*):** las restricciones de caché requieren que los datos dentro de una respuesta a una solicitud se etiqueten implícita o explícitamente como almacenables en caché o no almacenables en caché. Si una respuesta se puede almacenar en caché, la caché de un cliente tiene derecho a reutilizar esos datos de respuesta para solicitudes posteriores equivalentes.

**Interfaz uniforme:** al aplicar el principio de generalidad de la ingeniería de software a la interfaz del componente, se simplifica la arquitectura general del sistema y se mejora la visibilidad de las interacciones. Para obtener una interfaz uniforme, se necesitan múltiples restricciones arquitectónicas para guiar el comportamiento de los componentes. REST se define por cuatro restricciones de interfaz:

- identificación de recursos
- manipulación de recursos a través de representaciones
- mensajes auto descriptivos
- hipermedia como motor del estado de la aplicación

**Sistema en capas:** el estilo de sistema en capas permite que una arquitectura se componga de capas jerárquicas al restringir el comportamiento de los componentes de manera que cada componente no pueda "ver" más allá de la capa inmediata con la que están interactuando.

## 2.4. Conceptos importantes

**Código a pedido (*code on demand*)** (opcional): REST permite ampliar la funcionalidad del cliente descargando y ejecutando código en forma de subprogramas o scripts. Esto simplifica a los clientes al reducir la cantidad de funciones que deben implementarse previamente.

Este proyecto utiliza servicios REST que son expuestos por el servidor de Matefun y llamados desde el cliente de Matefun para lograr comunicarse sin embargo no se considera una API RESTful dado que la sesión del usuario se mantiene en el servidor con un acotado conjunto de datos mantenidos en el cliente, un ejemplo de esto es los datos de los usuarios con sesión iniciada que son mantenidos por el servidor mientras dura la sesión y obtenidos por el cliente a través de su token de acceso. Servicios REST también son expuestos por cada instancia de Moodle que se accede desde el Backend de Matefun.

### 2.4.4. HTTP

HTTP es un protocolo de capa de aplicación para transmitir documentos hipermedia, como HTML. Fue diseñado para la comunicación entre navegadores web y servidores web, pero también se puede utilizar para otros fines. HTTP sigue un modelo clásico cliente-servidor, en el que un cliente abre una conexión para realizar una solicitud y luego espera hasta recibir una respuesta. HTTP es un protocolo sin estado, lo que significa que el servidor no guarda ningún dato (estado) entre dos solicitudes, aunque se puede modelar un estado del cliente mediante cookies por ejemplo. Aunque a menudo se basa en una capa TCP/IP, se puede usar en cualquier capa de transporte confiable, es decir, un protocolo que no pierde mensajes silenciosamente como lo hace UDP.

## Capítulo 3

# Integración Matefun-Moodle

### 3.1. Por qué Moodle?

La selección de Moodle como soporte para Matefun fue hecha en base a la simplicidad que conlleva realizar la implementación de interoperabilidad además de la probabilidad que efectivamente pueda ser utilizada.

De nada nos serviría conectar Matefun con un cierto sistema LMS si éste sistema no es utilizado por una cantidad considerable de instituciones (en general) y por los centros educativos de enseñanza secundaria en particular.

Por otro lado debió investigarse que las funcionalidades que fueran a ser conectadas efectivamente tuvieran soporte en el sistema LMS elegido, es decir, se debería contar con grupos, archivos, cursos, tareas, múltiples roles, etc. y que existiera una posibilidad para ser manejadas desde sistemas externos, por ejemplo, mediante web services.

Moodle es provisto bajo los términos del licenciamiento *GPLv3* [39], esto implica que Moodle es de código abierto [27], es decir, permite que los usuarios puedan editar su código fuente cambiando las funcionalidades o agregando nuevas si así lo desean, por lo que en futuros eventuales proyectos de grado relacionados con Matefun pueda ser modificado también Moodle para lograr una mayor adaptabilidad a lo que se pueda requerir aumentando así las posibilidades.

A nivel global Moodle cuenta con una cuota de mercado del 28% [22], en norteamérica la cuota de mercado es mayor al 20% [51] contando la cantidad de centros educativos en Estados Unidos y Canadá. Por otro lado para latinoamérica la cuota de mercado trepa a los dos tercios del total [20], este volumen de uso sumado a que provee de forma nativa herramientas para que se logren gestionar los centros educativos desde sistemas externos lo convierte en una de las mejores alternativas para esta tarea.

En Uruguay diversas instituciones lo utilizan como plataforma educativa, entre las que se pueden mencionar: la Universidad de la República, la Administración Nacional de Educación Pública, la Universidad ORT, la Universidad Católica, la Intendencia de Montevideo, Presidencia de la República, la Universidad del Trabajo de Uruguay, la Universidad de Montevideo, la Administración Nacional de Telecomunicaciones, entre otras [12].

## 3.2. Arquitectura y diseño

En la presente sección se presentan los cambios a nivel de arquitectura y diseño de Matefun que han sido introducidos en este proyecto junto a las decisiones que debieron tomarse en éste sentido.

A nivel de arquitectura global de Matefun no existieron cambios significativos más allá del acople de un sistema externo que antes no existía, a partir de ahora cada centro educativo es representado por una instancia de Moodle que se conecta a través de servicios web REST y que se detalla a continuación.

### 3.2.1. Cambios en arquitectura y diseño

Se debió tomar una decisión importante a nivel de arquitectura para lograr la mejor interoperabilidad de Matefun con las instancias de Moodle (que representan centros educativos).

Se evaluaron dos opciones:

- Matefun desacoplado de Moodle, comunicación únicamente vía web-services. (sistemas independientes).
- Matefun acoplado a Moodle como plugin/add-on.

Se evaluaron las posibles ventajas y desventajas de cada una de estas opciones.

#### Ventajas de utilizar Matefun como plugin de Moodle

- Comunicación más fluida entre el plugin y el propio Moodle.

#### Desventajas de utilizar Matefun como plugin de Moodle

- Al darse el hecho que Matefun se instala y "vive" dentro de Moodle el acoplamiento es mayor lo cual constituye una practica desaconsejada [81].
- Existe la posibilidad que surjan incompatibilidades entre distintas versiones del plugin y versiones de Moodle. En cambio la API REST de Moodle es más estable en el tiempo.
- Sería necesario re-implementar funcionalidades existentes de Matefun dentro del plugin de Moodle.
- Requiere desarrollar utilizando PHP con su correspondiente curva de aprendizaje.

#### Ventajas de elegir la opción de sistemas independientes

- En términos de mantenibilidad esta opción brinda mayores libertades a la hora de realizar cambios en backend de Matefun.

## Capítulo 3. Integración Matefun-Moodle

- Al estar más desacoplada cada subsistema (frontend/backend/intérprete/-Moodle) brinda mayor independencia de los desarrollos paralelos de este proyecto: mejoramiento de gráficas, extensión del lenguaje Matefun, etc. basta con documentar las interfaces a las que se hacen referencia y sus modificaciones.
- Al tener una menor configuración/cambios por parte de Moodle que la opción alternativa, se logra mayor portabilidad.
- Menor cantidad de cambios en sistemas de terceros. (las instancias de Moodle detrás de los centros educativos).
- No requiere el entendimiento de PHP (lenguaje sobre el cual se basa Moodle) ni de la arquitectura interna del propio Moodle.
- Permite un monitoreo preciso porque permite monitorear los subsistemas por separado más fácilmente.

### Desventajas utilizar la opción de sistemas independientes

- Al no estar completamente integrado Moodle y Matefun se enlentece parte del proceso, es decir Matefun y Moodle al estar en dos servidores distintos hace que la performance y los tiempos de respuesta pasen a depender también de las latencias de la red.

**Finalmente se decidió por tener Matefun desacoplado de Moodle como sistema independiente.** La razón de esta decisión radica en que se busca maximizar los atributos de calidad positivos del sistema en su conjunto como son la mantenibilidad, portabilidad, facilidad de desarrollos futuros, posibilidad de interoperar con otros sistemas LMS además de Moodle, mayor facilidad para realizar verificación (testing). A la vez que no requiere cambios de código fuente/deploy/re-instalaciones en las instancias de Moodle de los centros educativos con los que potencialmente se trabajará.

Por un lado el backend de Matefun se encargará de consumir los web services REST provistos por las instancias de Moodle, por otro manejará la lógica de negocio específica de Matefun incluyendo las funcionalidades adicionales de este proyecto que se especifican en el documento de requerimientos.

El proyecto busca que dentro de los requerimientos que Moodle ya puede realizar sean responsabilidad exclusiva de Moodle pero a través de Matefun.

La solución final requiere (al menos) un deploy de Moodle que exponga los web-services necesarios activados para que sean consumidos por parte del backend.

En la figura 3.1 se muestra el diagrama de distribución del sistema completo que queda como resultado de acoplar la instancia de Moodle de un centro educativo al sistema original.

### 3.3. Análisis de requerimientos

El análisis y recopilación de requerimientos fue realizada en base a lo que ya tenía implementado Matefun previo a la realización de este proyecto, es decir,

### 3.3. Análisis de requerimientos

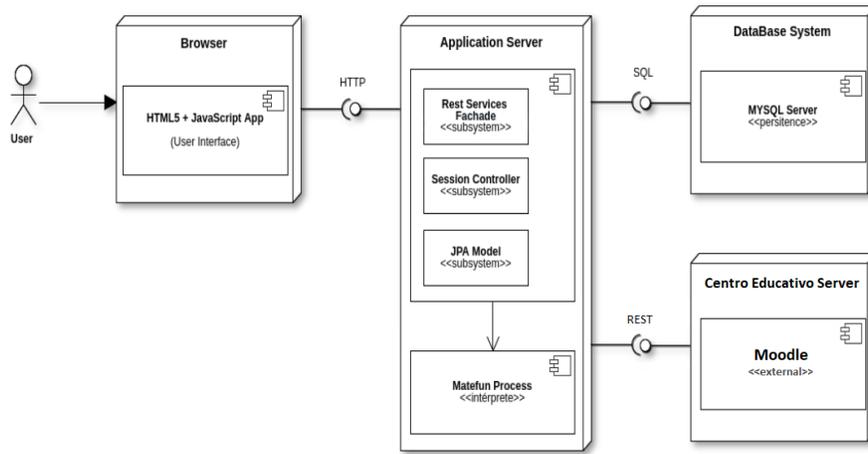


Figura 3.1: Diagrama de distribución del sistema.

las funcionalidades pre-existentes de Matefun debían ser respaldadas por el LMS (Moodle) subyacente.

Se quiso que el sistema pudiera interoperar con una o más instancias de Moodle que se corresponden a los centros educativos con los cuales Matefun trabajará.

Para esto se necesitó que existiera un usuario administrador de Matefun configurado que pudiera gestionar los parámetros necesarios para la conexión a estos centros educativos como ser URL de la instancia, nombre, etc. De esta forma se requirió la **existencia de un usuario administrador de Matefun**.

Luego, se necesitó implementar efectivamente la gestión de los centros educativos a los cuales Matefun se necesita conectar, de esta forma surgió el requerimiento de **gestión de centros educativos** (creación, actualización, borrado).

Inicialmente existió la intención que Matefun pudiera gestionar asignaturas, puntualmente crearlas y eliminarlas, si bien técnicamente la API de Moodle lo permite, este requerimiento fue descartado por ser un caso de uso central de Moodle y carece de sentido replicar esta funcionalidad.

Antes de iniciar este proyecto, Los usuarios lograban tener archivos compartidos entre ellos, esta era una de las funcionalidades centrales de Matefun. Dado esto fue necesario tener la funcionalidad de **Compartir archivos** a través de Moodle a un grupo, curso o a un usuario específico. Junto a esto, fue solicitado mejorar el manejo de edición conjunta de archivos compartidos. De esta forma surgió el requerimiento **manejo de edición simultanea de archivos compartidos**.

Matefun permitía a los docentes de la plataforma evaluar (corregir) archivos que los alumnos entregaban, de esta forma surgió el requerimiento de **Gestión de evaluaciones** (creación, edición de assignments) pero dado que la API de Moodle no lo permite fue descartado posteriormente [26] [35] [36] [40].

En cambio Moodle si permite las entregas de los alumnos a assignments previamente creados por un docente de un curso al cual el alumno pertenece,

dando surgimiento al requerimiento de **guardado y entrega de trabajos a tareas publicadas por parte de un alumno** a la vez que también permite la corrección de éstos por parte de un docente de un curso por lo que fue necesario crear el requerimiento de **corrección de tareas por parte de un docente y consulta de correcciones y calificaciones por parte de un alumno**.

La tarea más importante que realizaba (y realiza) la interfaz web de Matefun era justamente la gestión de archivos con los cuales se trabaja (además de su edición y compilación/ejecución), en particular en lo que refiere al sistema de archivos se hace referencia a creación/borrado/copiar/mover archivos y carpetas y edición de archivos compartidos y privados, de esta forma surge la necesidad de los requerimientos **eliminar/copiar archivo compartido/privado y mapeo de los archivos existentes para usuarios/docentes manteniendo navegación**.

Además de las tareas que realizaba Matefun previamente se detectó la necesidad de tener una sección donde se pudiera visualizar estadísticas generales del uso de la plataforma así como también parámetros de performance al usar la aplicación, surgiendo así los requerimientos **estado del uso de la aplicación y visualización de tiempos de acceso a funciones y estado de los servidores**

En la tabla 3.1 de la pagina 23 se presentan los requerimientos funcionales del proyecto y el resultado final de cada uno.

### 3.4. Proceso de desarrollo

El proceso de desarrollo fue encarado siguiendo un modelo iterativo, iterando sobre las funcionalidades más importantes a medida que iban terminando las anteriores. Si bien fue iterativo tampoco fue cumplido a rajatabla, es decir, el hecho de pasar a la siguiente iteración si y solo si la iteración anterior estuviera completamente finalizada no se cumplió porque dado el desconocimiento general que se tenía de Moodle y la capacitación del mismo que se iba logrando sobre la marcha hacia necesario volver hacía atrás sobre iteraciones anteriores para agregar faltantes.

El orden de las funcionalidades en la iteración fue el siguiente

1. Login, crear datos de prueba, ingreso de administrador Matefun.
2. Gestión de centros educativos.
3. Mapeo de archivos y carpetas, tareas sobre éstos. (crear/copiar/mover/-navegación/solo lectura/uso de barra de búsqueda)
4. Gestión de archivos compartidos y acceso a archivos de curso.
5. Entrega de archivos para *assignments*.
6. Listado y corrección de assignments.
7. Estado de subsistemas (módulo completo).
8. Edición conjunta de archivos compartidos y bloqueos.

### 3.4. Proceso de desarrollo

Requerimiento	Resultado al finalizar el proyecto
Generar un rol de usuario <i>administrador de Matefun</i>	<i>Completado</i>
Agregar institución educativa	<i>Completado</i>
Borrar institución educativa	<i>Completado</i>
Editar institución educativa	<i>Completado</i>
Gestión de asignaturas/cursos de un centro educativo desde Matefun.	Inicialmente fue incluido, pero luego se descartó por ser un caso de uso de Moodle y no de Matefun.
Compartir archivos a un curso	<i>Completado</i>
Compartir archivos a un grupo	<i>Completado</i>
Compartir archivos a otro usuario	<i>Completado</i>
Manejo de edición simultanea de archivos compartidos	<i>Completado</i>
Eliminar/copiar archivo compartido/privado	<i>Completado</i>
Gestión de evaluaciones (assignments, trabajos en grupo) vinculadas a una asignatura	Fue retirado porque la API de Moodle no lo permite [26] [35] [36] [40].
Guardado y entrega de trabajos a tareas publicadas por parte de un alumno	<i>Completado</i>
Consulta de correcciones de tareas/calificaciones por parte de un alumno	<i>Completado</i>
Tener acceso a archivos publicados en un curso	<i>Completado</i>
Corrección de tareas (assignments) por parte de un docente	<i>Completado</i>
Estado del uso de la aplicación	<i>Completado</i>
Visualización de tiempos de acceso a funciones y estado de los servidores	<i>Completado</i>
Mapeo de los archivos existentes para usuarios/docentes manteniendo navegación	<i>Completado</i>
Mantener la existencia del usuario invitado	<i>Completado</i>

Tabla 3.1: Tabla de requerimientos y resultados.

A medida que se iba logrando una versión estable de cada funcionalidad se procedía a testear a fondo dicha funcionalidad reportando cada defecto, a la vez que se iba investigando la API de Moodle para abordar la siguiente funcionalidad. En caso de necesidad de agregar faltantes (que se iban descubriendo a medida del conocimiento que se iba adquiriendo de la API de Moodle) se volvía atrás en las funcionalidades para implementar/verificar el faltante. Las pruebas también incluyeron tests de regresión para comprobar que no se hayan introducido defectos durante el desarrollo.

En lo que respecta a la documentación, se generaron seis recursos [62], los dos primeros (estado del arte [64] y decisiones de diseño [63]) se generaron antes de comenzar la implementación, los cuatro restantes (manual de instalación [65], manual de transferencia [66], manual de usuario [67] y el presente informe) fueron generados después de la implementación de las funcionalidades.

Con respecto al versionado del código fuente, este se encuentra en el repositorio GitLab de FING en las ramas: **matefun-moodle-integration** [60] y **master** [61].

## 3.5. Implementación

En esta sección se muestran detalles de la implementación de los requerimientos más significativos.

### 3.5.1. Existencia de usuario administrador

Existen dos formas equivalentes para configurar el usuario administrador de Matefun, la primera es ingresarlo directamente luego del primer inicio y la segunda es mediante una carga de datos de prueba que también está contemplado.

La información de este usuario administrador se guarda en la tabla **admin** del backend de Matefun que se muestra a continuación junto a la vista que muestra las opciones para configurar el usuario administrador de Matefun.



(a) Vista que se muestra al primer inicio indicando que no existe usuario administrador configurado

(b) Tabla utilizada para la configuración del usuario administrador

Figura 3.2: Requerimiento *existencia de usuario administrador*

Esta tabla contendrá solamente una tupla, el atributo **admin.username** hace referencia al nombre de usuario del administrador por su parte el atributo **admin.password** contiene la contraseña del usuario administrador luego de haberle aplicado una función de hash mediante el algoritmo SHA-1 [58] con el objetivo de no enviar la contraseña en texto plano por la red. Cuando se selecciona la opción *Cargar datos de prueba* los valores con los que se carga el usuario administrador se encuentran en:

- edu.proygrado.utils.Constants.**matefunAdminUsername**
- edu.proygrado.utils.Constants.**matefunAdminNombre**

- edu.proygrado.utils.Constants.**matefunAdminApellido**
- edu.proygrado.utils.Constants.**matefunAdminPassword**

### 3.5.2. Gestión de centros educativos

Vinculado al requerimiento anterior también se requiere lograr registrar centros educativos que serán el soporte para Matefun, para esto se crea una pantalla que será solamente accesible por el administrador de Matefun.

The screenshot shows a web interface with a blue header containing the name 'Ramiro Rodriguez'. Below the header are two panels. The left panel, titled 'Ingresar un nuevo centro educativo', contains five input fields: 'Nombre del centro educativo' (filled with 'Escuela Superior de Informatica - Buceo'), 'Url del centro educativo' (filled with 'https://esi.utu.edu.uy'), 'Nombre del usuario dueño de archivos compartidos' (filled with 'sharefilesmatefun'), 'Nombre del servicio definido para Matefun' (filled with 'matefunservice'), and 'Token del usuario dueño de archivos compartidos' (filled with a masked token). A green 'Ingresar' button is at the bottom. The right panel, titled 'Editar/Eliminar centro educativo', displays two rows of data for existing centers. Each row includes a pencil icon for editing and a trash icon for deletion. The first row shows 'Liceo 40' with URL 'http://ec2-13-59-219-116.us-east-2.compute.amazonaws.com/moodle39/' and service 'sharefilesmatefun'. The second row shows 'Liceo 38' with URL 'http://ec2-13-59-219-116.us-east-2.compute.amazonaws.com/moodle38/' and service 'serviciomatefun'.

Figura 3.3: Gestión de centros educativos.

Desde esta pantalla se pueden cargar todos los datos necesarios para poder utilizar Matefun con una instancia de Moodle como respaldo de un centro educativo. Todos los datos que son necesarios cargar y la configuración de la instancia son especificados en el documento de configuración de Moodle del actual proyecto [65].

#### Implementación de este requerimiento

Para implementar este requerimiento se ha creado una nueva entidad JPA con los datos necesarios, esto al momento de levantar el servidor Matefun crea una tabla que se utiliza a estos fines. Esta funcionalidad implementa las operaciones **CRUD**: *create*, *replace*, *update*, *delete* una vez validados la presencia de los datos y su formato del lado del frontend en el navegador, el backend persiste estos datos.

Luego del ingreso de un nuevo centro educativo, éste queda accesible para que pueda ser utilizado al momento del login de cualquier docente o alumno utilizando la misma contraseña que utiliza en Moodle.

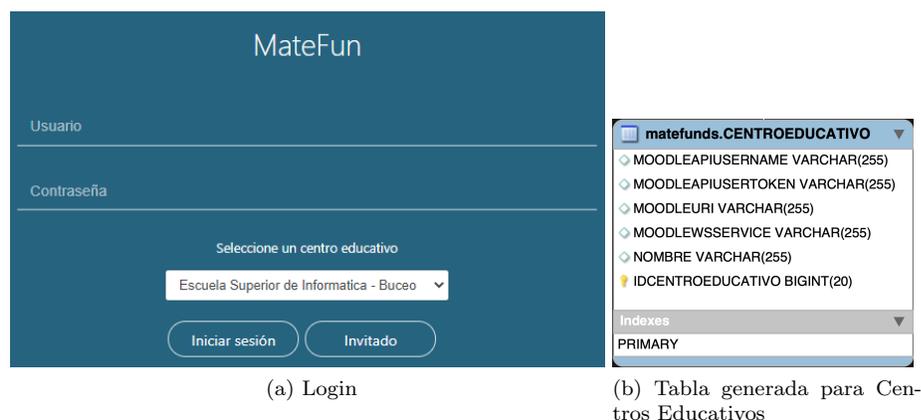


Figura 3.4: Pantalla de login y tabla para nuevos centros educativos

#### 3.5.3. Mapeo de archivos de usuario

Cada archivo que se muestra en la sección *Archivos* de Matefun debía corresponderse con un archivo en Moodle, independiente del tipo de archivos al que se haga referencia. Dentro de Matefun existe cuatro tipos de archivos:

- **Mis archivos:** Son archivos y carpetas propios del usuario, se mapean con los archivos privados de Moodle para ese usuario.
- **Archivos de curso:** Son archivos compartidos previamente por un docente (en Moodle) de un curso al cual el usuario pertenece. Esta sección corresponde a una nueva pestaña que se ha agregado en este proyecto.
- **Archivos Compartidos:** Se explica en sub-secciones posteriores.
- **Entregas:** Son entregas para una tarea (*assignment*) hecha por el alumno (corregidas o no).

Un sistema de archivos, más allá de como se represente internamente en Moodle (o en un sistema operativo) se ve como una estructura arborescente con niveles, los cuales son accedidos mediante directorios.

Dado que en el proyecto Matefun original los archivos se cargaban/guardaban desde una base de datos accedida directamente desde en backend de Matefun (muchas veces alojada en el mismo equipo físico), el acceso a la estructura del sistema de archivos era mucho más rápida porque tenía una menor carga de procesamiento en la mayoría de los casos.

El acceso al sistema de archivos del proyecto actual se hace mediante una consulta desde el navegador al backend, luego a Moodle y luego a la base de datos de Moodle, lo cual al procesamiento por parte de la instancia de Moodle que antes no existía se le debe sumar un acceso a la red que antes no existía, dado que las instancias de Moodle se encuentran en hosts diferentes al host que se encuentra el backend de Matefun.

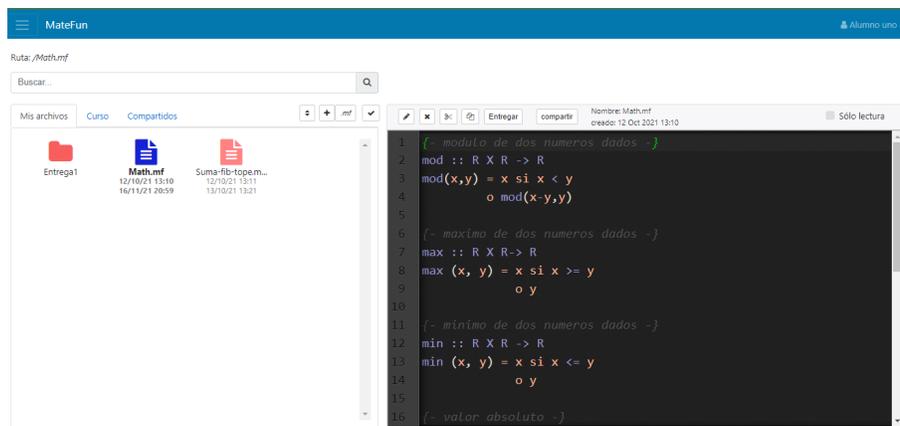
El tipo de acceso que se hacía en el proyecto original, al ser más rápido permitía que se cargara todo el árbol de archivos del usuario antes que se mostrara

### 3.5. Implementación

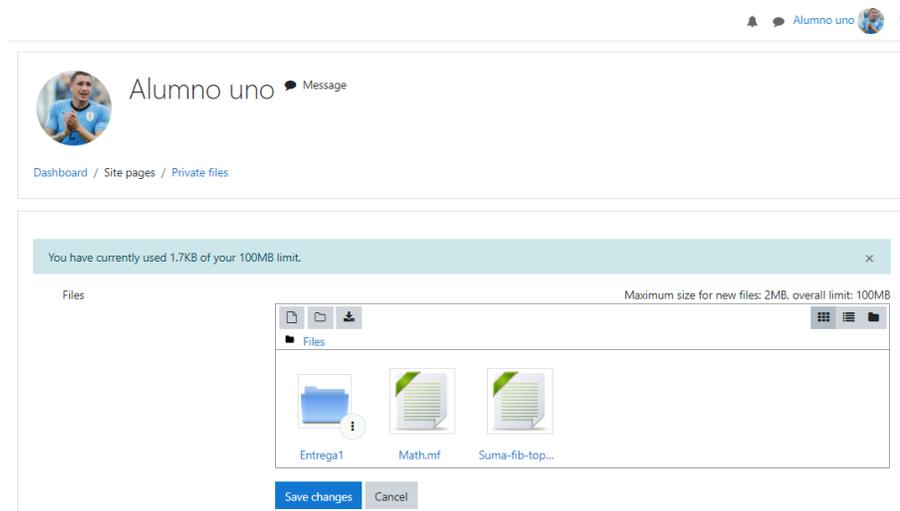
por primera vez, se notó que esto no iba a ser posible una vez que se intentaran acceder a los archivos alojados en Moodle.

Para evitar esta demora se decidió cargar el contenido de cada directorio (y los directorios *hijos*) a medida que se solicitan. Además de cargar el contenido de los archivos solamente cuando se solicita, evitando así cargar datos que no se usarían en el futuro. Ésta técnica se llama *lazy loading* [47].

Dado que los archivos compartidos se modelan como archivos privados también (como se explicará posteriormente). En la siguiente figura se muestran los archivos de un usuario particular como se ven en Moodle y sus correspondientes en Matefun.



(a) Archivos mapeados de Moodle en Matefun. El ícono azul indica que el archivo se encuentra seleccionado, debajo de cada archivo se muestran las fechas de creación y última edición respectivamente.



(b) Archivos originales en Moodle

Figura 3.5: Mapeo de archivos y carpetas en Matefun (arriba) y Moodle (abajo).

### 3.5.4. Compartir archivos

Uno de los requerimientos que se desprende de la implementación de Matefun original fue lograr compartir archivos (como ya existía originalmente) pero con la diferencia que deberían estar respaldados en Moodle.

Moodle por defecto (sin el uso de plugins) no permite compartir archivos entre estudiantes sin complicar excesivamente la configuración previa para poder usarlo, al menos hasta la última versión al momento de escribir el documento, por lo que fue necesario encontrar una alternativa para lograr implementar el requerimiento dada esta limitante.

Una posibilidad que podría existir es utilizar plugin(s) para lograr este cometido, pero esto obligaría a tener siempre el plugin correspondiente instalado (y en su versión correcta) para poder utilizar la funcionalidad de compartir archivos, lo cual reduce el público objetivo.

#### Implementación

La solución que se decidió implementar es tener un usuario específico (*sharedfilesMatefun* [65]) para manejar archivos compartidos, es decir los archivos compartidos entre los usuarios de un centro educativo son los archivos privados de un usuario particular creado para este cometido al cual todos tienen acceso.

Cada usuario de un centro educativo tiene un *token* definido para utilizar todos los *web-services* de la instancia de Moodle subyacente, además de tener disponible el *token* del usuario que es dueño de los archivos compartidos logrando así tener acceso a los archivos que le corresponden.

En Moodle los archivos privados del usuario *sharedfilesMatefun* se persisten con un formato en su nombre de archivo que permite conocer a que usuario le es correspondiente acceder.

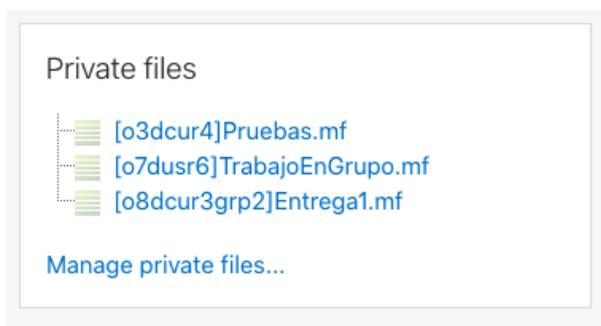


Figura 3.6: Formato en el nombre de archivos compartidos visualizados en Moodle.

Al momento de compartir un archivo cuyo *filename* es **nombre.mf**, éste se guarda en el espacio de archivos privados del usuario *sharedfilesMatefun* con alguno de los tres formatos siguientes:

1. [oX $d$ curY]nombre.mf
2. [oX $d$ usrY]nombre.mf

## 3. [oXdcuYgrpZ]nombre.mf

El primer caso corresponde a un usuario que comparte un archivo con todo un curso al cual este usuario pertenece, **X** corresponde con el identificador de usuario en Moodle que comparte e **Y** corresponde al identificador de curso de destino en Moodle, es decir todos los alumnos/docentes anotados a ese curso podrán acceder al archivo.

El segundo caso corresponde a un archivo que lo comparte un usuario cuyo identificador en Moodle es **X** hacia un usuario cuyo identificador es **Y**, solamente estos dos usuarios tienen acceso a este archivo.

El tercer caso corresponde a un archivo que lo comparte un usuario cuyo identificador en Moodle es **X** hacia todos los usuarios del grupo con identificador **Z** que está en el curso con identificador **Y**.

En todos los casos los usuarios destinatarios de los archivos son usuarios que comparten algún curso con el usuario que comparte el archivo, es decir, no se puede compartir archivos con usuarios que realizan cursos totalmente distintos a los cursos que pertenece el usuario que comparte.

Cuando un usuario desea consultar por sus archivos compartidos desde la interfaz web de Matefun, el backend de Matefun consulta por todos los archivos compartidos existentes y se filtran los archivos que al usuario no le corresponde tener acceso, mostrando solamente:

- Los archivos que el usuario ha compartido a otros usuarios.
- los archivos que se han compartido a un curso al cual el usuario está inscripto. (caso 1)
- Los archivos que le han compartido directamente (caso 2)
- los archivos que se han compartido a un grupo (de un curso) al cual el usuario pertenece. (caso 3)

## 3.5.5. Manejo de archivos compartidos y bloqueos

Se logró implementar una mejora al manejo de archivos compartidos, pudiendo ahora un archivo ser bloqueado mientras se edita o ejecuta por un usuario de modo que los demás usuarios que tengan ese archivo abierto solamente puedan ejecutar el código Matefun, pero no puedan editar el archivo. Si un usuario intenta editar un archivo bloqueado Matefun le obliga a guardar los cambios en otra ubicación dentro de sus archivos privados.

Al guardar cambios en un archivo compartido la interfaz web ahora ofrece dos opciones:

- Guardar y bloquear
- Solamente guardar

Si el usuario decide solamente guardar simplemente se modifica su contenido en Moodle dejando la posibilidad que otro usuario guarde cambios propios a ese archivo posteriormente. En cambio si el usuario decide guardar y bloquear

### Capítulo 3. Integración Matefun-Moodle

tendrá la posibilidad que otros usuarios no puedan editar ese archivo mientras esta bloqueado. El archivo permanecerá bloqueado por un tiempo configurable de inactividad que será renovado cuando se guarden cambios.

Luego de que el usuario que tiene bloqueado el archivo termine con su edición tendrá tres opciones:

- Solo desbloquear
- Guardar y mantener bloqueado
- Guardar y liberar

Si **solo se desbloquea el archivo**, éste quedara liberado para la edición para otros usuarios pero no se guardarán cambios en Moodle.

Se define un timer de inactividad en el backend (*timerMaxInactividadMs*) para que si se desea **guardar y mantener bloqueado** se guardarán los cambios en Moodle y se renueva este *timer*, dicho valor se utiliza para correr un hilo en el servidor que desbloquea los archivos bloqueados inactivos (que no reciben actualizaciones) porque se asume que no se han desbloqueado por alguna razón ajena a la voluntad del editor del archivo.

Al **Guardar y liberar** como su nombre lo explica se persisten los cambios en Moodle para luego dejar el archivo abierto para que otros usuarios lo puedan editar.

#### Implementación

Cada vez que se abre un archivo compartido (bloqueado o no) se utiliza un intervalo de tiempo (*requestTimerMs*) en el cual se hace una consulta al servidor de Matefun si se han hecho cambios en el archivo compartido por otro usuario. Simplemente el navegador envía la ultima fecha de modificación del archivo además de un booleano que indica si se encuentra bloqueado o no y en caso que hayan sido modificados estos valores se retorna al Backend la información con los nuevos cambios que se hayan efectuado (i.e. en nuevo archivo y/o la nueva información de bloqueo).

- El valor de *requestTimerMs* es configurable en el frontend en:  
*Frontend Angular 4/src/app/shared/config.ts*
- El valor de *timerMaxInactividadMs* es configurable en el backend en:  
*Servidor JEE/src/main/java/edu/proygrado/utis/Constants.java*

Si bien a cada vencimiento de *requestTimerMs* enviar una petición al servidor por cada cliente resulta ser una solución no escalable, en contrapartida para mitigar esto se mantiene la ultima versión del archivo compartido *cacheada* en el backend, es decir no se consulta a Moodle cada vez sino que se consulta a un valor guardado en la memoria del servidor reduciendo así los tiempos de acceso que han resultado ser del orden de decenas de milisegundos solamente, esta solución puede ser mejorada en términos de performance utilizando web-sockets [33].

### 3.5.6. Corrección de evaluaciones

La mayor limitante respecto a esta funcionalidad surgió a la hora de listar las evaluaciones y archivos entregados para estas evaluaciones, esto se debe al tipo de contrato que tiene Moodle en su API para esta funcionalidad.

El proyecto original de Matefun fue desarrollado con una pantalla que listaba todas las evaluaciones existentes en cursos donde el usuario logueado tiene rol de docente para poder corregir evaluaciones.

La intención es reducir al máximo los cambios en esta pantalla, es decir que la interfaz de usuario tenga el mismo flujo de ser posible. Lamentablemente Moodle no cuenta con una forma ágil para consultar estos datos.

Cuando se quiere consultar todas las entregas para una evaluación (*assignment*) de un curso, se debe ejecutar una consulta a Moodle por cada alumno que pertenece a dicho curso y si a la vez existen varias evaluaciones en dicho curso la cantidad de consultas se multiplica aun más.

Las evaluaciones pueden ser individuales o grupales, cada evaluación tiene su identificador. Cuando se consulta por entregas de una evaluación grupal basta con consultar por un solo alumno de ese grupo, se muestra el siguiente ejemplo. Si un usuario es docente en un curso con 15 alumnos que tiene tres tareas

1. Tarea A (grupal, con cinco grupos de tres alumnos)
2. Tarea B (individual)
3. Tarea C (grupal, con tres grupos de cinco alumnos)

Lo que resulta en que solo para este curso es necesario realizar  $5 + 15 + 3 = 23$  consultas a Moodle.

Por causa de esta limitante se decidió realizar las consultas en paralelo desde el backend de Matefun. Dado un curso sobre el cual se consulta, se paraleliza las consultas sobre alumnos y evaluaciones.

A continuación se presenta un pseudocódigo de este proceso.

---

**Algorithm 1:** Consulta de entregas para evaluaciones para el curso  $i$

---

```

alumnos ← obtenerAlumnosDeCursoDesdeMoodle( $i$ );
evaluaciones ← obtenerEvaluacionesDeCursoDesdeMoodle( $i$ );
tuplas = [];
for alumno en alumnos do
  | for evaluacion en evaluaciones do
  | | tuplas.agregar(<alumno,evaluacion>);
  | end
end
for  $t$  en tuplas (en paralelo) do
  | archivos = traerArchivosEvaluacionesDeMoodle( $i$ ,  $t$ .alumno,
  | |  $t$ .evaluación);
  | resultado.agregar(archivos);
end
devolver resultado;

```

---

## Capítulo 3. Integración Matefun-Moodle

Como el hilo que hace la llamada a Moodle se quedaría bloqueado esperando por la respuesta, se utilizan varios hilos de ejecución para poder hacer las llamadas concurrentemente. Para esto se utiliza parallel streams [52].

### 3.5.7. Borrado de archivos

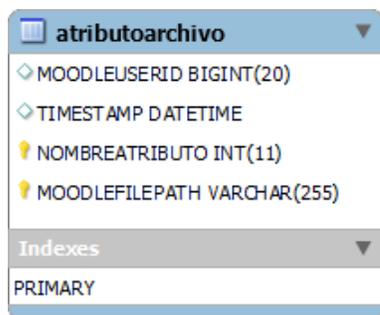
Moodle no permite realizar eliminaciones físicas de archivos mediante web services utilizando la API que provee.

Dada esta limitación fue necesario implementar un borrado lógico utilizando la base de datos propia de Matefun.

Existe una tabla donde se guardan los atributos de los archivos de todos los centros educativos, estos atributos pueden tener tres valores:

1. Solo lectura
2. **Eliminado**
3. Bloqueado

Cuando Matefun lista los archivos privados de un usuario, luego de consultarlos a la instancia de Moodle del centro educativo correspondiente consulta la lista de atributos definidos sobre ese archivo en la tabla creada, si tiene el atributo *eliminado* el archivo simplemente se filtra para que no sea mostrado.



atributoarchivo	
MOODLEUSERID	BIGINT(20)
TIMESTAMP	DATETIME
NOMBREATRIBUTO	INT(11)
MOODLEFILEPATH	VARCHAR(255)
Indexes	
PRIMARY	

Figura 3.7: Tabla para manejar los atributos de los archivos tales como eliminado/readonly/bloqueado.

Arriba se muestra la tabla que se utiliza para implementar la eliminación lógica de archivos, esta misma tabla es la que se utiliza para manejar el bloqueo de archivos compartidos como se describe en la subsección *Manejo de archivos compartidos y bloqueos*

### 3.5.8. Estado de subsistemas

Uno de los requerimientos planteados fue implementar una sección de *estado de subsistemas* disponible solamente para el administrador de Matefun que permitiera conocer fallas y hacer un seguimiento del uso que se le da al sistema además de datos relacionados a la performance de Matefun.

Consta de tres subsecciones:

### 3.5. Implementación

1. General
2. Tiempos de acceso
3. Estado Servidores

#### Registro para la pestaña *General*

La primera sección se desarrolló con la intención de realizar un seguimiento del uso de la aplicación, cada vez que se realizan acciones a través de Matefun se guarda en un registro correspondiente para poder desplegar en esta sub-sección.

Con *acciones* se hace referencia a:

- Acceso a archivos
- Logins en tiempo real.
- Conteo de histórico de logins.
- Tareas entregadas por curso.
- Archivos compartidos por curso.
- Datos generales del centro educativo, etc.

Para esto se crearon cuatro nuevas entidades JPA (que corresponden a cuatro tablas en la base de datos de Matefun), éstas son:

- **REGISTROACCESOARCHIVO**, utilizada cuando se obtiene el contenido de un archivo (sea éste compartido, privado o de curso) a través de Matefun, registrando la información de tamaño, usuario, fecha y curso si corresponde.
- **REGISTROCOMPARTIRARCHIVO**, utilizada cada vez que un usuario comparte un archivo se genera un registro en esta tabla, guarda como información la fecha del evento, tipo destinatario (curso entero/individual/subgrupo dentro de un curso), centro educativo, etc.
- **REGISTROLOGIN**, utilizada cada vez que un usuario inicia sesión en Matefun guardando los datos del usuario así como un registro de fecha del evento.
- **REGISTROTAREA**, utilizada cada vez que se hace una corrección o entrega de una tarea (*assignment*), guardando toda la información relativa a la tarea, el archivo que se entrega, curso y la información del usuario.

Cada vez que se guarda un registro en alguna de estas tablas se realiza de manera asíncrona en un *thread* distinto al hilo principal que realiza la acción, esto se hace con el objetivo de no demorar la respuesta ya que la información que se guarda (y cálculos intermedios) no se utilizan para la respuesta.

## Capítulo 3. Integración Matefun-Moodle

registroaccesoarchivo	registrocompartirarchivo	registrologin	registrotarea
CANTIDADVECES BIGINT(20)	TAMANIO BIGINT(20)	ID BIGINT(20)	ID BIGINT(20)
TAMANIO BIGINT(20)	FECHA DATE	FECHA DATE	ASSIGNMENTID BIGINT(20)
FECHA DATE	MOODLEUSERID BIGINT(20)	IDCENTROEDUCATIVO BIGINT(20)	CURSOID BIGINT(20)
MOODLEUSERID BIGINT(20)	TIPO INT(11)	MOODLEUSERID BIGINT(20)	FECHA DATE
TIPO INT(11)	RUTA VARCHAR(255)	MOODLEUSERNAME VARCHAR(255)	NOMBRETAREA VARCHAR(255)
RUTA VARCHAR(255)	MOODLEUSERFULLNAME VARCHAR(255)		TIPOACCION INT(11)
IDCENTROEDUCATIVO BIGINT(20)	ENTITYID1 BIGINT(20)		TIPOAREA INT(11)
CURSOID BIGINT(20)	IDCENTROEDUCATIVO BIGINT(20)		IDCENTROEDUCATIVO BIGINT(20)
	CURSOID BIGINT(20)		
	ENTITYID2 BIGINT(20)		
Indexes	Indexes	Indexes	Indexes
PRIMARY	PRIMARY	PRIMARY	PRIMARY
			FK_REGISTROTAREA_IDCENTROEDUCATIVO

Figura 3.8: Tablas utilizadas en la pestaña general.

### Consulta para la pestaña *General*

Luego para realizar la consulta de estos datos simplemente se hace un recorrido por las tablas para obtener esta información. El frontend realiza una consulta *GET /estadosistema* al servidor de Matefun retornando esta información una única vez, teniendo la opción de refrescar los datos.

### Registro para la pestaña *tiempos de acceso*

Análogamente para esta sección se crea una entidad nueva entidad JPA, y con esto una nueva tabla llamada **TIEMPOACCESO** que guarda la información diaria de cada una de las interacciones con los servidores que se realizan, estas consultas pueden ser de dos tipos:

- Llamadas al servidor de *Matefun*
- Llamadas a alguna instancia de *Moodle*

Cuando se menciona más arriba *información diaria* se hace referencia a que por cada día calendario se guarda la información de una llamada dada una única vez, actualizando ese registro cuando ocurran llamadas sucesivas. Esto se puede explicar de una manera más técnica como que la fecha, el nombre del endpoint y el identificador del centro educativo forman parte de la clave primaria de la tabla **TIEMPOACCESO**.

Dada que cada tupla de esta tabla registra uno o más llamadas al servidor (Matefun o Moodle) se guarda un valor máximo, mínimo y promedio diarios.

Se considera una consulta de tipo Moodle a un acceso a una instancia de un centro educativo ejecutando una operacion sobre el LMS, todas las interacciones con Moodle se hacen apuntando a la misma URL pero variando un parametro que corresponde al nombre de la función, ejemplos:

- **core\_group\_get\_course\_groups**: (para obtener grupos de un curso)
- **core\_user\_add\_user\_private\_files**: (para obtener archivos privados de usuario)

### 3.5. Implementación

- **mod\_assign\_get\_assignments**: (para obtener las tareas existentes en un curso)

En cambio las consultas de tipo Matefun son accesos a la API que provee el backend de Matefun, ejemplos:

- **GET /login/getMatefunAdmin**: para obtener el nombre de usuario del administrador de Matefun
- **GET /centroeducativo/getAllSchools**: para obtener la información de todos los centros educativos existentes.
- **PUT /archivo/atributo**: para crear o modificar un atributo a un archivo existente

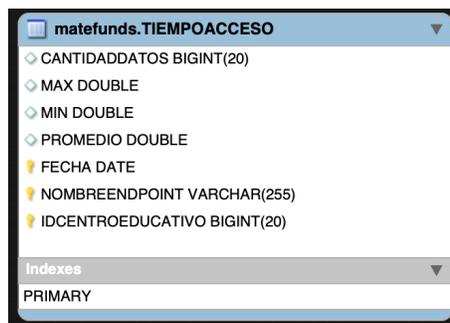


Figura 3.9: Tabla donde se guardan los tiempos de acceso para consultas Moodle/Matefun.

#### Consulta para la pestaña *tiempos de acceso*

En lo que a consultar estos datos se refiere para la segunda pestaña se ha desarrollado una interfaz (similar al caso general) donde ante cada cambio de controles en la pantalla se actualiza la vista.

#### Pestaña *estado Servidores*

En esta pestaña se requiere mostrar el estado de los servidores en tiempo real, información como parámetros de carga, uso de la memoria, uso de cpu, etc.

Estos datos se obtienen desde el backend de Matefun consultando mediante comandos de consola (*bash*) directamente al sistema operativo donde está desplegado Matefun.

Algunos de estos comandos son:

- **top** [17]: ej. para obtener carga, porcentaje uso de CPU, memoria, etc.
- **lsof** [16]: para obtener el identificador de proceso que corre el servidor
- **curl** [15]: ej. para obtener la IP del servidor

### Capítulo 3. Integración Matefun-Moodle

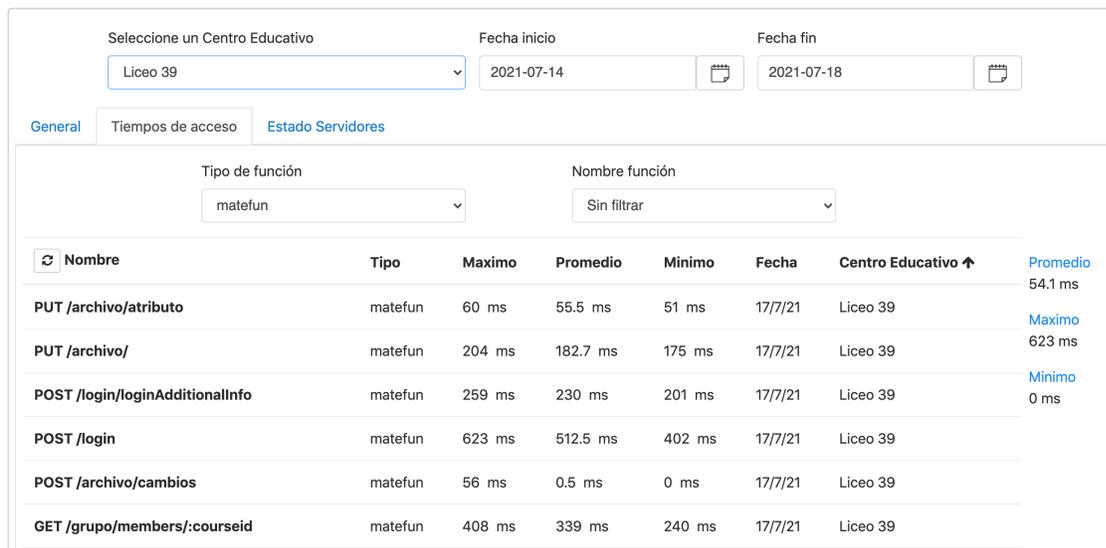


Figura 3.10: Vista de la pestaña tiempos de acceso.

Se realiza una consulta cada 1.5 segundos para obtener estos datos al servidor Matefun, solamente se recalculan los datos variables, carga, %memoria, %cpu, etc, datos que son fijos como direcciones ip, puertos no son recalculados.

En la vista se dispone de un control que permite cambiar la frecuencia de consulta de los datos, así como también cambiar el intervalo de consulta.

Como el entorno de desarrollo fue Apple macOS y el entorno de testing fue Ubuntu (como se vera más adelante) de forma nativa utilizan Bash y Windows no lo utiliza es por eso que esta funcionalidad esta disponible solamente en Linux y macOS.

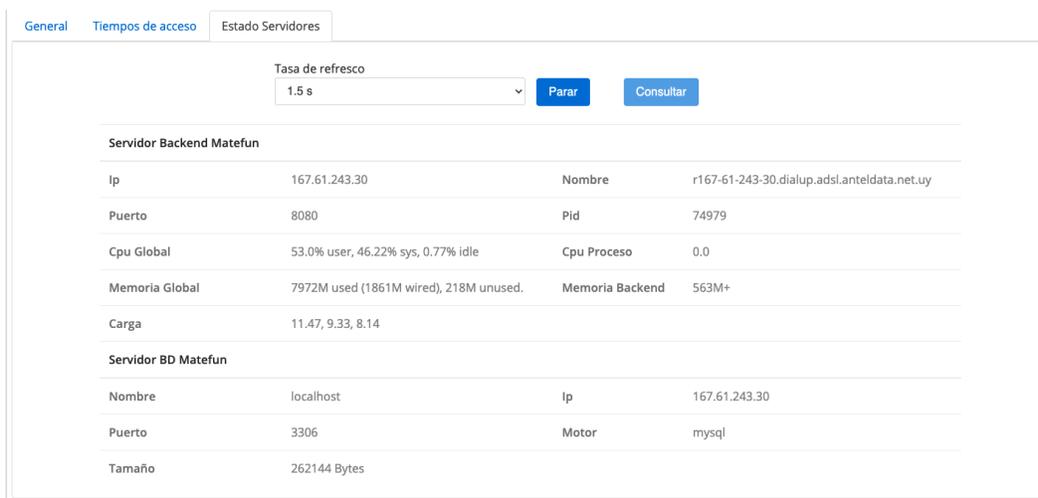


Figura 3.11: Vista de la pestaña estado de servidores.

## Capítulo 4

# Experimentación

Como se ha mencionado anteriormente, este proyecto fue realizado utilizando un modelo de proceso iterativo y también de esta manera fue encarada la experimentación. Las etapas fueron creadas siguiendo un orden de ejecución de los casos de uso más importantes:

1. Login, crear datos de prueba, ingreso de administrador Matefun.
2. Gestión de centros educativos.
3. Mapeo de archivos y carpetas, tareas sobre éstos. (crear/copiar/mover/-navegación/solo lectura/uso de barra de búsqueda)
4. Gestión de archivos compartidos y acceso a archivos de curso.
5. Entrega de archivos para *assignments*.
6. Listado y corrección de *assignments*.
7. Estado de subsistemas (módulo completo)
8. Edición conjunta de archivos compartidos y bloqueos.

A medida que se iba logrando terminar cada punto de los descriptos arriba se procedía a hacer testing exploratorio [23] sobre cada punto logrando completar cada flujo mientras se documentaron los casos de prueba, es decir, a medida que se iban completando las pruebas de los casos de uso más utilizados, éstas se iban documentando al mismo tiempo y se presentan en la sección 4.3.

Por simplicidad y para mantener una cantidad acotada de casos de prueba presentados en este documento, no se incluyeron casos de prueba para las últimas dos etapas de la lista, ésta es la razón por la que las últimas dos iteraciones de la lista no presentan referencias a la sección 4.3.

Por otro lado, al tener este proyecto como objetivo lograr una extensión a un producto pre-existente se hizo aun más importante realizar pruebas de regresión [73] mientras que este proyecto iba avanzando para verificar que a medida que se fueran creando nuevas funcionalidades no se introduzcan nuevas fallas en partes del sistema donde antes funcionaba correctamente. Estas pruebas

## Capítulo 4. Experimentación

fueron realizadas luego de cada finalización de los puntos que así lo requerían (ej. lograr ejecutar los archivos *.mf* mapeados desde Moodle para comprobar que se siguen pudiendo ejecutar). Las pruebas de regresión fueron documentadas en la sección 4.4

### 4.1. Ambiente de pruebas

En lo que a ambiente de pruebas se refiere, se evaluaron dos posibilidades de soluciones de PaaS (platform as a service, básicamente servicio de cloud hosting) a la hora de alojar la instancia de Moodle:

- Heroku
- Amazon Elastic Compute Cloud (o EC2, parte de Amazon Web Services)

Si bien la primera tiene interacción mucho más fluida con Git (la herramienta de control de versiones del proyecto) y el servicio de hosting GitHub, la configuración llega a ser bastante más engorrosa a lo que configuración de proyecto se refiere.

EC2 simplemente ofrece máquinas virtuales con sistemas operativos pre-instalados donde incluso la base de datos de Moodle podría encontrarse en el mismo filesystem logrando mejores tiempos de acceso en el caso promedio (en comparación que necesita add-ons y los accesos son externos). Heroku sin embargo su plan gratuito ofrece un espacio disponible mucho más reducido en comparación a EC2, teniendo en cuenta esto, se decide por alojar las instancias de Moodle y Matefun en EC2.

### 4.2. Deploy en AWS

Se realizó un deploy en la plataforma de cloud computing *amazon web services* con el objetivo de probar la interacción de Matefun-Moodle con otros entornos/sistemas operativos distintos a los utilizados en el momento del desarrollo. Los detalles de la plataforma se muestran a continuación.

- **Tipo de instancia:** t2.medium (en esta plataforma, un *tipo de instancia* determina una configuración de CPU, memoria, almacenamiento, red, sistema operativo, etc.)
- **SO:** Ubuntu Server 18.04 LTS (Bionic Beaver)
- **CPU:** 2 CPU's virtuales AMD x86 (64bit) a 2.3Ghz
- **Memoria:** 4 GB
- **HD:** 8GB SSD, IOPS: 100

Esta instancia tiene instalado:

- Oracle Java Development kit 8 update 291
- wildfly version 10.1.0.Final

### 4.3. Casos de prueba

- mysql version 8.0.25
- UnZip 6.00
- libz3 (necesario para correr el compilador de Matefun)
- libgomp1 (necesario para ejecutar el compilador de Matefun)

Todas las instancias de Moodle que corresponden a los centros educativos se encuentran (por motivos de simplicidad y costo) bajo el mismo servidor virtual en AWS EC2 respondiendo desde la misma dirección IP, a continuación se detallan las especificaciones de este servidor.

- **SO:** Ubuntu Server 16.04 LTS (Xenial Xerus)
- **Tipo de instancia:** t2.small
- **CPU:** 1 CPU virtual AMD x86 (64bit) a 2.5Ghz
- **Memoria:** 2 GB
- **HD:** 8GB SSD, IOPS: 100

Esta instancia tiene instalado:

- PHP 7.4.10
- Moodle 3.8
- Moodle 3.9
- mysql Ver 14.14

La razón por la que los tipos de instancia donde se despliegan Matefun y Moodle son distintas entre si (*t2.medium* y *t2.small*) radica en los precios que Amazon cobra por cada tipo de instancia sumado a que Matefun (frontend + backend) no lograba iniciar en una instancia *t2.small* por falta de recursos, por esto se debió recurrir a una instancia de tipo *t2.medium* para el despliegue de Matefun.

Se incluye un nuevo script en el repositorio del proyecto Matefun que permite hacer este deploy automáticamente, realiza las tareas de compilar el frontend y backend, crear el recurso WAR y enviarlo a la instancia en AWS, resetear la base de datos remota y finalmente reiniciar el servidor Wildfly [69] [68].

En la siguiente sección se presentan algunos casos de prueba realizados para los ítems 1 al 6 de la lista.

Luego en el anexo se puede encontrar un resumen de todos de los casos de prueba realizados (ítems 1 al 6) siguiendo como modelo los lineamientos definidos en [31]

### 4.3. Casos de prueba

A continuación se presenta un resumen de casos de prueba realizados de funcionalidades nuevas (no incluye pruebas de regresión).

## Capítulo 4. Experimentación

### 4.3.1. Login

Titulo	<b>Inicio de sesión de alumno (o docente)</b>
Resumen	Se busca que se pueda iniciar sesión de un usuario configurado en Moodle
Dependencias	En la instancia de Moodle del centro educativo utilizado debe estar correctamente configurado el alumno o docente con el cual se desea iniciar sesión según [65]
Pre-condiciones	Existe al menos un centro educativo cargado (tabla 4.2)
Pasos	1: Ir a la pagina de login 2: Introducir nombre de usuario y contraseña 3: Seleccionar centro educativo (si hay más de uno cargado) 4: Click en 'Iniciar sesión'
Datos	no tiene
Resultado esperado	Se inicia sesión redirigiendo a la pagina de edición de archivos
Post-Condiciones	Se cargan las tablas de 'registrologin' y 'tiemposacceso' con los datos correctos
Resultado obtenido	Resultado esperado

Tabla 4.1: Test case [Login>Login usuario correcto

Casos de prueba adicionales de Login en el anexo.

### 4.3.2. Gestión de centros educativos

Titulo	<b>Ingresar centro educativo</b>
Resumen	Se busca ingresar un centro educativo
Dependencias	1: Debe haber sido ingresado un administrador de Matefun mediante alguno de los casos de prueba de las tablas 6.1 ó 6.2 en los anexos. 2: El centro educativo utilizado (Moodle) debe estar correctamente configurado según [65]
Pre-condiciones	No tiene
Pasos	1: Iniciar sesión con administrador de Matefun 2: Ir a la pagina de gestión de centros educativos 3: En la sección de la izquierda ingresar los cinco campos según [65] 4: Click en <i>Ingresar</i>
Datos	No tiene
Resultado esperado	Se agrega un centro educativo en la lista de la derecha con los datos ingresados
Post-Condiciones	Se carga la tabla 'tiempoacceso' Se agrega el centro educativo en la tabla 'centroeducativo'
Resultado obtenido	Resultado esperado

Tabla 4.2: Test case [Gestión de centros educativos]Ingreso de centro educativo

Casos de prueba adicionales de gestión de centros educativos en el anexo.

## 4.3. Casos de prueba

### 4.3.3. Mapeo de archivos y carpetas

Título	<b>Carga directorio raíz de archivos privados</b>
Resumen	Se busca que Matefun logre cargar los archivos y directorios privados contenidos en el directorio raíz
Dependencias	1: Se logra ejecutar correctamente el caso de prueba de inicio de sesión 2: Debe existir un centro educativo ingresado mediante el caso de prueba de la tabla 4.2
Pre-condiciones	1: En la instancia de Moodle del centro educativo utilizado debe estar correctamente configurado el alumno o docente con el cual se desea iniciar sesión según [65] 2: El alumno debe pertenecer al centro educativo al cual inicia sesión
Pasos	1: Iniciar sesión con un alumno en Matefun en su centro educativo 2: Ir a la sección de archivos
Datos	1: El alumno debe tener archivos privados en el directorio raíz en la instancia de Moodle 2: Ninguno de los archivos fue eliminado o movido (desde o hacia el directorio raíz) desde Matefun anteriormente
Resultado esperado	Se listan en Matefun exactamente los mismos archivos y carpetas que se muestran en la sección de archivos privados de Moodle
Post-Condiciones	Se carga la tabla 'tiempoacceso'
Resultado obtenido	Resultado esperado

Tabla 4.3: Test case [Mapeo de archivos y carpetas]Carga de archivos privados

Casos de prueba adicionales de mapeo de archivos y carpetas en el anexo.

## Capítulo 4. Experimentación

### 4.3.4. Gestión de archivos compartidos

Titulo	<b>Compartir un archivo 'arch' a todos los integrantes de un curso</b>
Resumen	Se busca que un usuario 'u' de Matefun logre compartir un archivo 'arch' a todos los usuarios anotados en un curso 'c'
Dependencias	Se logra ejecutar correctamente el caso de prueba de inicio de sesión
Pre-condiciones	No tiene
Datos	<ol style="list-style-type: none"> <li>1: El usuario 'u' debe pertenecer a la instancia de Moodle del centro educativo al cual accede</li> <li>2: El usuario 'u' debe estar inscripto en un curso 'c'</li> <li>3: Debe existir al menos un archivo 'arch' (de contenido no vacío) que se pretenda compartir en la vista de 'Mis archivos' en Matefun</li> <li>4: Para que tenga sentido este test case debe haber al menos un usuario distinto de 'u' inscripto al curso 'c'</li> </ol>
Pasos	<ol style="list-style-type: none"> <li>1: Realizar login como alumno/docente 'u'</li> <li>2: Ir a la sección de archivos</li> <li>3: Ir a la tab 'Mis archivos'</li> <li>4: Clic en el archivo 'arch' que se desea compartir</li> <li>5: Click en el botón 'Compartir' (derecha)</li> <li>6: Seleccionar un curso 'c'</li> <li>7: Seleccionar la opción 'Compartir a todo el curso <i>nombre de curso</i>'</li> <li>8: Click en '<i>Compartir</i>'</li> </ol>
Resultado esperado	<ol style="list-style-type: none"> <li>1: Se muestra un mensaje de éxito del tipo '<i>Se ha compartido el archivo</i>'</li> <li>2: Se logra ver una copia del archivo compartido en la tab <i>compartidos</i> de la sección de Archivos</li> </ol>
Post-Condiciones	<ol style="list-style-type: none"> <li>1: Se carga la tabla 'tiempoacceso'</li> <li>2: Se carga la tabla 'registroaccesoarchivo'</li> <li>3: Se carga la tabla 'registrocompartirarchivo'</li> <li>4: Existe un archivo con nombre del tipo <math>[oXdcuY]nombreamchivo.mf</math> en el área de archivos privados del usuario dueño de archivos compartidos en Moodle (<i>sharedfilesMatefun</i>)</li> </ol>
Resultado obtenido	Resultado esperado

Tabla 4.4: Test case [Gestión de archivos compartidos]Compartir un archivo a un curso

Casos de prueba adicionales de gestión de archivos compartidos en el anexo.

### 4.3. Casos de prueba

#### 4.3.5. Entrega de archivos para assignments

Titulo	<b>Entrega de archivos a un <i>assignment</i></b>
Resumen	Se busca que el alumno 'a' logre hacer una entrega de un archivo 'arch' para una tarea publicada 't' en el curso 'c'
Dependencias	Se logra ejecutar correctamente el caso de uso de inicio de sesión
Pre-condiciones	1: Existe una tarea ( <i>assignment</i> ) publicada 't' en el curso 'c' desde la instancia Moodle del centro educativo en el cual se accede. 2: La tarea 't' se encuentra habilitada para entregas al momento de ejecutar el caso de uso
Datos	1: El alumno 'a' debe pertenecer a la instancia de Moodle del centro educativo al cual accede 2: El alumno 'a' debe estar inscripto en el curso 'c'
Pasos	1: Iniciar sesión con el alumno 'a' 2: Ir a la sección 'Archivos' 3: Seleccionar el archivo 'arch' en la pestaña 'Mis archivos' 4: Click en el botón 'Entregar' 5: Seleccionar el curso 'c' 6: Seleccionar la actividad ( <i>assignment</i> ) 't' 7: Click en el botón verde 'Entregar'
Resultado esperado	Se muestra un mensaje del tipo ' <i>Archivo entregado con éxito</i> '
Post-Condicion	1: Se carga la tabla 'tiempoacceso' 2: Se carga la tabla 'registrotarea'
Resultado obtenido	Resultado esperado

Tabla 4.5: Test case [Entrega de archivos para *assignments*]Entrega tarea

Casos de prueba adicionales de entrega de archivos para assignments en el anexo.

## Capítulo 4. Experimentación

### 4.3.6. Listado y corrección de assignments

Título	<b>Corrección de <i>assignment</i> a un alumno</b>
Resumen	Se busca que el docente 'd' logre listar y corregir las tareas entregadas 't' del alumno 'a' del curso 'c'
Dependencias	1: Se logra ejecutar correctamente el caso de uso de entrega de archivos
Pre-condiciones	1: Existe una tarea (assignment) publicada 't' en el curso 'c' del centro educativo en el cual se accede. 2: La tarea 't' se encuentra habilitada para corrección al momento de ejecutar este caso de prueba.
Datos	1: El alumno 'a' debe pertenecer a la instancia de Moodle del centro educativo al cual accede 2: El alumno 'a' debe estar inscripto en el curso 'c' 3: Existe una tarea publicada 't' en el curso 'c' 4: El alumno 'a' hizo previamente una entrega a la tarea 't'
Pasos	1: Iniciar sesión con el docente 'd' 2: Ir a la sección 'Grupos', allí se listarán las asignaturas donde 'd' es docente 3: Seleccionar el curso 'c', se listan las tareas del curso 'c' 4: Seleccionar la tarea 't', se listan los alumnos del curso 'c' 5: Seleccionar el alumno 'a', se muestran las entregas de 'a' en 't' 6: Seleccionar el archivo 'arch', se muestra el contenido 7: Click en 'Calificar' 8: Ingresar puntaje y detalle 9: Click en el botón 'Calificar' (botón verde)
Resultado esperado	1: Se muestra un mensaje del tipo: ' <i>Archivo evaluado</i> ' 2: El icono del archivo pasa a estar verde mostrando que esta evaluado
Post-Condicion	1: Se carga la tabla 'tiempoacceso' 2: Se carga la tabla 'registroaccesoarchivo' 3: Se carga la tabla 'registrotarea' 4: Se puede ver la corrección en Moodle por parte del docente
Resultado obtenido	Resultado esperado

Tabla 4.6: Test case [Listado y corrección de *assignments*]Corrección de tareas a un alumno

Casos de prueba adicionales de listado y corrección de assignments en el anexo.

## 4.4. Pruebas de regresión

A continuación se presenta un resumen de algunos de casos de prueba de regresión que se han realizado.

#### 4.4. Pruebas de regresión

Título	<b>[Prueba de regresión]Modificar configuración de usuario</b>
Resumen	Se busca que un usuario 'u' de Matefun logre modificar su configuración de usuario y estos cambios persistan luego de re-iniciar sesión.
Dependencias	Se logra ejecutar correctamente el caso de prueba de inicio de sesión.
Pre-condiciones	No tiene.
Datos	1: El usuario 'u' debe pertenecer a la instancia de Moodle del centro educativo al cual accede.
Pasos	1: Realizar login como alumno/docente 'u'. 2: Click en el botón <i>configuración</i> . 3: Modificar el tema <i>dracula</i> por <i>isotope</i> . 4: Cambiar el tamaño de fuente de <i>12px</i> a <i>24px</i> . 5: Cambiar el estado de todos los checkboxes. 6: Click en <i>Guardar</i> . 7: Cerrar sesión. 8: Iniciar sesión nuevamente con el usuario 'u'. 9: Click nuevamente en el botón <i>configuración</i> .
Resultado esperado	1: Luego del paso 6, se aplican los cambios. 2: Luego del paso 8, los cambios permanecen modificados. 3: Luego del paso 9, se logra ver en el panel de configuración los datos modificados.
Post-Condiciones	1: Se carga la tabla 'tiempoacceso'. 2: Se carga la tabla 'registrologin'. 3: Se carga la tabla 'configuracion'.
Resultado obtenido	Resultado esperado.

Tabla 4.7: [Prueba de regresión]Modificar configuración

Capítulo 4. Experimentación

Titulo	<b>[Prueba de regresión]Exportar archivo Matefun</b>
Resumen	Se busca que un usuario 'u' de Matefun logre exportar un archivo Matefun 'a' a su sistema de archivos local.
Dependencias	Se logra ejecutar correctamente el caso de prueba de carga de archivo privado.
Pre-condiciones	No tiene.
Datos	1: El usuario 'u' debe pertenecer a la instancia de Moodle del centro educativo al cual accede. 2: El usuario 'u' tiene un archivo 'a' en su sección de archivos privados.
Pasos	1: Realizar login como alumno/docente 'u'. 2: Navegar a la sección <i>Archivos</i> . 3: Seleccionar el archivo 'a'. 4: Click en el botón <i>editar</i> . 5: Click en el botón <i>exportar</i> . 6: Editar el archivo en matefun insertando un contenido diferente al original. 7: Click en el botón <i>exportar</i> .
Resultado esperado	1: Luego del paso 5, se descarga el archivo 'a'. 2: Luego del paso 7, se descarga el archivo 'a' con el contenido modificado.
Post-Condiciones	1: Se carga la tabla 'tiempoacceso'. 2: Se carga la tabla 'registrologin'.
Resultado obtenido	Resultado esperado.

Tabla 4.8: [Prueba de regresión]Exportar archivo Matefun

#### 4.4. Pruebas de regresión

Título	<b>[Prueba de regresión]Ejecución programa Matefun</b>
Resumen	Se busca que un usuario 'u' de Matefun logre cargar un archivo Matefun 'a' que contiene un programa válido y ejecutar funciones.
Dependencias	Se logra ejecutar correctamente el caso de prueba de carga de archivo privado.
Pre-condiciones	No tiene.
Datos	1: El usuario 'u' debe pertenecer a la instancia de Moodle del centro educativo al cual accede. 2: El usuario 'u' tiene un archivo 'a' en su sección de archivos privados cuyo contenido es un programa válido.
Pasos	1: Realizar login como alumno/docente 'u'. 2: Navegar a la sección <i>Archivos</i> . 3: Seleccionar el archivo 'a'. 4: Click en el botón <i>Editar</i> . 5: Click en el botón <i>Cargar programa</i> . 6: Ejecutar en la consola de la derecha alguna función definida en el programa ingresando correctamente sus parámetros de entrada.
Resultado esperado	1: Luego del paso 5, se modifica el <i>prompt</i> de la consola cargando el archivo. 2: Luego del paso 6, la función ejecutada devuelve el valor esperado.
Post-Condiciones	1: Se carga la tabla 'tiempoacceso'. 2: Se carga la tabla 'registrologin'.
Resultado obtenido	Resultado esperado.

Tabla 4.9: [Prueba de regresión]Ejecución programa Matefun

Capítulo 4. Experimentación

Titulo	<b>[Prueba de regresión]Reiniciar interprete</b>
Resumen	Se busca que un usuario 'u' de Matefun logre reiniciar el interprete de Matefun.
Dependencias	Se logra ejecutar correctamente el caso de prueba de ejecución de programa Matefun.
Pre-condiciones	No tiene.
Datos	1: El usuario 'u' debe pertenecer a la instancia de Moodle del centro educativo al cual accede. 2: El usuario 'u' tiene un archivo 'a' en su sección de archivos privados cuyo contenido es un programa válido.
Pasos	1: Realizar login como alumno/docente 'u'. 2: Navegar a la sección <i>Archivos</i> . 3: Seleccionar el archivo 'a'. 4: Click en el botón <i>Editar</i> . 5: Click en el botón <i>Cargar programa</i> . 6: Click en el botón <i>Reiniciar interprete</i> .
Resultado esperado	1: Luego del paso 5, se modifica el <i>prompt</i> de la consola cargando el archivo 'a'. 2: Luego del paso 6, se reinicia el interprete quedando como resultado que no hay archivo cargado.
Post-Condiciones	1: Se carga la tabla 'tiempoacceso'. 2: Se carga la tabla 'registrologin'.
Resultado obtenido	Resultado esperado.

Tabla 4.10: [Prueba de regresión]Reiniciar interprete

#### 4.4. Pruebas de regresión

Título	<b>[Prueba de regresión]Ejecutar atajos de teclado</b>
Resumen	Se busca que un usuario 'u' de Matefun logre ejecutar los atajos de teclado definidos.
Dependencias	Se logra ejecutar correctamente el caso de prueba de carga de archivo privado.
Pre-condiciones	No tiene.
Datos	1: El usuario 'u' debe pertenecer a la instancia de Moodle del centro educativo al cual accede. 2: El usuario 'u' tiene un archivo 'a' en su sección de archivos privados.
Pasos	1: Realizar login como alumno/docente 'u'. 2: Navegar a la sección <i>Archivos</i> . 3: Seleccionar el archivo 'a'. 4: Click en el botón <i>Editar</i> . 5: Cambiar el contenido del archivo cargado 'a'. 6: Presionar <i>CTRL + G</i> . 7: Presionar <i>CTRL + P</i> . 8: Presionar <i>CTRL + R</i> . 9: Presionar <i>CTRL + E</i> . 10: Presionar <i>CTRL + O</i> . 11: Presionar <i>CTRL + A</i> .
Resultado esperado	1: Luego del paso 6, se muestra la notificación del archivo guardado, guardando el mismo. 2: Luego del paso 7, se carga el archivo 'a' en la consola en caso de ser un programa válido. 3: Luego del paso 8, se reinicia el interprete quedando como resultado que no hay archivo cargado. 4: Luego del paso 9, se descarga el archivo 'a' con el contenido que aparece en la izquierda. 5: Luego del paso 10, se abre el menú de configuración. 6: Luego del paso 11, se abre el modal de creación de nuevo archivo.
Post-Condiciones	1: Se carga la tabla 'tiempoacceso'. 2: Se carga la tabla 'registrologin'.
Resultado obtenido	Resultado esperado.

Tabla 4.11: [Prueba de regresión]Ejecutar atajos de teclado

Capítulo 4. Experimentación

Titulo	<b>[Prueba de regresión]Cargar figura 2D</b>
Resumen	Se busca que un usuario 'u' de Matefun logre ejecutar una función que carga una figura 2D.
Dependencias	Se logra ejecutar correctamente el caso de prueba de ejecución programa Matefun.
Pre-condiciones	No tiene.
Datos	1: El usuario 'u' debe pertenecer a la instancia de Moodle del centro educativo al cual accede. 2: El usuario 'u' tiene un archivo 'a' en su sección de archivos privados cuyo contenido es un programa válido e incluye una función 'f' que dibuja una figura 2D.
Pasos	1: Realizar login como alumno/docente 'u'. 2: Navegar a la sección <i>Archivos</i> . 3: Seleccionar el archivo 'a'. 4: Click en el botón <i>Editar</i> . 5: Click en el botón <i>Cargar programa</i> . 6: Ejecutar la función 'f' con parámetros validos.
Resultado esperado	1: Luego del paso 6, Se cambia automáticamente de la pestaña 'Programa' a la pestaña 'Figuras 2D' cargando la figura 2D correspondiente.
Post-Condiciones	1: Se carga la tabla 'tiempoacceso'. 2: Se carga la tabla 'registrologin'.
Resultado obtenido	Resultado esperado.

Tabla 4.12: [Prueba de regresión]Cargar figura 2D

## Capítulo 5

# Conclusiones y trabajo futuro

A continuación se mencionan las conclusiones obtenidas luego de finalizar el proyecto y algunas de las posibilidades que se pueden poner en practica en caso que futuros proyectos basados en Matefun se deseen implementar.

### 5.1. Conclusiones

Al iniciar el proyecto se tuvo que realizar una recopilación de información sobre los posibles sistemas de gestión de aprendizaje en el mercado para dilucidar cual era la mejor opción para conectar con Matefun, se considera que la opción elegida fue satisfactoria aunque al inicio se esperaba más de Moodle.

Las opciones reales en si eran pocas en el sentido que la mayoría de las alternativas (otros sistemas LMS) carecen de formas de integrarse con sistemas externos tales como puede ser una API rest/soap hacia la cual se pudieran realizar *requests* para interactuar con el LMS, la elección se definía entre Moodle y Schoology.

Luego que Moodle fue el sistema LMS elegido para interoperar con Matefun surgió la necesidad de elegir entre dos formas de integración: convertir a Matefun un un *plugin* de Moodle o convertir a Matefun en un *consumidor* de los servicios que Moodle provee públicamente. Finalmente se tomó la decisión de escoger la segunda alternativa pensando principalmente en la mantenibilidad y compatibilidad.

Cuando se manifiesta que *se esperaba más de Moodle* se hace en referencia a que la API pública carece de funciones que hubieran permitido a Matefun realizar una mejor implementación de algunas funcionalidades, por ejemplo, ésta API no tiene funciones que contemplen el manejo de archivos compartidos nativamente, para implementar este requerimiento se utilizaron los archivos privados de un usuario particular como archivos compartidos. Por otro lado tampoco se permite eliminar o mover archivos privados, para implementar esta funcionalidad se hizo uso de la base de datos de Matefun para eliminar/mover archivos de forma lógica.

Luego de elegir Moodle se notó que faltan funciones necesarias en la API de forma nativa (recordando que el sistema LMS subyacente no podía ser modificado) por lo que algunas de las funcionalidades no pudieron ser implementadas

## Capítulo 5. Conclusiones y trabajo futuro

(ejemplo: gestión de evaluaciones).

Dejando de lado de estas complicaciones que no se pudieron prever con anticipación, se logró implementar una parte importante de las funcionalidades requeridas, quedando sin realizarse la gestión de evaluaciones (crear una nueva *tarea*) por parte de un docente directamente en Matefun. Las actividades que se pueden realizar en el aula con Matefun ahora tienen respaldo en Moodle, sobre todo las actividades que realizan los alumnos que son en sí las más importantes y en lo que se debe enfocar principalmente la plataforma.

### 5.2. Trabajo Futuro

A continuación se muestran algunas de las posibilidades de extensión para Matefun, más allá que la actualización de dependencias se considera la de mayor prioridad, el resto de las propuestas no guardan ningún orden de prioridad particular.

#### 5.2.1. Actualización de dependencias

Como se mencionó en los demás documentos asociados a este proyecto, es muy importante actualizar las dependencias sobre las cuales corre Matefun. Al momento de escribir este documento las últimas versiones de las dependencias más importantes son las siguientes:

- Frontend
  1. Angular 12.1.4 (Matefun utiliza 4.0.0) [1]
  2. Node.js 14.17.5 LTS (Matefun utiliza 10.1.0) [48]
  3. TypeScript 4.3 (Matefun utiliza 2.2.0) [79]
- Backend
  1. WildFly 24.0.1 (Matefun utiliza 10.1.0) [59]

Angular ha sido heredado en este proyecto desde su implementación original en su versión 4, versión que se encuentra actualmente sin mantenimiento. Al momento de la edición de este documento Angular ya ha sido lanzado en su versión 12 [29].

Las dependencias arriba enumeradas son las más importantes y las que por lo tanto deben ser actualizadas antes de comenzar un nuevo proyecto basado en Matefun. Opcionalmente se puede considerar la actualización de las versiones de JavaSE/EE.

#### Justificación de la falta de actualización

La razón por la que las dependencias no fueron actualizadas radica en que al mismo tiempo que se desarrollaba este proyecto, otros proyectos de grado basados en la interfaz web de Matefun (frontend y backend) también se estaban desarrollando.

Actualizar las dependencias en estas condiciones podría llevar a incompatibilidades porque dichos proyectos utilizaron las mismas bibliotecas y/o frameworks para su desarrollo, además de ser una tarea que toma un tiempo considerablemente alto para no estar dentro de los requerimientos iniciales.

### 5.2.2. Movimientos/eliminaciones físicas

Como se mencionó oportunamente en este documento, Moodle no permite realizar eliminaciones o movimientos de archivos privados utilizando el conjunto web services provistos nativamente, las eliminaciones que se implementan en este proyecto son eliminaciones lógicas.

Por esta causa es que se propone crear un plugin de Moodle que introduzca estos web services necesarios, así como también hacer las modificaciones necesarias en el backend de Matefun para que se pueda implementar esta funcionalidad.

Eventualmente al momento de distribuir Matefun, esto se haría en conjunto al plugin desarrollado.

### 5.2.3. Merge de cambios de archivos compartidos utilizando git

Supongamos que se da el escenario en que un usuario **u1** tiene bloqueado y esta editando el archivo compartido **f1** a la vez que el usuario **u2** tiene cargado el mismo archivo compartido **f1** y también lo está editando (no puede guardar cambios porque no lo tiene bloqueado para sí mismo).

Como fue explicado anteriormente, en el momento que el usuario **u1** guarde sus cambios en **f1** y libere el archivo, al usuario **u2** se le presentarán solamente las siguientes opciones:

- Mantener su copia local
- Cargar íntegramente la versión que el usuario **u1** guardó recientemente en Moodle.

pero también puede ser interesante obtener un *merge* de esas dos versiones.

**GIT**, la herramienta de versionado de este proyecto, incluye la utilidad *git-merge-file* que permite hacer un merge de dos archivos independientes siempre y cuando tengan un ancestro común, es decir una misma versión de partida. Simplemente se debe utilizar el siguiente comando:

```
git merge-file -p v1.mf ancestro.mf v2.mf > destino.mf
```

Este comando permite realizar un merge de dos archivos y enviar el resultado a un archivo de destino. En caso de surgir conflictos, estos quedan marcados en el mismo archivo de destino.

### 5.2.4. Compartir archivos utilizando plugin/addon

Si bien la implementación de *compartir archivos* es completa en el sentido de los posibles destinatarios del archivo, existen diferentes métodos que pueden ser utilizados para compartir archivos sin instalar nuevos plugins de Moodle, o sea nativamente.

## Capítulo 5. Conclusiones y trabajo futuro

Se puede utilizar las actividades *Database/Forum* [42] o utilizando el resource *folder* [77], pero estas opciones requieren mucha configuración del lado de la instancia Moodle del centro educativo (en particular en el curso) y tampoco permiten compartir archivos entre dos alumnos que pertenezcan a cursos distintos.

Existen varios plugins que pueden ser utilizados para mejorar la implementación de compartir archivos, tales como *Student folder* [11] o *Public files* [78] pero ninguno ofrece la funcionalidad que Matefun implementa (compartir a usuario/-grupo/curso), por esa razón es que se aconseja crear un nuevo plugin para poder tener esta funcionalidad y que sea instalado junto a Matefun.

### 5.2.5. Soporte Schoology

Como se menciono anteriormente Schoology al igual que Moodle también es un sistema LMS ampliamente usado el cual también contiene API's para desarrolladores accesible públicamente [71]. Como trabajo futuro se puede considerar también implementar la integración de centros educativos que utilicen Schoology.

# Capítulo 6

## Anexos

### 6.1. Modelo Entidad-Relación

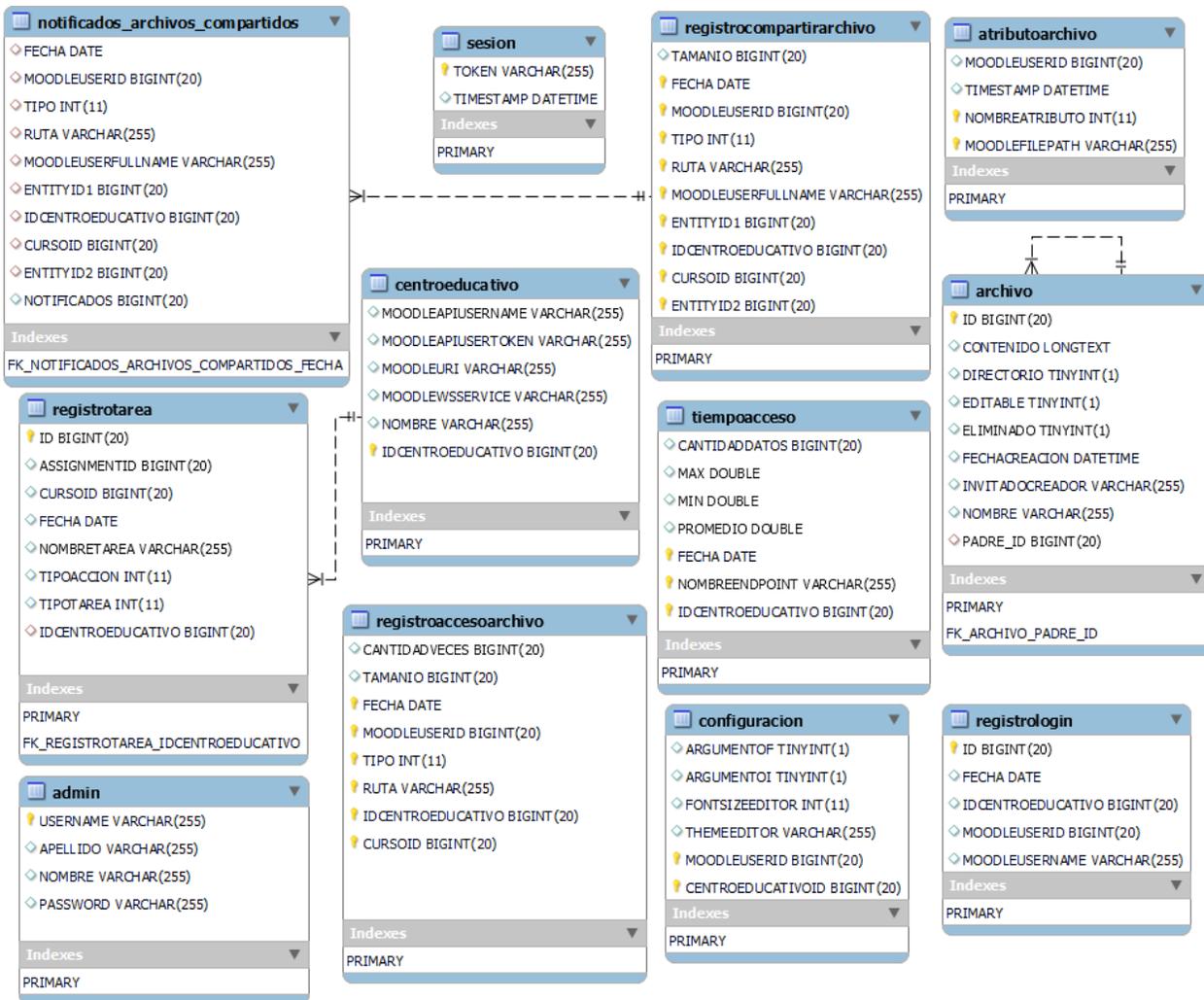


Figura 6.1: Modelo de entidad-relación mostrando todas las entidades que maneja Matefun.

## 6.2. Casos de prueba realizados

### 6.2.1. Login

Titulo	<b>Ingresar datos de prueba</b>
Resumen	Se busca que se puedan ingresar los datos de prueba de la aplicación (centros educativos y administrador de Matefun)
Dependencias	No tiene
Pre-condiciones	Asegurarse que es el primer inicio (no hay datos previamente configurados)
Pasos	1: Ingresar al sistema 2: Click en <i>cargar datos de prueba</i>
Datos	No debe haber administrador ni centros educativos pre cargados
Resultado esperado	Se recarga la pagina de login nuevamente
Post-Condiciones	Se cargan las tablas con los datos
Resultado obtenido	Resultado esperado

Tabla 6.1: Test case [Login]Ingresar datos de prueba

Titulo	<b>Ingresar administrador de Matefun</b>
Resumen	Se busca que se pueda ingresar manualmente el administrador de Matefun (sin cargar datos de prueba)
Dependencias	No tiene
Pre-condiciones	Asegurarse que es el primer inicio (no hay datos previamente configurados)
Pasos	1: Introducir en los campos <i>nombre de usuario, contraseña y confirmación</i> 2: Click en 'Cargar admin'
Datos	No debe haber administrador pre cargado
Resultado esperado	Se inicia sesión (automáticamente) redirigiendo a la pagina de carga de centros educativos
Post-Condiciones	Se carga la tabla 'admin' con los datos del administrador
Resultado obtenido	Resultado esperado

Tabla 6.2: Test case [Login]Ingresar administrador

## 6.2. Casos de prueba realizados

Titulo	<b>Intento fallido de inicio de sesión de usuario</b>
Resumen	Se busca que por error en los datos no se logre iniciar sesión
Dependencias	En la instancia de Moodle del centro educativo utilizado debe estar correctamente configurado el alumno o docente con el cual se desea iniciar sesión según [65]
Pre-condiciones	Existe al menos un centro educativo cargado (tabla 4.2)
Pasos	1: Ir a la pagina de login 2: Introducir nombre de usuario existente y contraseña incorrecta/vacía 3: Seleccionar centro educativo (si hay más de uno cargado) 4: Click en 'Iniciar sesión'
Datos	no tiene
Resultado esperado	Se muestra un mensaje de error ' <i>Error! Usuario no encontrado/mal configurado o contraseña incorrecta.</i> '
Post-Condiciones	No tiene
Resultado obtenido	Resultado esperado

Tabla 6.3: Test case [Login]Login usuario fallido

### 6.2.2. Gestión de centros educativos

Titulo	<b>Eliminar centro educativo</b>
Resumen	Se busca eliminar un centro educativo existente
Dependencias	Debe existir un centro educativo ingresado mediante el caso de prueba 4.2
Pre-condiciones	No tiene
Pasos	1: Iniciar sesión con administrador de Matefun 2: Ir a la sección de gestión de centros educativos 3: En la lista de la derecha seleccionar un centro educativo y hacer click en el icono de eliminar (papelera) 4: Click en <i>OK</i> en el popup de confirmación
Datos	No tiene
Resultado esperado	Se elimina el centro educativo de la lista de la derecha
Post-Condiciones	Se borra el centro educativo de la tabla 'centroeducativo' Se carga la tabla 'tiempoacceso'
Resultado obtenido	Resultado esperado

Tabla 6.4: Test case [Gestión de centros educativos]Borrado de centro educativo

## Capítulo 6. Anexos

Titulo	<b>Edición de centros educativos</b>
Resumen	Se busca editar atributos de un centro educativo existente
Dependencias	Debe debe existir un centro educativo ingresado mediante el caso de prueba 4.2
Pre-condiciones	No tiene
Pasos	1: Iniciar sesión con administrador de Matefun 2: Ir a la pagina de gestión de centros educativos 3: En la lista de la derecha seleccionar un centro educativo y hacer click en el icono de lápiz (editar) 4: Editar nombre de centro educativo 5: Click en el icono de guardar (diskette)
Datos	No tiene
Resultado esperado	Se editan los valores modificados del centro educativo de la lista de la derecha
Post-Condiciones	1: Se editan los valores modificados del centro educativo de la tabla 'centroeducativo' 2: Se carga la tabla 'tiempoacceso'
Resultado obtenido	Resultado esperado

Tabla 6.5: Test case [Gestión de centros educativos]Edición de centro educativo

### 6.2.3. Mapeo de archivos y carpetas

Titulo	<b>Carga de contenido de archivo privado</b>
Resumen	Se busca que Matefun logre cargar el contenido de un archivo que se encuentra dentro del directorio raíz del sistema de archivos privados de un alumno
Dependencias	El caso de prueba de carga de directorio raíz (tabla 4.3) debe funcionar correctamente
Pre-condiciones	No tiene
Pasos	1: Realizar login como alumno 2: Ir a la sección de archivos 3: Clic en un archivo listado
Datos	1: El alumno debe pertenecer' al centro educativo al cual inicia sesión 2: El alumno debe tener archivos privados en el directorio raíz en la instancia de Moodle 3: El archivo tiene contenido no vacío
Resultado esperado	Se carga exactamente el mismo contenido en el panel de la derecha que el que tiene el archivo en la instancia de Moodle
Post-Condiciones	1: Se carga la tabla 'tiempoacceso' 2: Se carga la tabla 'registroaccesoarchivo'
Resultado obtenido	Resultado esperado

Tabla 6.6: Test case [Mapeo de archivos y carpetas]Carga de contenido de archivo privado

## 6.2. Casos de prueba realizados

Titulo	<b>Carga de contenido de archivo 'arch' de curso 'c'</b>
Resumen	Se busca que Matefun logre cargar el contenido de un archivo 'arch' que se encuentra en la pagina de un curso 'c' en Moodle
Dependencias	Se logra ejecutar correctamente el caso de prueba de inicio de sesión
Pre-condiciones	No tiene
Datos	1: El alumno debe pertenecer a la instancia de Moodle del centro educativo al cual accede 2: El alumno debe estar inscripto en un curso 'c' 3: Debe existir al menos un archivo 'arch' (de contenido no vacío) compartido como recurso en el curso 'c' en Moodle
Pasos	1: Realizar login como alumno 2: Ir a la sección de archivos 3: Ir a la tab 'cursos' 4: Clic en el archivo 'arch'
Resultado esperado	1: El archivo 'arch' se muestra en el listado 2: En el listado debajo del nombre del archivo 'arch' aparece el nombre del curso 'c' 3: Al hacer clic en el archivo 'arch' se muestra su contenido en el panel de la derecha
Post-Condicion	1: Se carga la tabla 'tiempoacceso' 2: Se carga la tabla 'registroaccesoarchivo'
Resultado obtenido	Resultado esperado

Tabla 6.7: Test case [Mapeo de archivos y carpetas]Carga de contenido de archivo de curso

### 6.2.4. Gestión de archivos compartidos

Titulo	<b>Visualización de notificaciones de archivos compartidos</b>
Resumen	Se busca que el usuario 'u2' logre visualizar una notificación luego que el usuario 'u1' comparte un archivo 'arch' a un curso 'c' al cual los usuarios 'u1' y 'u2' pertenecen
Dependencias	Se logra ejecutar correctamente el caso de compartir archivo a curso, tabla 4.4
Pre-condiciones	No existe un inicio de sesión de 'u2' entre que se comparte el archivo 'arch' y se ejecuta este caso de prueba
Datos	1: El usuario 'u2' debe pertenecer a la instancia de Moodle del centro educativo al cual accede. 2: El usuario 'u2' debe estar inscripto en el curso 'c'
Pasos	Iniciar sesión con el usuario 'u2'
Resultado esperado	Inmediatamente luego de iniciar sesión se muestra un mensaje del tipo: 'El día <i>fecha</i> el usuario <i>nombre u1</i> ha compartido el archivo <i>nombre archivo 'a'</i> a el curso <i>nombre 'c'</i> que ud. pertenece.'
Post-Condicion	1: Se carga la tabla 'registroaccesoarchivo' 2: Se carga la tabla 'notificados_archivos_compartidos'
Resultado obtenido	Resultado esperado

Tabla 6.8: Test case [Gestión de archivos compartido]Visualización de notificaciones

Capítulo 6. Anexos

Titulo	<b>Acceso de archivo compartido</b>
Resumen	Se busca que el usuario 'u2' logre acceder al archivo 'arch' que compartió el usuario 'u1' a un curso 'c' al cual los usuarios 'u1' y 'u2' pertenecen
Dependencias	Se logra ejecutar correctamente el caso de compartir archivo a curso, tabla 4.4
Pre-condiciones	No tiene
Datos	1: El usuario 'u2' debe pertenecer a la instancia de Moodle del centro educativo al cual accede 2: El usuario 'u2' debe estar inscripto en el curso 'c'
Pasos	1:Iniciar sesión con el usuario 'u2' 2: Ir a la sección 'Archivos' 3: Ir a la pestaña 'Compartidos' 4: Click en el archivo 'arch'
Resultado esperado	Se carga el mismo contenido del archivo que existe en Moodle
Post-Condiciones	1: Se carga la tabla 'tiempoacceso' 2: Se carga la tabla 'registroaccesoarchivo'
Resultado obtenido	Resultado esperado

Tabla 6.9: Test case [Gestión de archivos compartido]Acceso de archivo compartido

## 6.2. Casos de prueba realizados

### 6.2.5. Entrega de archivos para assignments

Titulo	<b>Intento fallido de entrega de archivos a <i>assignment</i></b>
Resumen	Se busca que el alumno 'a' <b>NO</b> logre hacer una entrega de un archivo 'arch' para una tarea publicada 't' en el curso 'c' porque se encuentra vencido el plazo de entrega
Dependencias	Se logra ejecutar correctamente el caso de uso de inicio de sesión
Pre-condiciones	1: Existe una tarea ( <i>assignment</i> ) publicada 't' en el curso 'c' del centro educativo en el cual se accede. 2: La tarea 't' no se encuentra habilitada para entregas al momento de ejecutar el caso de prueba porque el campo <i>due date</i> en Moodle es anterior al día de la ejecución de este caso de prueba
Datos	1: El alumno 'a' debe pertenecer a la instancia de Moodle del centro educativo al cual accede. 2: El alumno 'a' debe estar inscripto en el curso 'c'
Pasos	1: Iniciar sesión con el alumno 'a' 2: Ir a la sección 'Archivos' 3: Seleccionar el archivo 'arch' en la pestaña 'Mis archivos' 4: Click en el botón 'Entregar' 5: Seleccionar el curso 'c' 6: Seleccionar la actividad ( <i>assignment</i> ) 't'
Resultado esperado	1: Se muestra un mensaje del tipo: ' <i>No puedes realizar la entrega, el periodo venció en: {fecha}</i> ' 2: El botón de entrega se encuentra inactivo y no responde.
Post-Condiciones	1: Se carga la tabla 'tiempoacceso'
Resultado obtenido	Resultado esperado

Tabla 6.10: Test case [Entrega de archivos para *assignments*]Intento fallido de entrega de archivo

6.2.6. Listado y corrección de assignments

Titulo	<b>Consulta de corrección por parte de un docente</b>
Resumen	Se busca que un docente 'd1' consulte datos de la corrección de un archivo 'arch' hecha a un alumno 'a' por un docente 'd2' en el marco de una tarea individual 't' dentro de un curso 'c'
Dependencias	1: Se logran ejecutar correctamente los casos de uso de inicio de sesión, entrega de archivos y correccion de assignments.
Pre-condiciones	1: Existe una tarea individual publicada 't' en el curso 'c' del centro educativo en el cual se accede. 2: El alumno 'a' ha hecho una entrega de un archivo 'arch' a la tarea 't' de 'c'. 3: El docente 'd2' ha corregido la tarea 't' de 'c' para el alumno 'a'.
Datos	1: El alumno 'a' debe pertenecer a la instancia de Moodle del centro educativo al cual accede. 2: El alumno 'a' debe estar inscripto en el curso 'c'.
Pasos	1: Iniciar sesión con el docente 'd'. 2: Ir a la sección 'Grupos'. 3: Seleccionar el curso 'c'. 4: Seleccionar la tarea 't'. 5: Seleccionar el alumno 'a'. 6: Seleccionar el archivo 'arch'. 7: Click en 'Ver Detalle'.
Resultado esperado	1: Luego del paso 5 el archivo 'arch' aparece con icono verde. 2: Luego del paso 7 se muestra un modal con la nota y la descripción.
Post-Condiciones	1: Se carga la tabla 'tiempoacceso'. 2: Se carga la tabla 'registrologin'.
Resultado obtenido	Resultado esperado.

Tabla 6.11: Test case [Listado y corrección de *assignments*]Consulta de corrección por parte de un docente

## 6.2. Casos de prueba realizados

Título	<b>Consulta de corrección por parte de un alumno</b>
Resumen	Se busca que un alumno 'a' consulte datos de la corrección de un archivo 'arch' entregado por si mismo en el marco de una tarea individual 't' dentro de un curso 'c'
Dependencias	1: Se logran ejecutar correctamente los casos de uso de inicio de sesión, entrega de archivos y correccion de assignments.
Pre-condiciones	1: Existe una tarea individual publicada 't' en el curso 'c' del centro educativo en el cual se accede. 2: El alumno 'a' ha hecho una entrega de un archivo 'arch' a la tarea 't' de 'c'. 3: Algún docente ha corregido la tarea 't' de 'c' para el alumno 'a'.
Datos	1: El alumno 'a' debe pertenecer a la instancia de Moodle del centro educativo al cual accede. 2: El alumno 'a' debe estar inscripto en el curso 'c'.
Pasos	1: Iniciar sesión con el alumno 'a'. 2: Ir a la sección 'Archivos'. 3: Click en el botón 'Ver Calificaciones'. 4: Seleccionar el curso 'c'. 5: Seleccionar la tarea 't'.
Resultado esperado	1: Luego del paso 5, se muestran los datos de corrección del archivo 'arch' de la tarea 't' en el curso 'c', nombre de archivo, fecha de entrega, fecha de corrección, calificación e información de corrección.
Post-Condicioness	1: Se carga la tabla 'tiempoacceso'. 2: Se carga la tabla 'registrologin'.
Resultado obtenido	Resultado esperado.

Tabla 6.12: Test case [Listado y corrección de *assignments*]Consulta de corrección por parte de un alumno

# Índice de tablas

2.1. Tabla comparativa de sistemas LMS. . . . .	11
3.1. Tabla de requerimientos y resultados. . . . .	23
4.1. Test case [Login]Login usuario correcto . . . . .	40
4.2. Test case [Gestión de centros educativos]Ingreso de centro educativo . . . . .	40
4.3. Test case [Mapeo de archivos y carpetas]Carga de archivos privados . . . . .	41
4.4. Test case [Gestión de archivos compartidos]Compartir un archivo a un curso . . . . .	42
4.5. Test case [Entrega de archivos para <i>assignments</i> ]Entrega tarea . . . . .	43
4.6. Test case [Listado y corrección de <i>assignments</i> ]Corrección de ta- reas a un alumno . . . . .	44
4.7. [Prueba de regresión]Modificar configuración . . . . .	45
4.8. [Prueba de regresión]Exportar archivo Matefun . . . . .	46
4.9. [Prueba de regresión]Ejecución programa Matefun . . . . .	47
4.10. [Prueba de regresión]Reiniciar interprete . . . . .	48
4.11. [Prueba de regresión]Ejecutar atajos de teclado . . . . .	49
4.12. [Prueba de regresión]Cargar figura 2D . . . . .	50
6.1. Test case [Login]Ingresar datos de prueba . . . . .	56
6.2. Test case [Login]Ingresar administrador . . . . .	56
6.3. Test case [Login]Login usuario fallido . . . . .	57
6.4. Test case [Gestión de centros educativos]Borrado de centro edu- cativo . . . . .	57
6.5. Test case [Gestión de centros educativos]Edición de centro educativo . . . . .	58
6.6. Test case [Mapeo de archivos y carpetas]Carga de contenido de archivo privado . . . . .	58
6.7. Test case [Mapeo de archivos y carpetas]Carga de contenido de archivo de curso . . . . .	59
6.8. Test case [Gestión de archivos compartido]Visualización de noti- ficaciones . . . . .	59
6.9. Test case [Gestión de archivos compartido]Acceso de archivo com- partido . . . . .	60
6.10. Test case [Entrega de archivos para <i>assignments</i> ]Intento fallido de entrega de archivo . . . . .	61
6.11. Test case [Listado y corrección de <i>assignments</i> ]Consulta de co- rrección por parte de un docente . . . . .	62

6.12. Test case [Listado y corrección de *assignments*]Consulta de corrección por parte de un alumno . . . . . 63

# Índice de figuras

3.1. Diagrama de distribución del sistema. . . . .	21
3.2. Requerimiento <i>existencia de usuario administrador</i> . . . . .	24
3.3. Gestión de centros educativos. . . . .	25
3.4. Pantalla de login y tabla para nuevos centros educativos . . . . .	26
3.5. Mapeo de archivos y carpetas en Matefun (arriba) y Moodle (abajo). . . . .	27
3.6. Formato en el nombre de archivos compartidos visualizados en Moodle. . . . .	28
3.7. Tabla para manejar los atributos de los archivos tales como eliminado/readonly/bloqueado. . . . .	32
3.8. Tablas utilizadas en la pestaña general. . . . .	34
3.9. Tabla donde se guardan los tiempos de acceso para consultas Moodle/Matefun. . . . .	35
3.10. Vista de la pestaña tiempos de acceso. . . . .	36
3.11. Vista de la pestaña estado de servidores. . . . .	36
6.1. Modelo de entidad-relación mostrando todas las entidades que maneja Matefun. . . . .	55

# Referencias

- [1] Angular. Angular on github. <https://github.com/angular/angular/releases/tag/12.1.4>. [Accedido 17-Ago-2021].
- [2] apereo.org. Apero. <https://www.apereo.org/>. [Accedido 11-Nov-2021].
- [3] Babel. Babel. <https://babeljs.io/>. [Accedido 23-May-2021].
- [4] Ellen D. Wagner Beth Davis, Colleen Carmean. The evolution of the lms: From management to learning. Página 4. [Accedido 15-Oct-2021].
- [5] BetterBuys.com. Google Classroom Review. <https://www.betterbuys.com/lms/reviews/google-classroom/>. [Accedido 11-Nov-2021].
- [6] businesswire.com. Noodle partners changes corporate brand name to noodle. <https://www.businesswire.com/news/home/20201118005936/en/Noodle-Partners-Changes-Corporate-Brand-Name-to-Noodle>. [Accedido 10-Nov-2021].
- [7] Visual Studio Code. Visual Studio Code. <https://code.visualstudio.com/>. [Accedido 23-May-2021].
- [8] World Wide Web Consortium. Introduction to CSS3. <https://www.w3.org/TR/2001/WD-css3-roadmap-20010523/>. [Accedido 13-Dic-2021].
- [9] World Wide Web Consortium. About W3c. <https://www.w3.org/Consortium/>, 2004. [Accedido 20-May-2021].
- [10] World Wide Web Consortium. Web Services Glossary. <https://www.w3.org/TR/2004/NOTE-ws-gloss-20040211/#webservice>, 2004. [Accedido 20-May-2021].
- [11] AMC Academic Moodle Cooperation. Student folder module. [https://moodle.org/plugins/mod\\_publication](https://moodle.org/plugins/mod_publication). [Accedido 19-Ago-2021].
- [12] Programa de entornos virtuales de aprendizaje. Moodle. <https://proeva.udelar.edu.uy/moodle/>. [Accedido 13-Dic-2021].
- [13] Deno. Deno. <https://deno.land/>. [Accedido 23-May-2021].
- [14] Google developers. Classroom API. <https://developers.google.com/classroom/reference/rest>. [Accedido 11-Nov-2021].

## Referencias

- [15] die.net. curl manual page. <https://linux.die.net/man/1/curl>. [Accedido 24-Jul-2021].
- [16] die.net. lsof manual page. <https://linux.die.net/man/8/lsof>. [Accedido 24-Jul-2021].
- [17] die.net. top command manual page. <https://linux.die.net/man/1/top>. [Accedido 24-Jul-2021].
- [18] edsurge.com. Edmodo Offers API To Developers Who Publish on Site. <https://www.edsurge.com/news/edmodo-opens-up-api-to-developers>. [Accedido 11-Nov-2021].
- [19] elearningindustry.com. Sakai pricing. <https://elearningindustry.com/directory/elearning-software/sakai/pricing>. [Accedido 11-Nov-2021].
- [20] eLiterate. Academic LMS Market Share: A view across four global regions. <https://eliterate.us/academic-lms-market-share-view-across-four-global-regions>. [Accedido 23-Jul-2021].
- [21] eltiempo.com. Edmodo, la red social de la educación. <https://www.eltiempo.com/archivo/documento/CMS-13579131>. [Accedido 10-Nov-2021].
- [22] enlyft.com. Academic Learning Management products . <https://enlyft.com/tech/academic-learning-management>. [Accedido 11-Nov-2021].
- [23] Beatriz Pérez et al. Testing exploratorio en la práctica. <https://ces.com.uy/wp-content/uploads/2007/02/JIISIC-2007.pdf>. [Accedido 03-Set-2021].
- [24] Femxa. Edmodo. la plataforma educativa gratuita n<sup>o</sup>1. <https://www.grupofemxa.es/la-plataforma-educativa-gratuita-no1-edmodo/>. [Accedido 10-Nov-2021].
- [25] Typescript for visual studio code. Typescript. <https://code.visualstudio.com/docs/languages/typescript>. [Accedido 23-May-2021].
- [26] Moodle.org forums. No API to create an assignment/module. <https://moodle.org/mod/forum/discuss.php?d=354690>. [Accedido 20-Ago-2021].
- [27] GNU. GPLv3 licence. <https://www.gnu.org/licenses/gpl-3.0.html>. [Accedido 18-Jul-2021].
- [28] Martín Méndez Gonzalo Cameto. Presentación de Matefun en Congreso de Educación (Salto). [https://gitlab.fing.edu.uy/matefun/Frontend/uploads/8241134beadb825fb524e7a6a56acf83/matefun\\_congreso.pdf](https://gitlab.fing.edu.uy/matefun/Frontend/uploads/8241134beadb825fb524e7a6a56acf83/matefun_congreso.pdf). [Accedido 20-Ago-2021].

- [29] Google. Angular 12. <https://github.com/angular/angular/releases/tag/12.0.0>. [Accedido 30-May-2021].
- [30] Google. Google Support forums. <https://support.google.com/edu/classroom/thread/4811970/hola-me-gustaria-saber-%C2%BFqu%C3%A9-tipo-de-licencia-tiene-el-servicio-de-google-classroom?hl=en>. [Accedido 11-Nov-2021].
- [31] Software Testing Help. Sample Test Case Template With Test Case Examples. <https://www.softwaretestinghelp.com/test-case-template-examples/>. [Accedido 07-Set-2021].
- [32] html.com. HTML5. [https://html.com/html5/#What\\_is\\_HTML5](https://html.com/html5/#What_is_HTML5). [Accedido 23-May-2021].
- [33] IETF. The websocket protocol. <https://datatracker.ietf.org/doc/html/rfc6455>. [Accedido 02-Jul-2021].
- [34] Business Insider. Edmodo: cómo funciona la mejor plataforma online gratis para el aprendizaje a distancia. <https://www.businessinsider.es/edmodo-como-funciona-plataforma-educativa-aula-859237>. [Accedido 10-Nov-2021].
- [35] Moodle issue tracker. core\_course\_create\_modules. <https://tracker.moodle.org/browse/MDL-40779>. [Accedido 20-Ago-2021].
- [36] Moodle issue tracker. New webservice: add/update assignment in course. <https://tracker.moodle.org/browse/MDL-63577>. [Accedido 20-Ago-2021].
- [37] ECMAScript language. ECMAScript language. <https://tc39.es/ecma262/#sec-overview>. [Accedido 23-May-2021].
- [38] Moodle. Moodle acronym. [https://docs.moodle.org/24/en/About\\_Moodle#:~:text=The%20word%20Moodle%20was%20originally,to%20programmers%20and%20education%20theorists](https://docs.moodle.org/24/en/About_Moodle#:~:text=The%20word%20Moodle%20was%20originally,to%20programmers%20and%20education%20theorists). [Accedido 11-Nov-2021].
- [39] Moodle. Moodle licence. <https://docs.moodle.org/dev/License>. [Accedido 18-Jul-2021].
- [40] Moodle. Web service API functions. [https://docs.moodle.org/dev/Web\\_service\\_API\\_functions](https://docs.moodle.org/dev/Web_service_API_functions). [ver funciones del area **mod\_assign**, Accedido 20-Ago-2021].
- [41] Moodlecloud. Moodlecloud. <https://moodlecloud.com/app/en/login>. [Accedido 11-Nov-2021].
- [42] moodle.org. discussion forums. <https://moodle.org/mod/forum/discuss.php?d=344320>. [Accedido 18-Ago-2021].
- [43] moodle.org. moodle downloads. <https://download.moodle.org/>. [Accedido 11-Nov-2021].

## Referencias

- [44] moodle.org. Moodle history. <https://docs.moodle.org/311/en/History>. [Accedido 10-Nov-2021].
- [45] moodle.org. Web service API functions. [https://docs.moodle.org/dev/Web\\_service\\_API\\_functions](https://docs.moodle.org/dev/Web_service_API_functions). [Accedido 11-Nov-2021].
- [46] neolms.com. neolms. [https://www.neolms.com/info/neo\\_vs\\_schoolology#:~:text=We%20have%20heard%20anecdotal%20that,contract%20for%20the%20Enterprise%20plan](https://www.neolms.com/info/neo_vs_schoolology#:~:text=We%20have%20heard%20anecdotal%20that,contract%20for%20the%20Enterprise%20plan). [Accedido 11-Nov-2021].
- [47] Mozilla Developers Network. Lazy loading. [https://developer.mozilla.org/en-US/docs/Web/Performance/Lazy\\_loading](https://developer.mozilla.org/en-US/docs/Web/Performance/Lazy_loading). [Accedido 18-Jul-2021].
- [48] NodeJS. NodeJS. <https://nodejs.org/es/>. [Accedido 23-May-2021].
- [49] Noodle. 32 innovative online tools to use in 2015. <https://web.archive.org/web/20171115085513/https://www.noodle.com/articles/32-innovative-online-tools-to-use-in-2015>. [Accedido 17-Jun-2021 a través de <https://web.archive.org/> dado que actualmente no es accesible].
- [50] noodle.com. About noodle. <https://partners.noodle.com/about-noodle>. [Accedido 10-Nov-2021].
- [51] Phil on ed tech. State of Higher Ed LMS Market for US and Canada: Year-End 2020 Edition. <https://philonedtech.com/state-of-higher-ed-lms-market-for-us-and-canada-year-end-2020-edition/>. [Accedido 23-Jul-2021].
- [52] Oracle. Parallelism. [https://docs.oracle.com/javase/tutorial/collections/streams/parallelism.html#executing\\_streams\\_in\\_parallel](https://docs.oracle.com/javase/tutorial/collections/streams/parallelism.html#executing_streams_in_parallel). [Accedido 24-Jul-2021].
- [53] packtpub.com. An overview of web services in Sakai. <https://hub.packtpub.com/overview-web-services-sakai/>. [Accedido 11-Nov-2021].
- [54] programmableweb. Edmodo REST API. <https://www.programmableweb.com/api/edmodo-rest-api>. [Accedido 15-Dic-2021].
- [55] Confluence Sakai Project. How to use the Sakai Web Services. <https://confluence.sakaiproject.org/display/WEBSVCS/WS-Java>. [Accedido 15-Dic-2021].
- [56] Sakai Project. GitHub Sakai Project. <https://github.com/sakaiproject/sakai>. [Accedido 11-Nov-2021].
- [57] Sakai Project. Sakai releases. <http://source.sakaiproject.org/release/21.1/>. [Accedido 11-Nov-2021].
- [58] FEDERAL INFORMATION PROCESSING STANDARDS PUBLICATION. Secure Hash Standard (SHS). <https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf>. [Accedido 11-Nov-2021].

- [59] RedHat. Wildfly downloads. <https://www.wildfly.org/downloads/>. [Accedido 17-Ago-2021].
- [60] Ramiro Rodriguez. Codigo fuente del presente proyecto. <https://gitlab.fing.edu.uy/matefun/Frontend/-/tree/matefun-moodle-integration>.
- [61] Ramiro Rodriguez. Codigo fuente del presente proyecto. <https://gitlab.fing.edu.uy/matefun/Frontend/-/tree/master>.
- [62] Ramiro Rodriguez. Documentacion del presente proyecto de grado. <https://gitlab.fing.edu.uy/matefun/Frontend/-/wikis/Interoperabilidad-de-Matefun-con-Moodle>.
- [63] Ramiro Rodriguez. Interoperabilidad Matefun-Moodle, Decisiones de diseño. [https://gitlab.fing.edu.uy/matefun/Frontend/-/wikis/uploads/d4a144eb48347f3a4535bf30489e7ba9/\\_Matefun-Moodle\\_Decisiones-de-dise%C3%B1o.pdf](https://gitlab.fing.edu.uy/matefun/Frontend/-/wikis/uploads/d4a144eb48347f3a4535bf30489e7ba9/_Matefun-Moodle_Decisiones-de-dise%C3%B1o.pdf). [Accedido 02-Jul-2021].
- [64] Ramiro Rodriguez. Interoperabilidad Matefun-Moodle, Estado del arte. [https://gitlab.fing.edu.uy/matefun/Frontend/-/wikis/uploads/3422eb89c837ab0269818078b443d26a/Estado\\_del\\_arte.pdf](https://gitlab.fing.edu.uy/matefun/Frontend/-/wikis/uploads/3422eb89c837ab0269818078b443d26a/Estado_del_arte.pdf). [Accedido 02-Jul-2021].
- [65] Ramiro Rodriguez. Interoperabilidad Matefun-Moodle, Manual de configuracion de Moodle. [https://gitlab.fing.edu.uy/matefun/Frontend/-/wikis/uploads/8ffc172e471fa5512073c0f9694fc63e/configuracion\\_Moodle.pdf](https://gitlab.fing.edu.uy/matefun/Frontend/-/wikis/uploads/8ffc172e471fa5512073c0f9694fc63e/configuracion_Moodle.pdf). [Accedido 02-Jul-2021].
- [66] Ramiro Rodriguez. Interoperabilidad Matefun-Moodle, Manual de transferencia. [https://gitlab.fing.edu.uy/matefun/Frontend/-/wikis/uploads/e271b9e43db31b5fc80b49074d0aa21a/Informe\\_de\\_transferencia.pdf](https://gitlab.fing.edu.uy/matefun/Frontend/-/wikis/uploads/e271b9e43db31b5fc80b49074d0aa21a/Informe_de_transferencia.pdf). [Accedido 02-Jul-2021].
- [67] Ramiro Rodriguez. Interoperabilidad Matefun-Moodle, Manual de usuario. [https://gitlab.fing.edu.uy/matefun/Frontend/-/wikis/uploads/32172d9fda3475b3a317f9fc41c5c5ee/Matefun-Moodle-manual\\_de\\_usuario.pdf](https://gitlab.fing.edu.uy/matefun/Frontend/-/wikis/uploads/32172d9fda3475b3a317f9fc41c5c5ee/Matefun-Moodle-manual_de_usuario.pdf). [Accedido 02-Jul-2021].
- [68] Ramiro Rodriguez. Matefun - Script deploy AWS para Unix. <https://gitlab.fing.edu.uy/matefun/Frontend/-/blob/matefun-moodle-integration/scripts/deployAWS.sh>. [Accedido 11-Nov-2021].
- [69] Ramiro Rodriguez. Matefun - Script deploy AWS para Windows. <https://gitlab.fing.edu.uy/matefun/Frontend/-/blob/matefun-moodle-integration/scripts/deployAWS-Win.bat>. [Accedido 11-Nov-2021].
- [70] sakai project. Try Sakai. <https://www.sakailms.org/try-sakai-lms-free>. [Accedido 11-Nov-2021].

## Referencias

- [71] Schoology. API Documentation. <https://developers.schoology.com/api>. [Accedido 19-Ago-2021].
- [72] Schoology. Schoology terms of use. <https://www.schoology.com/terms-of-use/es>. [Accedido 11-Nov-2021].
- [73] softwaretestinghelp.com. What Is Regression Testing? <https://www.softwaretestinghelp.com/regression-testing-tools-and-methods/>. [Accedido 11-Nov-2021].
- [74] Open source initiative. Educational Community License, Version 2.0. <https://opensource.org/licenses/ec12.php>. [Accedido 11-Nov-2021].
- [75] W3techs surveys. Usage statistics of javascript as client-side programming language on websites. <https://w3techs.com/technologies/details/cp-javascript/>. [Accedido 10-Nov-2021].
- [76] W3techs surveys. Usage statistics of javascript libraries for websites. [https://w3techs.com/technologies/overview/javascript\\_library](https://w3techs.com/technologies/overview/javascript_library). [Accedido 10-Nov-2021].
- [77] Lafayette technology help. Managing Folders in Moodle. <https://help.lafayette.edu/allowing-students-to-upload-and-share-files-in-moodle/>. [Accedido 18-Ago-2021].
- [78] tim st.clair. Block Public files. [https://moodle.org/plugins/block\\_public\\_files](https://moodle.org/plugins/block_public_files). [Accedido 19-Ago-2021].
- [79] Typescript. Typescript download. <https://www.typescriptlang.org/download>. [Accedido 17-Ago-2021].
- [80] Typescript. TypeScript for the New Programmer. <https://www.typescriptlang.org/docs/handbook/typescript-from-scratch.html>. [Accedido 10-Nov-2021].
- [81] Programacion4 Udelar. Criterios de Asignación de Responsabilidades GRASP. [https://eva.fing.edu.uy/pluginfile.php/55264/mod\\_page/content/69/11%20-%20disenio%20-%20grasp.pdf](https://eva.fing.edu.uy/pluginfile.php/55264/mod_page/content/69/11%20-%20disenio%20-%20grasp.pdf). [Accedido 23-Jul-2021].