



UNIVERSIDAD NACIONAL DE EDUCACIÓN A DISTANCIA

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA

Proyecto de Fin de Grado en Ingeniería Informática

## **Análisis del aprendizaje y evaluación en plataformas en entornos Big Data y servicios en la nube**

Nombre alumno: Sergio Fraile Pérez

Dirigido por: Rafael Pastor Vargas

Codirector: Antonio Robles Gómez

Curso: 2022-2023 Junio





**Análisis del aprendizaje y evaluación en plataformas en entornos Big Data y servicios en la nube**

**Proyecto de Fin de Grado en Ingeniería Informática  
de modalidad específica**

Realizado por: Sergio Fraile Pérez

Dirigido por: Rafael Pastor Vargas

Codirector: Antonio Robles Gómez

Fecha de lectura y defensa: XX de Junio de 2023



## Agradecimientos

Quiero aprovechar esta sección para dar las gracias a todas las personas que me han ayudado y apoyado durante estos duros años de estudio.

En primer lugar y más importante agradecer a mi familia, en especial a mi mujer Eva y mi hija Alba, su apoyo incondicional, su tiempo, su paciencia y su comprensión sin los cuales hubiera sido imposible la superación de este Grado.

También quiero expresar mi agradecimiento a mis compañeros de trabajo de la Academia de Infantería de Toledo, que al igual que mi familia me han provisto de un apoyo sin el cual no habría tenido nunca el tiempo suficiente de dedicación al estudio.

Recalcar también la ayuda y enseñanzas obtenidas de los diferentes docentes en las muchas asignaturas de este grado.

Por último, dar las gracias a mi director y codirector por sus directrices, supervisión, consejos y sugerencias, imprescindibles para la finalización del TFG.



## Resumen

Este proyecto se centra en el Learning Analytics, una disciplina cuyo objetivo es mejorar el aprendizaje y la enseñanza a través del análisis de los datos obtenidos a partir de los estudiantes, contenidos y procesos de enseñanza, identificando patrones que permiten medir el rendimiento del estudiante, así como la detección de problemas derivados de los recursos educativos y proponer soluciones o decisiones que mejoren la capacidad de la enseñanza.

Por ello el objetivo general del presente proyecto será mejorar la actuación docente mediante el análisis de los datos recopilados, proporcionando información útil para la modificación de recursos y/o mejorar la individualidad del aprendizaje.

Para cumplir con el objetivo propuesto se desarrolla una arquitectura que posibilita la recopilación, almacenamiento y análisis de datos obtenidos de plataformas de aprendizaje interactivo mediante la implementación de todos los subsistemas necesarios. La integración de dichos subsistemas se llevará a cabo mediante una tecnología de virtualización ligera y portable, utilizando contenedores Docker. Dicha tecnología permitirá desplegar el sistema rápida y eficientemente en cualquier equipo.

En referencia a la estructura de los datos, se utilizará el estándar xAPI. Estándar con mayor flexibilidad y claridad, con una disposición similar al lenguaje humano, facilitando el proceso de análisis. La fase de análisis se llevará a cabo mediante funciones estadísticas y un algoritmo de inteligencia artificial.

La motivación de este proyecto deriva de la posibilidad de mejorar el aprendizaje solucionando problemas que mejoren la calidad de la enseñanza y a su vez trabajar con diferentes tecnologías como servidores, bases de datos, inteligencia artificial, páginas web....

## Palabras clave

xAPI, LRS, LMS, Docker, cliente, servidor



## Abstract

This project focuses on Learning Analytics, a discipline whose objective is to improve learning and teaching through the analysis of data obtained from students, content and teaching processes, identifying patterns that allow measuring student performance, as well as the detection of problems derived from educational resources and propose solutions or decisions that improve the teaching capacity.

For this reason, the general objective of this project will be to improve teaching performance through the analysis of the collected data, providing useful information for the modification of resources and/or improving the individuality of learning.

To meet the proposed objective, an architecture is developed that enables the collection, storage and analysis of data obtained from interactive learning platforms through the implementation of all the necessary subsystems. The integration of these subsystems will be carried out through a lightweight and portable virtualization technology, using Docker containers. This technology will allow the system to be deployed quickly and efficiently on any computer.

In reference to the data structure, the xAPI standard will be used. Standard with greater flexibility and clarity, with a layout similar to human language, facilitating the analysis process. The analysis phase will be carried out using statistical functions and an artificial intelligence algorithm.

The motivation of this project derives from the possibility of improving learning by solving problems that improve the quality of teaching and in turn working with different technologies such as servers, databases, artificial intelligence, web pages...

## Keywords

xAPI, LRS, LMS, Docker, client, server



# Índice

Agradecimientos .....	5
Resumen.....	7
Palabras clave.....	7
Abstract .....	9
Keywords.....	9
Índice de Tablas.....	15
Índice de Figuras .....	17
1. Introducción, Objetivos y Estructura.....	19
1.1 Introducción .....	19
1.2 Objetivos .....	21
1.2.1 Objetivo General .....	21
1.2.2 Objetivos específicos.....	22
1.2.3 Motivación.....	22
1.3 Planificación y presupuesto .....	23
1.3.1 Planificación .....	23
1.3.2 Presupuesto.....	24
1.4 Estructura .....	25
2. Estado del arte .....	27
2.1 Introducción .....	27
2.2 Estado del arte .....	27
2.3 Uso actual.....	31
3. Análisis.....	33
3.1 Modelo de dominio.....	33
3.2 Casos de uso.....	34
3.3 Requisitos y elección de los componentes .....	36
3.3.1 NGINX .....	37
3.3.2 LRS.....	37
3.3.3 Base de datos .....	37
3.3.4 Caché de datos .....	37
3.3.5 Módulo de análisis .....	38
3.3.6 Docker .....	38
4. Metodología.....	41

4.1 Recopilación y almacenamiento de datos.....	41
4.1.1 Descripción y proceso .....	42
4.2 Análisis y visualización de los resultados .....	43
5. Propuesta de arquitectura y diseño .....	45
5.1 Introducción .....	45
5.2 Servidor .....	46
5.2 LRS .....	47
5.4 Módulo de análisis .....	48
6. Implementación .....	51
6.1 LRS .....	51
6.2 Módulo Analytics.....	54
6.2.1 Declaraciones .....	55
6.2.2 Alumnos.....	56
6.2.3 Análisis.....	56
6.3 Algoritmo de Regresión Lineal .....	61
6.4 Arquitectura final .....	64
6.5 Conclusiones.....	65
7. Pruebas.....	67
7.1 Introducción .....	67
7.2 Pruebas Unitarias .....	67
7.2.1 Nginx.....	67
7.2.2 LRS .....	69
7.2.3 Módulo de Análisis .....	69
7.3 Pruebas de integración .....	70
7.4 Pruebas de validación .....	70
7.5 Conclusiones.....	71
8. Conclusiones y trabajos futuros .....	73
9.1 Conclusiones.....	73
9.2 Trabajos futuros .....	74
Bibliografía .....	77
Glosario .....	79
Anexos.....	83
I. Manual de la aplicación.....	83
I.I Requisitos de LMS para su uso y compatibilidad con el módulo de análisis .....	83

I.II Requisitos del sistema .....	83
I.III Creación de las imágenes .....	84
I.IV Estructura del LRS .....	86
I.V Módulo Analytics .....	86
II. Código Fuente .....	91



## Índice de Tablas

Tabla 1. Cronograma .....	24
Tabla 2. Presupuesto .....	25
Tabla 3. Tabla de Declaraciones .....	56
Tabla 4. Tabla de Alumnos .....	56
Tabla 5. Histórico del Análisis General .....	57
Tabla 6. Tabla de Sentencias analizables .....	57
Tabla 7. Resultados Análisis General.....	58
Tabla 8. Histórico del Análisis Individual .....	58
Tabla 9. Selección de alumno.....	59
Tabla 10. Tabla binaria del alumno seleccionado .....	59
Tabla 11. Tabla binaria de los alumnos con nota.....	60
Tabla 12. Resultado del Análisis Individual .....	60
Tabla 13. Porcentaje de actividades realizadas .....	60
Tabla 14. Conjunto de entrenamiento del Algoritmo .....	63



## Índice de Figuras

Figura 1. Pasos del LA.....	21
Figura 2. Esquema del funcionamiento de un LRS.....	31
Figura 3. Modelo de dominio.....	34
Figura 4. Modelo de casos de uso del LRS.....	35
Figura 5. Modelo de casos de uso del módulo Analytics.....	36
Figura 6. Recopilación y almacenamiento de datos.....	42
Figura 7. Análisis de datos.....	43
Figura 8. Modelo cliente-servidor.....	45
Figura 9. Diseño del servidor.....	46
Figura 10. Diseño del LRS.....	47
Figura 11. Diseño del módulo Analytics.....	49
Figura 12. Contenedores del LRS.....	53
Figura 13. Opciones del curso.....	55
Figura 14. Gráfica del coste de la función.....	60
Figura 15. Esquema del algoritmo de Regresión Lineal.....	61
Figura 16. Arquitectura completa.....	65
Figura 17. Prueba de carga del servidor 250 usuarios.....	68
Figura 18. Prueba de carga del servidor 500 usuarios.....	69
Figura 19. Subcarpetas del proyecto.....	84
Figura 20. Contenedores activos.....	85
Figura 21. Página de inicio del módulo de análisis.....	87
Figura 22. Vista del usuario.....	87
Figura 23. Vista de los cursos del usuario.....	88
Figura 24. Vista de los tipos de análisis.....	88
Figura 25. Vista de la selección de alumnos para el análisis individual.....	89
Figura 26. Vista de la selección del rango de fechas en el análisis general.....	90
Figura 27. Vista del resultado del análisis general.....	90
Figura 28. Vista del resultado del análisis individual.....	91
Figura 29. Estructura del módulo de análisis.....	92
Figura 30. Estructura de las carpetas de servicios.....	93



## 1. Introducción, Objetivos y Estructura

### 1.1 Introducción

El mundo de la educación está en constante transformación evolutiva, impulsando la creatividad e incorporando las nuevas tecnologías en todos los ámbitos educativos, utilizando el análisis de datos para lograr la personalización del aprendizaje mediante la creación de un modelo que refleje el grado de aprovechamiento del aprendizaje y la visualización de los posibles problemas encontrados.

El uso de entornos virtuales de aprendizaje o plataformas de aprendizaje interactivo (Learning Manager System, LMS) (G.Almonte, 2021), como por ejemplo Moodle (Partners, 2022), ocasiona necesariamente una interacción entre el usuario y el sistema, esta interacción proporciona numerosa y valiosa información acerca de las actividades de los usuarios. Todos los datos recopilados constituyen el Big Data, en este caso, educacional. Dicha recopilación y el posterior análisis de esos datos forman lo que denominamos Learning Analytics (LA) (Phil Long, 2011). Según el experto en innovación pedagógica George Siemens (Siemens, 2013):

*“Learning Analytics es el uso de datos inteligentes, de datos producidos por los alumnos y de modelos de análisis, para descubrir información y conexiones sociales que permitan predecir y asesorar el aprendizaje de las personas”*

Gracias al uso del Big Data (Daniel Amo, 2017) podemos optimizar o mejorar el aprendizaje de los alumnos mediante el análisis, ofreciendo una mejor comprensión de los recursos educativos y recomendaciones para la mejora de la calidad educativa a los creadores de éstos. Así mismo nos permite la evaluación del progreso de los alumnos adaptando el contenido de los recursos según los resultados obtenidos.

El tipo de datos que conforman el Big Data se pueden clasificar en dos tipos:

- Cuantitativos: como por ejemplo el número de actividades que realiza un alumno o la nota de un examen.
- Cualitativos: como opiniones de usuarios o la dificultad de una actividad.

Una vez definidos los tipos de datos de los que podemos disponer debemos enumerar varias metodologías de análisis del LA atendiendo a un objetivo marcado o para satisfacer las necesidades que se presenten:

- Análisis de contenido: nos permite analizar el contenido utilizado por los alumnos y determinar su utilidad y uso, abriendo una vía para su posible adaptación.
- Análisis de patrones: gracias a este tipo de análisis podemos anticiparnos a las acciones y resultados de los usuarios posibilitando el cambio antes de un resultado no deseado.
- Análisis de rendimiento: mide el progreso de los estudiantes, detectando el posible éxito o fracaso académico.
- Análisis de tareas: evalúa a los alumnos en determinadas tareas, identificando habilidades individuales.
- Análisis del comportamiento: gracias a las interacciones con el sistema podremos determinar si el comportamiento será apto o no apto para un buen aprendizaje.
- Análisis de redes sociales: nos permiten obtener información del carácter, relaciones, etc., de los estudiantes.

A pesar de las posibilidades que nos brinda el LA, no debemos olvidar que no sustituye la labor docente ni la tutorial.

A continuación, nombramos los pasos necesarios de la aplicación del LA:

- La primera acción es plantear nuestro objetivo: debemos saber cuáles son las necesidades y/o problemas del aprendizaje de los alumnos y de los recursos educativos para establecer el tipo de datos que necesitamos a partir de los cuales realizar un análisis determinado que solucione la problemática surgida.
- A continuación, debemos recopilar los datos: en este paso debemos implementar un mecanismo o infraestructura que nos permita almacenar los datos obtenidos, además de establecer la estructura de estos.
- Seleccionar y preparar los datos necesarios: una vez almacenados todos los datos debemos ajustarlos a nuestras necesidades y objetivos establecidos, seleccionando solo los que nos aportaran información útil y desarrollar un sistema que nos permita su análisis.
- Establecer y ejecutar los métodos de análisis: seleccionados los datos pertinentes solo queda elegir el tipo de análisis adecuado, implementarlo de una manera eficiente y ejecutarlo.
- Presentar los resultados obtenidos en el paso anterior: los resultados se deben mostrar claros para una correcta interpretación por parte del docente.
- Establecer actuaciones a partir de los resultados obtenidos: el último paso es responsabilidad del docente; establecer unas pautas o soluciones para la mejora del aprendizaje y de los recursos educativos a partir de la información proporcionada.

En la Figura 1. Pasos del LA se representan los pasos comentados anteriormente:

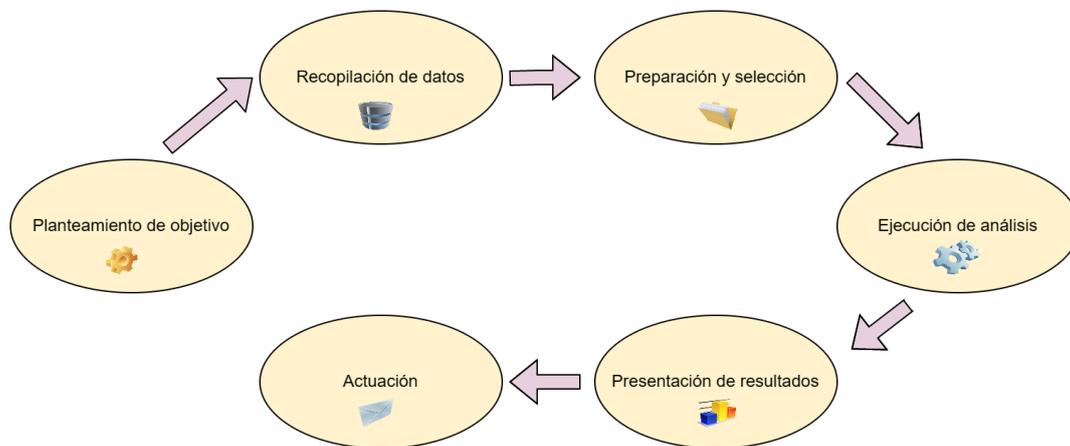


Figura 1. Pasos del LA

El uso progresivo de plataformas educativas nos permite extraer el rastro digital dejado por sus usuarios. Los cursos masivos online (MOOC) (Francisco José García-Peñalvo, 2017) son perfectos para el desarrollo del LA debido al seguimiento de un gran número de estudiantes que nos proporcionan una ingente cantidad de datos necesarios para un correcto y eficiente análisis, motivando el desarrollo de una arquitectura cliente-servidor sencilla, portable y escalable que nos permita ayudar o apoyar en la forma en que los educadores transmiten sus conocimientos al estudiante, creando una educación precisa e individualizada justificando así este proyecto.

## 1.2 Objetivos

### 1.2.1 Objetivo General

El objetivo general del proyecto es ayudar a los tutores o creadores de recursos educativos on-line a mejorar el aprendizaje de sus alumnos o usuarios mediante el análisis de toda la información recabada por medio de las interacciones realizadas con los LMSs, adelantándonos a los posibles problemas que puedan surgir en el aprendizaje y adaptando en consecuencia dichos recursos.

### 1.2.2 Objetivos específicos

Los objetivos específicos se basarían en la creación de una arquitectura del tipo cliente-servidor portable y escalable que nos permita:

- Almacenar y visualizar los datos obtenidos a través de la interacción de los usuarios con los LMSs. Para lograr un almacenamiento eficiente debemos estudiar el tipo y la estructura de los datos que se van a obtener eligiendo posteriormente la base de datos adecuada para dichos datos, que nos permita y facilite tanto su almacenamiento como su acceso y modificación, atendiendo a su vez a la escalabilidad y portabilidad de ésta. En cuanto a la visualización deberá ser lo más sencilla, clara y ordenada posible de manera que posibilite la comprensión de la información por parte del usuario.
- Analizar, en conjunto e individualmente los resultados obtenidos del progreso de los estudiantes implementando un sistema de análisis que haga posible conocer la evolución del aprendizaje tanto del grupo de estudiantes en general, así como de analizar la probabilidad de que un alumno pueda aprobar o suspender a partir de los datos obtenidos de anteriores alumnos.
- Ofrecer información que permita al tutor o docente dar soluciones a los problemas encontrados, de rendimiento del alumno, dificultad del curso, etc..., estableciendo una comunicación directa con los estudiantes y actualizando o solucionando los problemas detectados en los recursos educativos que ralentizan o empeoran el aprendizaje.

### 1.2.3 Motivación

El motivo del desarrollo del proyecto viene dado por la oportunidad de desarrollar y conocer nuevas habilidades técnicas con el uso de tecnologías de virtualización (Docker), algoritmos de inteligencia artificial (Regresión Lineal), creación de páginas web y el uso de diferentes lenguajes de programación (Python, JavaScript, HTML5, etc...), integrando todo este conocimiento en una única arquitectura capaz de lograr los objetivos propuestos. Así mismo también se presenta la ocasión de mejorar el aprendizaje aportando soluciones a los problemas detectados.

## 1.3 Planificación y presupuesto

### 1.3.1 Planificación

Atendiendo a la Tabla 1 que representa el cronograma semanal dispuesto para el proyecto, se distribuye el tiempo disponible entre las siguientes tareas:

- Definición de objetivos: tarea en la que se establecen los objetivos generales y específicos, detallados anteriormente. Identificando todos los recursos y conocimientos que serán necesarios para la consecución del proyecto. La duración de esta tarea no debe ser superior a 1 semana.
- Estudio del arte xAPI: una vez establecidos los objetivos, debemos aprender y obtener los conocimientos necesarios sobre el estándar xAPI como su estructura, funcionamiento, recursos, etc... La duración será de tres semanas.
- Análisis: fase que nos permitirá analizar y seleccionar los recursos establecidos en los objetivos con la finalidad de corroborar la idoneidad de estos. La fase de análisis tendrá una duración de dos semanas.
- Arquitectura y diseño: se desarrollará una arquitectura con un diseño que permita integrar todos los recursos analizados anteriormente, examinando compatibilidades y estableciendo criterios que determinen las estructuras de datos, lenguajes de programación, software, redes, comunicaciones entre servicios, escalabilidad, portabilidad.... La presente tarea tendrá una duración de 4 semanas.
- Implementación: obtenida y desarrollada la arquitectura y el diseño se procede a su implementación durante 5 semanas.
- Pruebas: se desarrollan las pruebas pertinentes que nos permitan comprobar que se han cumplido los objetivos establecidos. Se realizarán pruebas durante dos semanas.
- Finalización del proyecto: básicamente esta fase se centrará en el desarrollo y corrección de la documentación. Debería finalizar en una semana.

A continuación, en la Tabla 1. Cronograma, se representa gráficamente el desarrollo anterior:

ACTIVIDADES	CRONOGRAMA																	
	AÑO 2023																	
	FEBRERO				MARZO				ABRIL				MAYO				JUNIO	
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2
Definición de objetivos																		
Estudio del arte xAPI																		
Análisis																		
Arquitectura y Diseño																		
Implementación																		
Pruebas																		
Finalización del Proyecto																		

Tabla 1. Cronograma

### 1.3.2 Presupuesto

Al ser un proyecto que incluye múltiples características, siendo necesario el conocimiento de varias disciplinas como bases de datos, programación en distintos lenguajes (nodejs, Python, HTML5, Handlebars, BootStrap), tecnologías de virtualización, servidores, redes, inteligencia artificial, software específico (VSC, Docker, Postman) y desarrollo de páginas web, los costes serán elevados.

Atendiendo al cronograma de actividades, diferenciaremos entre horas de:

- Trabajos de estudio: durante los cuales se establecen los objetivos y se obtienen los conocimientos necesarios para la consecución del proyecto.
- Analista de sistemas informáticos: encargado de realizar el análisis del sistema óptimo que se adapte al objetivo del proyecto.
- Diseñador de arquitecturas de software: su misión será la de desarrollar la arquitectura del diseño a partir del análisis anterior.
- Desarrollador de software: codificará la arquitectura diseñada, en los lenguajes definidos.
- Administrador de sistemas y redes: implementará la red que contendrá todos los subsistemas, asegurándose de la conectividad de éstos.
- Programación: el trabajo de implementar las pruebas y probarlas será trabajo del técnico en programación, bajo supervisión del analista de sistemas.
- Administrativo: trabajo realizado por un técnico administrativo, será el encargado de redactar toda la documentación necesaria.

Seguidamente en la Tabla 2. Presupuesto, representamos la tabla del presupuesto.

ACTIVIDADES	TIPO	HORAS	COSTE/HORA	TOTAL
DEFINICIÓN DE OBJETIVOS	ESTUDIO	25	55€	<b>1375€</b>
ESTUDIO DEL ARTE xAPI	ESTUDIO	70	55€	<b>3850€</b>
ANÁLISIS	ANALISTA DE SISTEMAS	50	80€	<b>4000€</b>
ARQUITECTURA Y DISEÑO	DISEÑADOR DE ARQUITECTURAS	90	80€	<b>7200€</b>
IMPLEMENTACIÓN ESTRUCTURAL	DESARROLLADOR DE SOFTWARE	110	80€	<b>8800€</b>
IMPLEMENTACION DE REDES	ADMINISTRADOR DE SISTEMAS Y REDES	30	80€	<b>2400€</b>
PRUEBAS	PROGRAMACIÓN	50	70€	<b>3500€</b>
DOCUMENTACIÓN	ADMINISTRATIVO	25	60€	<b>1500€</b>
TOTAL, SIN IVA				<b>32625€</b>
TOTAL, CON IVA 21%				<b>39476.25€</b>

Tabla 2. Presupuesto

## 1.4 Estructura

A continuación, se describe la estructura de este documento enumerando los diferentes capítulos:

- En el Capítulo 2 Estado del arte se procede al estudio del estándar xAPI. Recabando información principalmente en sitios web, comenzando con una introducción detallando los orígenes de este estándar. Posteriormente se describe su estado actual, describiendo su estructura, ventajas y usos. Por último, se desarrolla y explica su uso actual.
- En el desarrollo del Capítulo 3 Análisis se establece un modelo de dominio que nos permite entender y visualizar todos los componentes que formarán parte de nuestro trabajo, realizando un análisis de los requisitos necesarios para la implementación de la arquitectura, complementado con los diagramas correspondientes del modelo de dominio y de los casos de uso. En este capítulo

- también se relaciona la elección de los componentes junto con su descripción y función.
- La creación de una metodología se explicará en el Capítulo 4 Metodología. En este capítulo se explica cómo se realiza la división del proyecto en fases, detallando la descripción de cada una y su proceso.
  - Una vez establecidas las fases del proyecto, en el Capítulo 5 Propuesta de arquitectura y diseño se describe el tipo de arquitectura que se utilizará, marcando los componentes de esta e identificando las relaciones y funciones entre ellos. Las explicaciones anteriores se complementan con figuras para una mejor comprensión.
  - El Capítulo 6 Implementación detalla cómo se realiza la creación de cada servicio y su puesta en funcionamiento, también se describen como serán las vistas del módulo de análisis y el tipo de datos que se mostrarán al usuario. Se incluyen diagramas y tablas para mejorar su entendimiento. Al final del capítulo se intenta dar una idea de la implementación del algoritmo de inteligencia artificial utilizado, acompañado de fórmulas y gráficos para su comprensión.
  - El detalle sobre la realización de pruebas unitarias de cada componente, de integración de los diferentes subsistemas y pruebas de validación se dan en el Capítulo 7 Pruebas, donde se ofrece una idea sobre los distintos tipos de pruebas realizadas y su finalidad. Se incluyen gráficas de rendimiento del software utilizado.
  - En el Capítulo 8 Conclusiones y trabajos futuros se intenta dar una idea global sobre los resultados del proyecto e incidir en posibles trabajos futuros que nos permitan mejorar el sistema desarrollado y dar ideas para la implementación de nuevos sistemas.

## 2. Estado del arte

### 2.1 Introducción

El estándar **xAPI** (LLC, 2023) apareció como iniciativa **ADL** (Advanced Distributed Learning) (Government, 2023) permitiendo el intercambio de datos sobre el desempeño y comportamiento de los alumnos, actualizando y mejorando el anterior sistema **SCORM** (Sharable Content Object Reference Model) (Chenoweth, 2017) más limitado en el acceso y recopilación de datos, debido a que se centraba en una educación con contenido cerrado y estático. Hoy en día el contenido ha evolucionado con las nuevas tecnologías dando el salto a espacios abiertos y masivos con contenido dinámico y distribuido por internet.

En 2011 la empresa Rustici Software recibió un contrato de ADL para mejorar el sistema SCORM, el resultado fue la creación del proyecto **Tin Can** y los primeros diseños xAPI. A finales de 2020 xAPI es estandarizado formalmente por el Instituto de Ingenieros Eléctricos y Electrónicos (IEEE) (Electrónicos, 2018) promoviendo su aceptación en todo el mundo.

Este sistema no solo funciona para entornos educativos, sino que puede usarse en cualquier entorno que disponga de un proveedor de actividades como por ejemplo una red social, un dispositivo médico, aplicaciones móviles, etc....

### 2.2 Estado del arte

Experience API o xAPI es una especificación que nos permite recopilar información procedente de la interacción del usuario con el sistema utilizando un vocabulario simple, similar en su estructura al lenguaje humano.

Especificaciones anteriores, como la ya comentada SCORM, establecían muchas restricciones que con el progreso tecnológico no nos permitían capturar todos los datos que pueden generarse, ya que no solo se obtienen datos de cursos tradicionales, sino que ahora podemos obtenerlos de simulaciones, mundos virtuales, aprendizaje móvil y redes sociales entre otros.

Las conclusiones de los análisis de varios estudios científicos nos permiten corroborar la utilidad y versatilidad de xAPI en todos los campos o situaciones donde se produzca una interacción entre un usuario y un sistema, y no solo en el campo educativo. Por ejemplo, en el área de los videojuegos (Cristina Alonso Fernandez, 2022)

donde se demuestra la capacidad del Learning Analytics combinado con xAPI para recopilar información más clara de los jugadores con el objetivo de mejorar los videojuegos y a la vez evaluar a los jugadores. Otro campo interesante es el de la realidad aumentada (Sommerauer, 2021) donde se demuestra la posibilidad de una evaluación precisa en la medición de la efectividad del aprendizaje en el entrenamiento de la realidad virtual, y por supuesto también queda demostrada su utilidad en el campo de la educación con diversos estudios (Carmen Fernandez-Morante, 2021).

### ¿Cómo funciona xAPI?

El funcionamiento de xAPI se basa en la recopilación de los datos o rastro digital que producen las interacciones de los usuarios con el sistema. Todos estos datos son enviados a un almacén de registros de aprendizaje o LRS (Learning Record Store), además un LRS puede compartir las declaraciones almacenadas con otros LRSs y existir como un sistema aislado o incluido dentro de otro sistema como un LMS. La estructura de los datos está determinada por expresiones de la forma:

#### ACTOR – VERBO – OBJETO

- **ACTOR:** persona que ejecuta la acción, en nuestro caso el estudiante.
- **VERBO:** describe el tipo de acción.
- **OBJETO:** define sobre qué se realiza la acción.

Un ejemplo de declaración sería: **Juan comienza el curso**

xAPI nos aporta una serie de ventajas:

- Nos permite registrar cualquier actividad que podamos describir con el lenguaje humano de una manera simple y entendible.
- Es posible compartir la experiencia del aprendizaje entre distintos LRSs.
- Nos proporciona la libertad de enviar declaraciones desde cualquier dispositivo que se encuentre en una conexión a internet o no, enviando las declaraciones cuando se disponga de ella.
- El seguimiento del aprendizaje de un alumno no depende únicamente de un LMS, pudiendo almacenar información del mismo alumno que utiliza diferentes recursos o dispositivos.

Las declaraciones pueden complementarse con campos opcionales (Downes, 2016) como el contexto donde realiza el actor la acción o el resultado de esta.

Algunos ejemplos de la capacidad del rastreo y almacenamiento de datos pueden ser:

- Ver, pausar o reiniciar un video.

- Descargar algún contenido.
- Iniciar y finalizar sesión.
- Leer un libro.
- Medir el tiempo de conexión.
- Puntuación de pruebas.

Las sentencias se estructuran en formato JSON (JavaScript Object Notation), formato de texto pensado para el intercambio de datos. A continuación, se describe la estructura de una sentencia xAPI:

**Actor:**

```
{ "NAME": "SALLY GLIDER", "NAME": "SALLY GLIDER",  
  "MBOX": "MAILTO:SALLY@EXAMPLE.COM" "MBOX": "MAILTO:SALLY@EXAMPLE.COM" }
```

Aquí se nos especifica el nombre de la persona que realiza una acción y su e-mail por el que podremos distinguirla de cualquier otra.

**Verbo:**

```
{ "ID": "HTTP://ADLNET.GOV/EXPAPI/VERBS/EXPERIENCED",  
  "DISPLAY": { "EN-US": "EXPERIENCED" "EN-US": "EXPERIENCED" } }
```

Los verbos se almacenarán y agregarán en un registro, constanding cada uno de ellos de un URI (identificador de recursos uniforme) que nos permite identificarlo. En el ejemplo anterior, el campo "id" nos permite identificar y encontrar el verbo requerido.

**Objeto:**

```
{ "ID": "HTTP://EXAMPLE.COM/ACTIVITIES/SOLO-HANG-GLIDING",  
  "DEFINITION": { "NAME": { "EN-US": "SOLO HANG GLIDING" } } }
```

Generalmente los objetos serán actividades, como en el caso anterior almacenadas en un registro. Todas las actividades estarán definidas por una única URI y podrán incluir una descripción de la actividad. Cada actividad tendrá su propia "id".

**Ejemplo de contexto:**

```
"CONTEXT": { "EXTENSIONS": { "HTTP://EXAMPLE.COM/WEATHERCONDITIONS": "RAINY" }
```

Se utilizará para añadir información de interés. También se podrán incluir otros campos como por ejemplo el resultado de una acción.

#### Sentencia completa:

```
{"ACTOR": {"NAME": "SALLY",  
  "MBOX": "MAILTO:SALLY@EXAMPLE.COM"},  
"VERB": {"ID": "HTTP://ADLNET.GOV/EXPAPI/VERBS/COMPLETED",  
  "DISPLAY": {"EN-US": "COMPLETED"}},  
"OBJECT": {"ID": "HTTP://EXAMPLE.COM/ACTIVITIES/SOLO-HANG-GLIDING",  
  "DEFINITION": {"TYPE": "HTTP://ADLNET.GOV/EXPAPI/ACTIVITIES/ASSESSMENT",  
    "NAME": {"EN-US": "SOLO HANG GLIDING"},  
    "EXTENSIONS": {"HTTP://EXAMPLE.COM/GLIDERCLUBID": "TEST-435"}},  
  "RESULT": {"COMPLETION": TRUE,  
    "SUCCESS": TRUE,  
    "EXTENSIONS": {"HTTP://EXAMPLE.COM/FLIGHT/AVERAGEPITCH": 0.05}},  
  "CONTEXT": {"EXTENSIONS": {"HTTP://EXAMPLE.COM/WEATHERCONDITIONS": "RAINY"}}
```

#### **¿Qué es un LRS?**

Básicamente un LRS (LLC, 2023) es un contenedor de datos, corazón de cualquier ecosistema xAPI, que sirve como repositorio para sentencias definidas en actividades ya comentadas anteriormente con el estándar xAPI.

Una de sus funciones es la de mantener todos los datos de manera coherente, aunque éstos se incrementen de forma masiva, permitiendo a su vez compartir dichos datos con otros LRSs.

#### **¿Cómo funciona?**

El LRS recibe las sentencias en formato xAPI vía web (HTTP o HTTPS), almacenándolas convenientemente. Los LRSs pueden coexistir con LMS mediante plugins desarrollados por terceros o individualmente como una base de datos. En la Figura 2. Esquema del funcionamiento de un LRS podemos visualizar el funcionamiento mencionado.

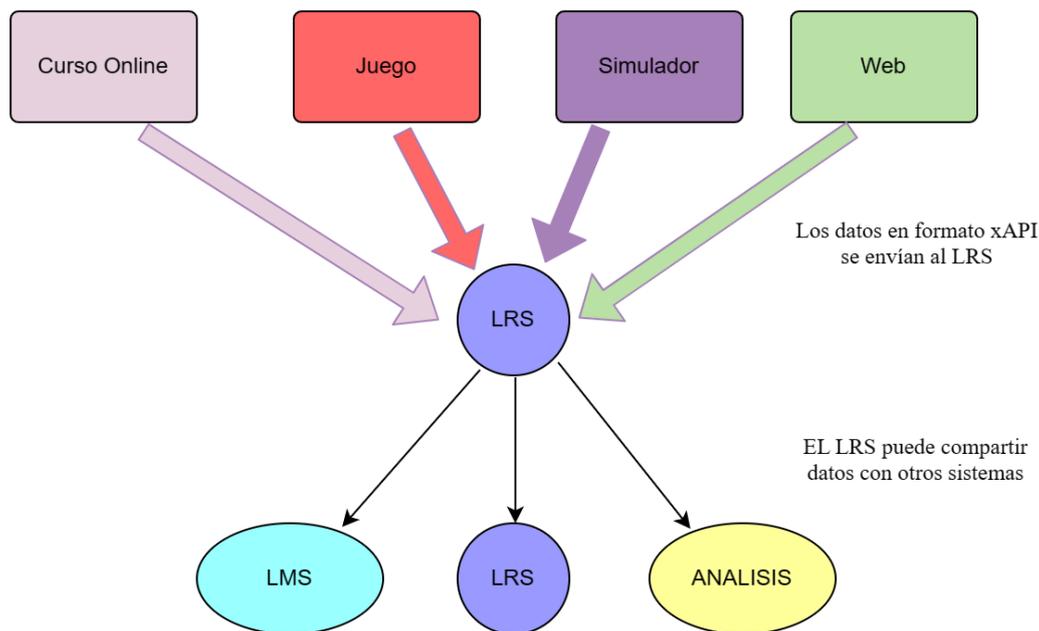


Figura 2. Esquema del funcionamiento de un LRS

### 2.3 Uso actual

En la actualidad el estándar xAPI es adoptado por múltiples organizaciones e instituciones como por ejemplo la universidad de Harvard y el departamento de defensa de Estados Unidos con la finalidad de encontrar patrones de comportamiento, seguir la actividad de los usuarios además de para la creación de sistemas de aprendizaje interactivo más adecuados al aprendizaje individual de sus usuarios.

Podemos encontrar diversa bibliografía (Amo Filv, 2021), donde se nos muestran algunos ejemplos bastante interesantes del uso de esta tecnologa como, por ejemplo y relacionado con el proyecto referenciado, la mejora de la docencia utilizando la anlisis de los datos de varios alumnos en la realizacin de unas pruebas tipo test. Dicho anlisis permitira, a travs de los resultados obtenidos, encontrar los problemas asociados al aprendizaje y por ello resolver dudas y modificar la forma en la que exponer el conocimiento.

Actualmente existen recursos gratuitos y de pago, tanto de LMSs como de LRSs que nos permiten explotar la anlisis de datos con fines docentes. Algunas plataformas

gratuitas como Moodle (Partners, 2022) nos permiten crear nuestros propios recursos educativos, proporcionando una enseñanza interactiva ajustada a nuestras preferencias, siendo compatible además con el estándar xAPI. En cuanto a los LRSs también podemos encontrar recursos gratuitos y de código abierto como LearningLocker (Pool, 2023) recurso que se explotará en este proyecto.

## 3. Análisis

### 3.1 Modelo de dominio

Para establecer un modelo de dominio necesitamos conocer exactamente el tipo de usuarios que harán uso del sistema, los medios técnicos que lo formarán y establecer la integración y cometido de cada componente.

Primero analizamos y definimos a los usuarios del sistema. El tipo de usuarios que harán uso del sistema serán: un administrador y el tutor o creador del recurso educativo.

Seguidamente se analiza el tipo de componentes necesarios para el diseño de la arquitectura.

Se implementará un sistema con dos servicios diferenciados, uno será el LRS encargado de recibir y mostrar las sentencias xAPI del LMS y el otro será un módulo de analítica de datos de nombre Analytics que como su nombre indica, seleccionará, analizará y mostrará los resultados de dichos análisis.

Para la implementación de los sistemas comentados, en primer lugar, necesitamos un **servidor** que nos dirija el tráfico hacia los dos servicios, el LRS y el módulo Analytics.

Los datos recibidos del LMS necesitan ser almacenados por lo que otro componente necesario será una **base de datos**.

Para acelerar el acceso y recuperación de datos también se hará indispensable una **memoria caché**, encargada de acelerar las búsquedas en la base de datos.

En la Figura 3. Modelo de dominio se representa esquemáticamente el modelo de dominio, donde podemos observar todos los componentes que lo incluyen y su relación entre ellos.

Observamos como el servidor dirige el tráfico entre ambos servicios, mostrando las vistas (páginas web) correspondientes a cada servicio. Cada servidor podrá dirigir el tráfico entre 1 o más LRSs y módulos Analytics.

Cada LRS tiene acceso a una memoria caché, de la cual recupera datos para acelerar las búsquedas y en caso de no encontrarlos buscará en la base de datos.

El módulo Analytics accede a los datos almacenados en una base de datos y los analiza, generando un informe de análisis.

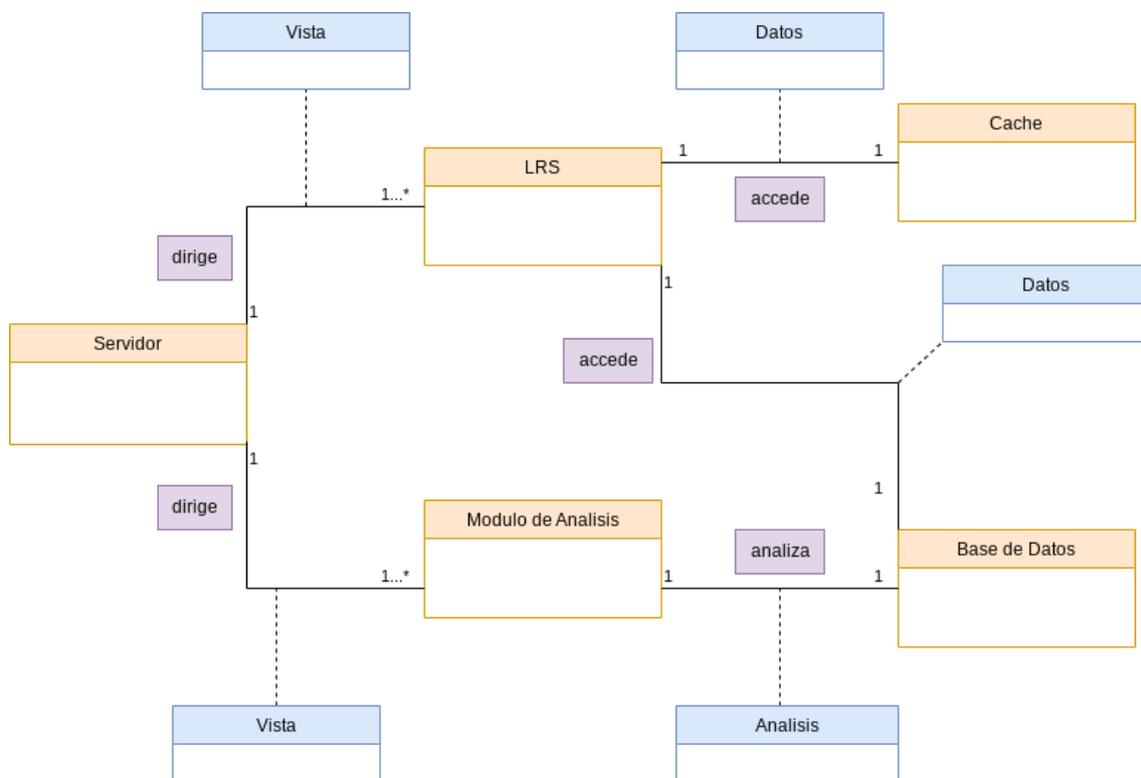


Figura 3. Modelo de dominio

### 3.2 Casos de uso

A continuación, detallamos los casos de uso de los usuarios con ambos servicios.

El **administrador del sistema** dispondrá de las siguientes funciones en el LRS:

- Configuración de todos los componentes.
- Iniciar, parar o reiniciar el sistema.
- Acceso a todos los subsistemas.
- Crear nuevos usuarios en el LRS.
- Tareas de mantenimiento y actualización.

El **tutor** podrá realizar las siguientes acciones:

- Acceder al LRS, iniciando sesión.
- Crear un cliente, que se corresponderá con cada curso.
- Crear un almacenamiento individual para cada cliente.

- Acceso a todas las sentencias recibidas por el LMS.

En la imagen Figura 4. Modelo de casos de uso del LRS se incluye el diagrama de casos de uso referente al LRS donde podemos observar los casos de uso de cada actor nombrados anteriormente:

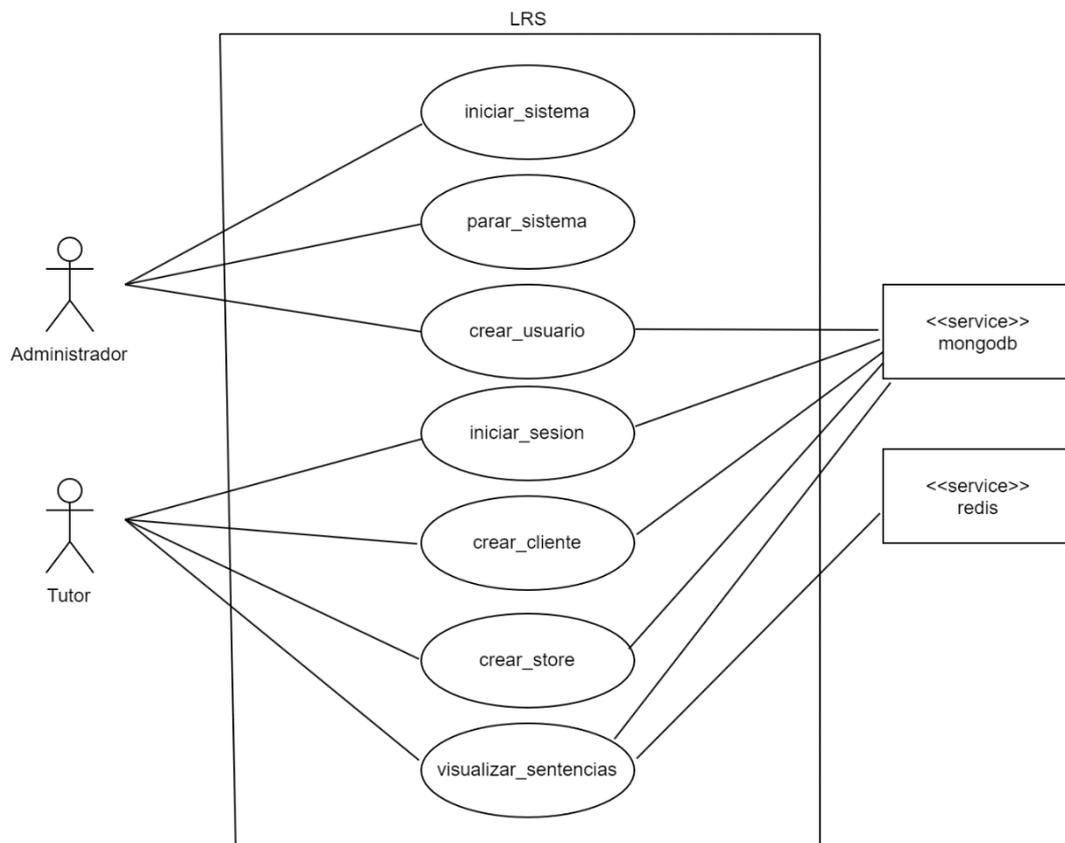


Figura 4. Modelo de casos de uso del LRS

En el módulo Analytics las funciones serían las siguientes:

El **administrador del sistema** dispondrá de las siguientes funciones:

- Únicamente se encargará del mantenimiento y actualización.

El **tutor** podrá realizar las siguientes acciones:

- Iniciar sesión.
- Podrá ver una lista de sus alumnos.

- Tendrá acceso a la visualización de las sentencias del curso.
- Ejecutar un análisis individual y/o general.

En la Figura 5. Modelo de casos de uso del módulo Analytics podemos observar el diagrama de casos de uso relacionado con el módulo Analytics y comentado anteriormente.

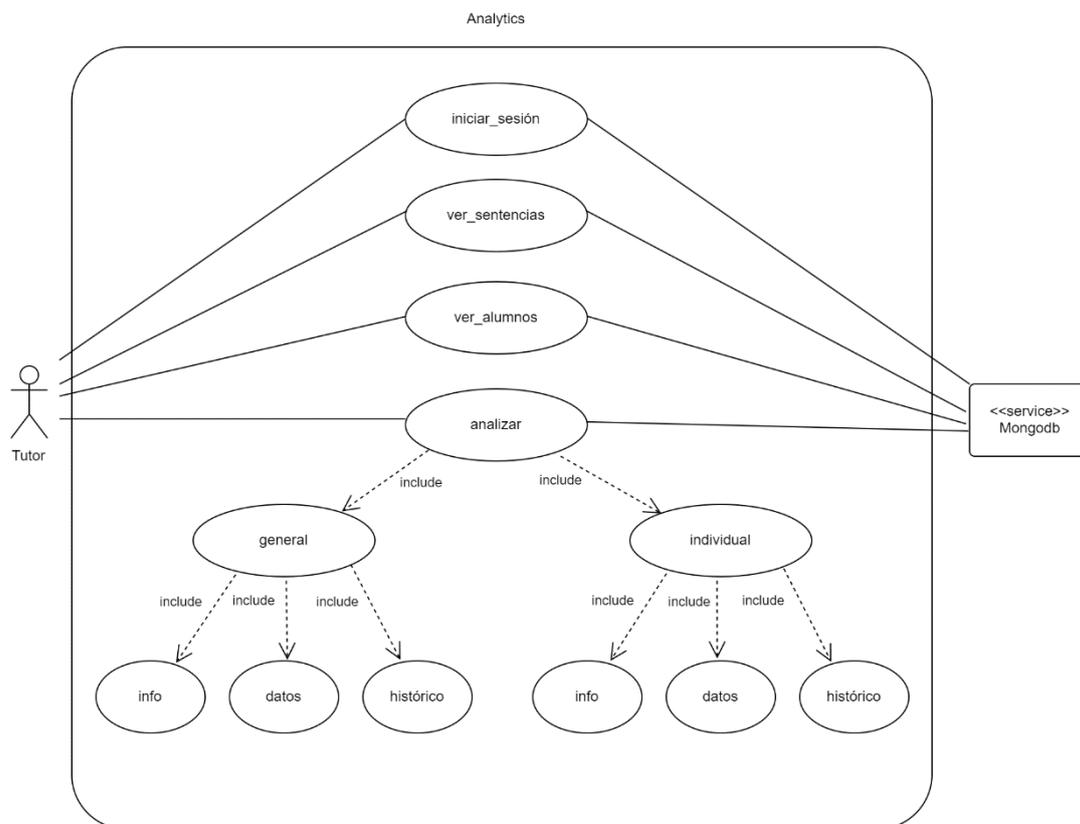


Figura 5. Modelo de casos de uso del módulo Analytics

### 3.3 Requisitos y elección de los componentes

Elegiremos los componentes de la arquitectura conforme a nuestras necesidades para que reúna los requisitos de sencillez, portabilidad y escalabilidad.

### 3.3.1 NGINX

Atendiendo a los sistemas citados en el modelo de dominio comenzaremos con la elección de un software de servidor web de código abierto y gratuito, **NGINX**.

Características de NGINX:

- Excelente capacidad de manejo en múltiples conexiones simultáneas
- Velocidad en el procesamiento de solicitudes.
- Posee una arquitectura asíncrona y controlada por eventos que nos permite, en una sola conexión de trabajo, atender 1024 solicitudes similares.
- Es altamente escalable.

Podemos encontrar toda la información referente a este software en su página oficial (Careers, 2023).

### 3.3.2 LRS

El siguiente sistema necesario es un LRS. Utilizaremos **LearningLocker** por ser gratuito y de código abierto. En su página oficial (Source, 2023) podemos encontrar toda la información referente a este recurso.

### 3.3.3 Base de datos

La elección de la Base de Datos debe ser acorde al modelo de datos que un LMS envía al LRS en nuestro caso a LearningLocker. Las sentencias xAPI se crearán en formato **JSON** por lo que **MongoDB** (base de datos orientada a documentos) se adaptará a nuestras necesidades. Esta base de datos facilita la escalabilidad horizontal, adaptando el sistema al aumento de información pudiendo trabajar con una gran cantidad de datos a gran velocidad.

### 3.3.4 Caché de datos

La caché de datos en memoria será utilizada por el LRS para acelerar las búsquedas de datos. **Redis** es un rápido almacén de datos clave-valor, en memoria, de código abierto. Ofrece un rápido rendimiento permitiendo millones de solicitudes por

segundo, con un tiempo de acceso a las bases de datos de milisegundos, disminuyendo la latencia de acceso a los datos y aliviando la carga que se produzca sobre la aplicación, así como sobre la base de datos.

### 3.3.5 Módulo de análisis

El módulo de análisis de datos se desarrollará en nuestro trabajo, y nos permitirá la elección, preparación y análisis de los datos obtenidos mediante el LRS, incluyendo un algoritmo de inteligencia artificial. **Se adaptará a un tipo concreto de LMS, que deberá constar de un conjunto de recursos educativos como videos, tareas, lecturas, etc..... y un examen final. Si la estructura del LMS cambiara, habría que modificar el módulo de análisis y adaptarlo a la nueva estructura.**

### 3.3.6 Docker

Conseguiremos la integración, interacción y la portabilidad de los múltiples sistemas anteriores mediante un mecanismo de virtualización ligero de sistemas denominado **Docker** o de contenedores.

Esta tecnología nos permitirá crear sistemas unitarios encapsulados denominados contenedores, funcionando de manera independiente, pero con la posibilidad de interactuar entre ellos. Cada sistema necesario en este trabajo se creará y optimizará en un contenedor independiente con acceso a los demás en el caso de que sea necesario.

Docker también nos permitirá el despliegue de nuestra arquitectura de forma rápida y eficiente, iniciando, reiniciando o parando todos los sistemas.

Podemos decir que Docker es una tecnología de virtualización ligera. Este sistema empaqueta el código y dependencias de una aplicación en una unidad denominada contenedor, independiente y estanca, dicha aplicación podrá ejecutarse en cualquier sistema reduciendo drásticamente los recursos utilizados.

Los sistemas de virtualización como Virtual-Vox o Hyper-V necesitan de una reinstalación completa de los sistemas operativos en cada virtualización con todo el uso de recursos que ello conlleva. Docker permite desplegar y ejecutar únicamente las aplicaciones que necesitamos sin necesidad de replicar sistemas operativos completos, elaborando una imagen de la aplicación o servicio y a partir de ella crear cuantos contenedores necesitemos. Por ejemplo, si necesitamos un servidor podemos

implementarlo en un contenedor y ejecutarlo en cualquier máquina independientemente del sistema operativo consiguiendo portabilidad en nuestra arquitectura. El único requisito será tener instalado el cliente Docker. En su página oficial (Docker, 2023) podemos encontrar toda la información referente a este sistema.

Además, encontramos diversa bibliografía como por ejemplo (GOUIGOUX, 2018), donde se nos explica paso a paso la creación de un sistema basado en microservicios con Docker, además de las características y funcionamiento de éste.



## 4. Metodología

Para facilitar el desarrollo del proyecto lo dividiremos en dos fases:

- Recopilación y almacenamiento de datos: En esta etapa se implementan todos los sistemas y características que nos permitirán la recopilación y comprobación de los datos enviados por el LMS para posteriormente almacenarlos en una base de datos.
- Análisis y visualización de los resultados: una vez construida la arquitectura que nos permite disponer de los datos se debe complementar con un módulo o sistema que posibilite el análisis y la visualización coherente de los resultados, permitiendo adoptar medidas que mejoren el aprendizaje general y/o individual, así como la modificación de los recursos educativos.

Cada fase requerirá de unos servicios o componentes concretos.

### 4.1 Recopilación y almacenamiento de datos

La implementación de la primera fase, recopilación y almacenamiento de datos compondrá la implementación de una arquitectura de microservicios, utilizando Docker, basada en contenedores que crearemos a partir de imágenes preconfiguradas para cumplir con nuestros objetivos.

Los servicios para desplegar serán:

- Servidor: NGINX.
- LearningLocker (LRS).
- Servicio xAPI.
- Base de datos: MONGODB.
- Caché de datos en memoria: REDIS.

En la práctica el servicio xAPI se implementa dentro de LearningLocker, en este proyecto se diseñará en contenedores diferentes como si fueran dos servicios diferenciados, aunque realmente sean un único servicio.

#### 4.1.1 Descripción y proceso

En esta fase el LMS enviará información de la interacción de los usuarios al LRS, que recibirá y verificará las sentencias xAPI. Dichas sentencias son enviadas y almacenadas en la base de datos.

El tutor tendrá acceso al LRS y por consiguiente a la visualización de las sentencias. También podrá crear el espacio para el almacenamiento de sentencias provenientes de distintos recursos.

En la Figura 6. Recopilación y almacenamiento de datos podemos observar gráficamente como los alumnos se conectan a través de cualquier dispositivo al LMS produciendo una interacción, a su vez este crea la sentencia xAPI en correlación a la actividad del usuario y la envía al LRS vía web. El LRS comprueba y verifica que la sentencia es correcta y la almacena en la base de datos. El tutor tendrá acceso a la visualización del historial almacenado del alumno.

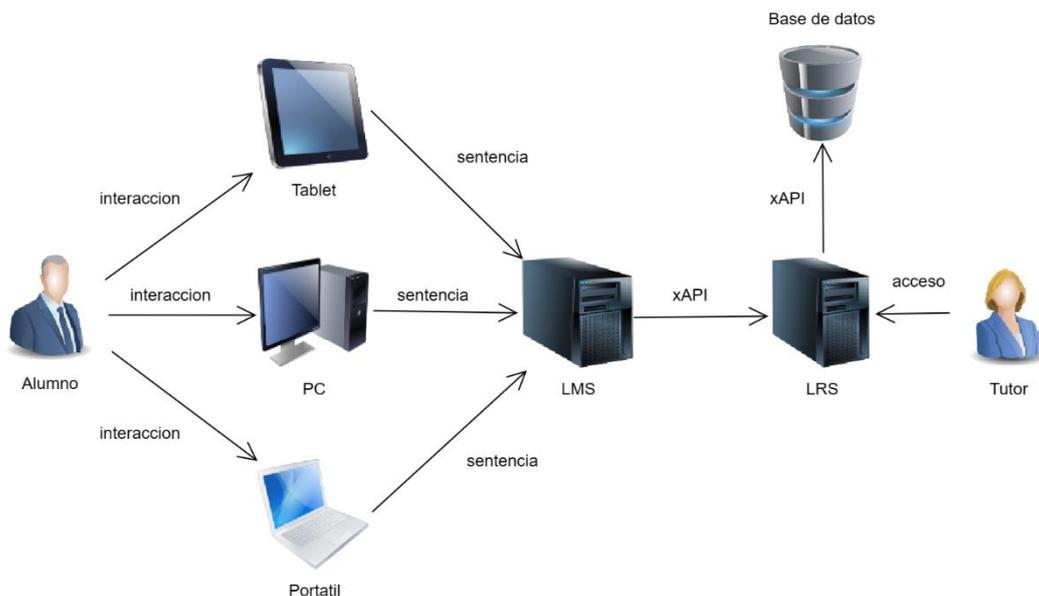


Figura 6. Recopilación y almacenamiento de datos

## 4.2 Análisis y visualización de los resultados

En esta fase implementaremos un módulo o servicio con acceso a la base datos que se encargará de obtener los datos necesarios para realizar un análisis que podrá ser individual de cada alumno o general, para evaluar diversos aspectos de los recursos educativos como la dificultad, el progreso general, el individual, etc.

Una vez ejecutado el análisis, se mostrará una visualización de estos.

Seguidamente en la Figura 7. Análisis de datos se comprueba como el módulo de análisis obtiene los datos necesarios, para ejecutar la analítica, de la base de datos. Una vez obtenidos y preparados dichos datos se ejecuta la lógica analítica mostrando los resultados al tutor, que tendrá acceso a dicha información y podrá seleccionar el tipo de análisis (individual o general).

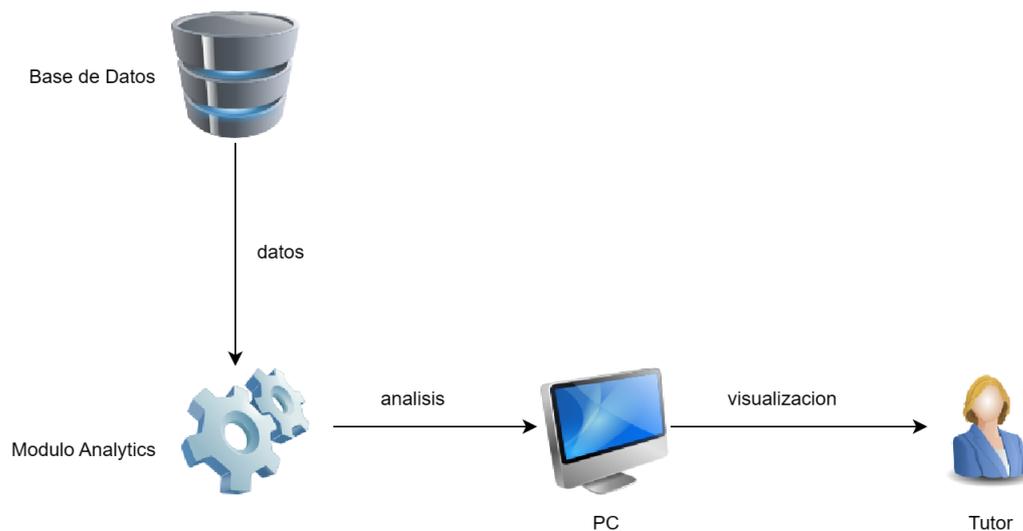


Figura 7. Análisis de datos



## 5. Propuesta de arquitectura y diseño

### 5.1 Introducción

Se propone una arquitectura del tipo **cliente-servidor**. En este tipo de sistemas un cliente enviará peticiones de diferentes tipos al servidor, el cual devolverá las respuestas correspondientes vía web. Las peticiones podrán ser de envío, actualización o de solicitud de información. Cabe destacar que en este tipo de arquitecturas la capacidad de procesamiento se reparte entre ambos.

En la Figura 8. Modelo cliente-servidor podemos observar la interacción de un sistema de estas características donde los dispositivos envían peticiones a través de internet al servidor y éste les devuelve las respuestas por el mismo medio.

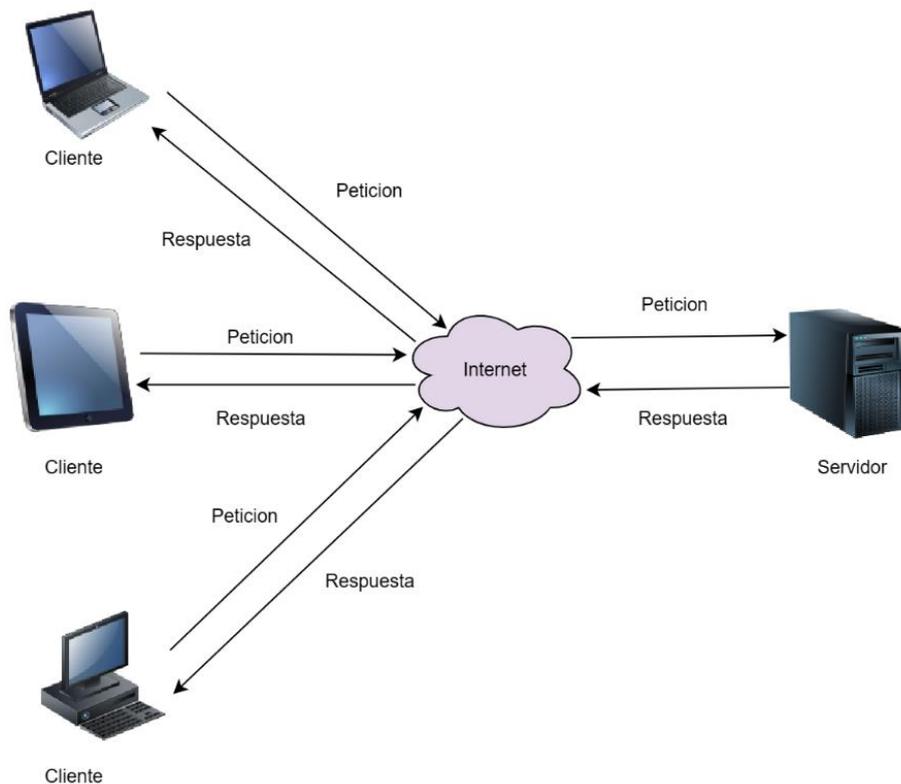


Figura 8. Modelo cliente-servidor

Los distintos servicios se integrarán y desplegarán utilizando la tecnología de Docker o contenedores. Estarán encapsulados y conectados entre sí, creándose a su vez una red interna asignando a cada servicio una dirección individual.

A continuación, describimos el diseño de los diferentes servicios.

## 5.2 Servidor

El servidor encargado del tráfico entre sistemas constará de un único contenedor, donde se instalará el servidor NGINX. Su función será la de regular el tráfico proveniente de los LMSs y de los usuarios que acceden al módulo Analytics y al LRS.

En la Figura 9. Diseño del servidor apreciamos el contenedor NGINX correspondiente al servidor que dirige el tráfico proveniente del LMS (sentencias xAPI) y al acceso del usuario a los contenedores LRS y Analytics donde se instalará el módulo de análisis.

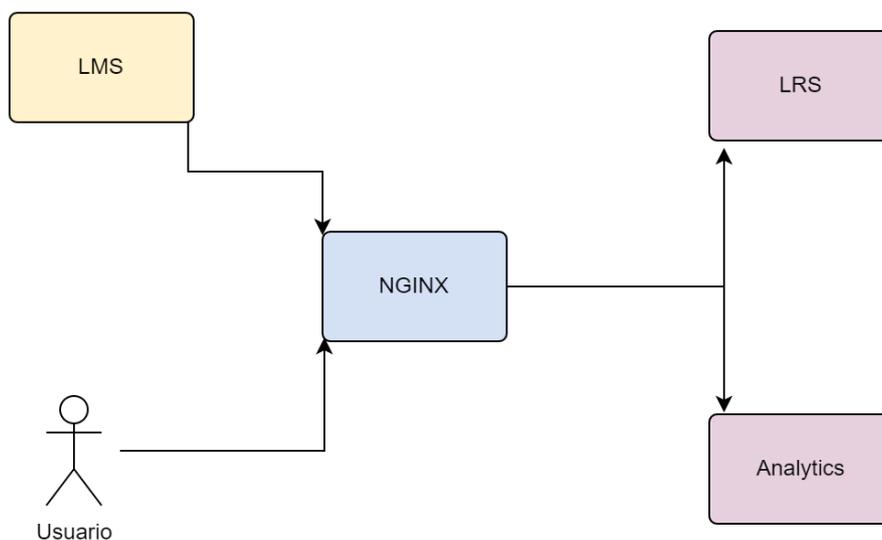


Figura 9. Diseño del servidor

## 5.2 LRS

Debemos dotar o administrar al LRS de los servicios esenciales para realizar su función, instalando cada servicio en un contenedor garantizando así una administración eficiente. Los servicios serán los siguientes:

- LearningLocker, núcleo del sistema que permitirá devolver y visualizar las sentencias estructuradas en el estándar xAPI. Para cumplir con su objetivo por supuesto, debe tener acceso a la base de datos y a la caché de memoria.
- Servicio xAPI, la función de este sistema será la de analizar las sentencias para asegurar su coherencia y que la estructura corresponda en el estándar especificado, para que posteriormente se pueda almacenar en la base de datos.
- MongoDB, base de datos encargada del almacenamiento de todos los datos enviados por el LRS.
- Redis, memoria de datos en caché que permitirá agilizar las búsquedas.

Serán alojados en una red interna, a la que solo se podrá acceder a través del servidor NGINX. Tanto el contenedor LearningLocker, como el servicio xAPI tendrán acceso a la base de datos y a la memoria caché. Esta estructura queda reflejada en la Figura 10. Diseño del LRS donde podemos observar los cuatro contenedores que conforman el LRS y el acceso del servicio xAPI y LearningLocker a MongoDB y Redis.

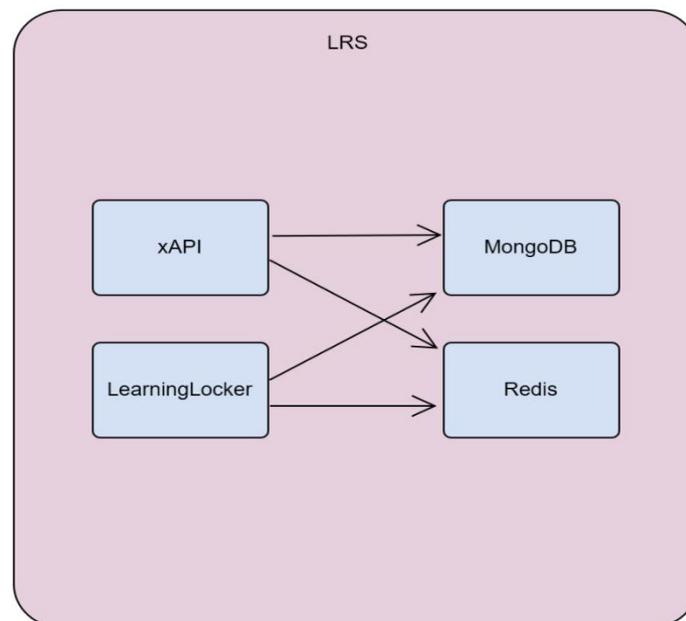


Figura 10. Diseño del LRS

## 5.4 Módulo de análisis

Este módulo complementará la anterior estructura con la finalidad de implementar una herramienta de análisis de datos. Constará de un único contenedor donde se instalarán los sistemas necesarios para realizar su función, accesible desde el servidor y con acceso al servicio del LRS de la base de datos. Estará incluido en la misma red que el LRS.

La función de este módulo será la de analizar las sentencias que contengan en el campo de resultado una nota numérica. No tendrá efectividad si no existen sentencias con algún resultado. En el Anexo 1 se especifican las condiciones que deben reunir las sentencias y el curso, para el cual se implementará el módulo.

Realizará dos tipos de análisis:

- **Análisis general:** se calcularán los siguientes conceptos estadísticos:
  - La media: media aritmética de todos los resultados.
  - La moda: valor más repetido.
  - La mediana: nos indica los valores que se encuentran en la zona media de la distribución.
  - La desviación típica: nos muestra una medida de dispersión de los datos.

Estos resultados nos permitirán medir algunas métricas del curso como:

- El desempeño.
- El progreso.
- La dificultad.

Los valores anteriores servirán al tutor para solucionar posibles problemas con el recurso educativo.

- **Análisis individual:** El análisis individual tiene como objetivo calcular o predecir el resultado probable que pudiera obtener un alumno a partir de los datos obtenidos de su interacción con un LMS.

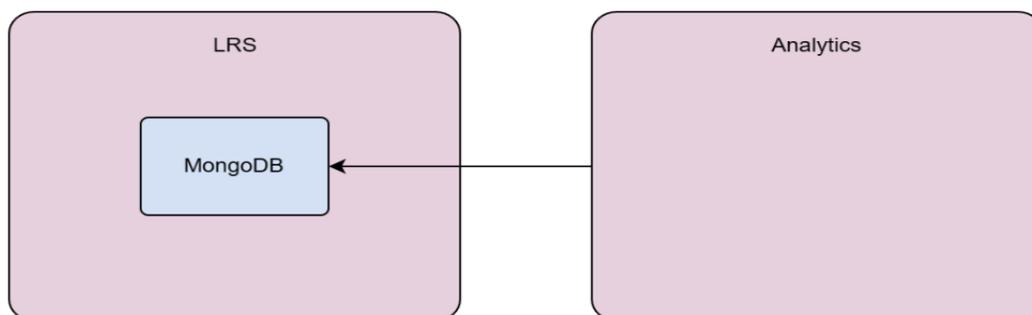
Para este fin se implementa un algoritmo de inteligencia artificial, de aprendizaje supervisado, denominado **Regresión Lineal Multivariable**.

La regresión lineal es un método que nos permitirá predecir el resultado o nota numérica de la realización de una prueba, dependiendo de sus interacciones con el LMS. Es decir, analizando los datos de las interacciones anteriores de otros alumnos, podremos predecir el resultado.

Una vez conocido el resultado, podremos anticipar soluciones para que ese alumno aumente sus probabilidades de mejorar dicho resultado.

Para complementar la visualización del resultado, también se mostrará el porcentaje de alumnos que han finalizado cada actividad que el alumno analizado no ha realizado, podremos utilizar dicha información para aconsejar al alumno la realización de diversas tareas, aumentando así las probabilidades de obtener una nota mayor.

La estructura simple de la Figura 11. Diseño del módulo Analytics nos permite observar el contenedor del módulo Analytics y su acceso a la base de datos.



*Figura 11. Diseño del módulo Analytics*



---

## 6. Implementación

Cada subsistema se organizará en una carpeta, que contendrá todos los archivos necesarios para su creación y configuración.

Docker nos permitirá crear una imagen por servicio y a partir de ésta podremos crear tantos contenedores como sean necesarios, por lo cual podremos crear varias arquitecturas simultaneas.

Debemos destacar que los datos almacenados en los diferentes subsistemas o contenedores desaparecerán al eliminar el contenedor, es por ello por lo que para establecer que los datos sean persistentes Docker dispone de un sistema de creación de volúmenes que enlaza una dirección de memoria del contenedor con una dirección de memoria del equipo, solucionando el problema y dándonos la posibilidad de almacenar únicamente los datos necesarios.

Utilizaremos el software gratuito **Visual Studio Code** (VSC) (Microsoft, 2023) para gestionar los archivos de cada imagen, crear el código del módulo Analytics, así como las imágenes y contenedores correspondientes. Este software también nos permitirá iniciar el sistema, pararlo y reiniciarlo, así como el acceso a los datos de cada contenedor.

### 6.1 LRS

Para establecer una arquitectura con varios contenedores y una conexión entre ellos nos apoyaremos de los archivos denominados **docker-compose.yml**.

La estructura de estos archivos sería la siguiente:

VERSION:

SERVICES:

NOMBRE\_DEL\_SERVICIO:

VARIABLE\_DE\_CONFIGURACION:

VALORES

VARIABLE\_DE\_CONFIGURACION:

VALORES

NOMBRE\_DE\_OTRO\_SERVICIO:

VARIABLE\_DE\_CONFIGURACION:

VALORES

Cada servicio dispondrá de:

- Un nombre del servicio
- El nombre del contenedor
- El nombre de la imagen de la cual se creará
- Los puertos expuestos
- Las dependencias entre contenedores (nos indicará que no se podrá iniciar un contenedor hasta que no se haya iniciado otro anteriormente)
- Adicionalmente también se especificarán las rutas de los datos permanentes en el caso de las bases de datos
- Opcionalmente podremos crear y configurar nuestra propia red

Una vez creadas todas las imágenes utilizaremos este archivo para iniciar el sistema.

Para la creación de imágenes implementaremos unos archivos de texto denominados **Dockerfile** que utilizan una gramática particular de Docker con la que podremos compilar una imagen desde una descripción textual.

Estructura de un archivo Dockerfile:

- **FROM:** todos los archivos de este tipo comienzan siempre con este comando, que nos permite definir una imagen básica sobre la que se construirá la nueva imagen.
- **RUN:** permite lanzar un proceso.
- **ENV:** permite determinar las variables de entorno.
- **VOLUME:** permite desacoplar el almacenamiento de datos del contenedor. Esto es especialmente útil en bases de datos. Los datos serán permanentes en la ubicación que establezcamos en cualquier equipo.
- **COPY:** permite copiar archivos locales en nuestra imagen.
- **ENTRYPOINT:** define el comando utilizado para inicializar el contenedor
- **EXPOSE:** permite exponer un puerto del contenedor al exterior
- **CMD:** son los argumentos por defecto que recibirá el comando establecido en ENTRYPOINT

Todos los archivos de configuración de cada servicio serán incluidos en la imagen creada.

En la Figura 12. Contenedores del LRS se muestra la arquitectura formada después del despliegue de los contenedores del LRS. Podemos observar como el acceso al sistema se realiza a través del contenedor que aloja el servidor nginx que distribuye el tráfico generado por los usuarios y el LMS entre el servicio xapi y LearningLocker, a su vez estos acceden a la base de datos y a la memoria caché en el caso de que sea necesario. El LMS enviará sentencias en el estándar xAPI que serán redirigidas al servicio xapi y el tutor tendrá acceso a dichas sentencias a través de LearningLocker.

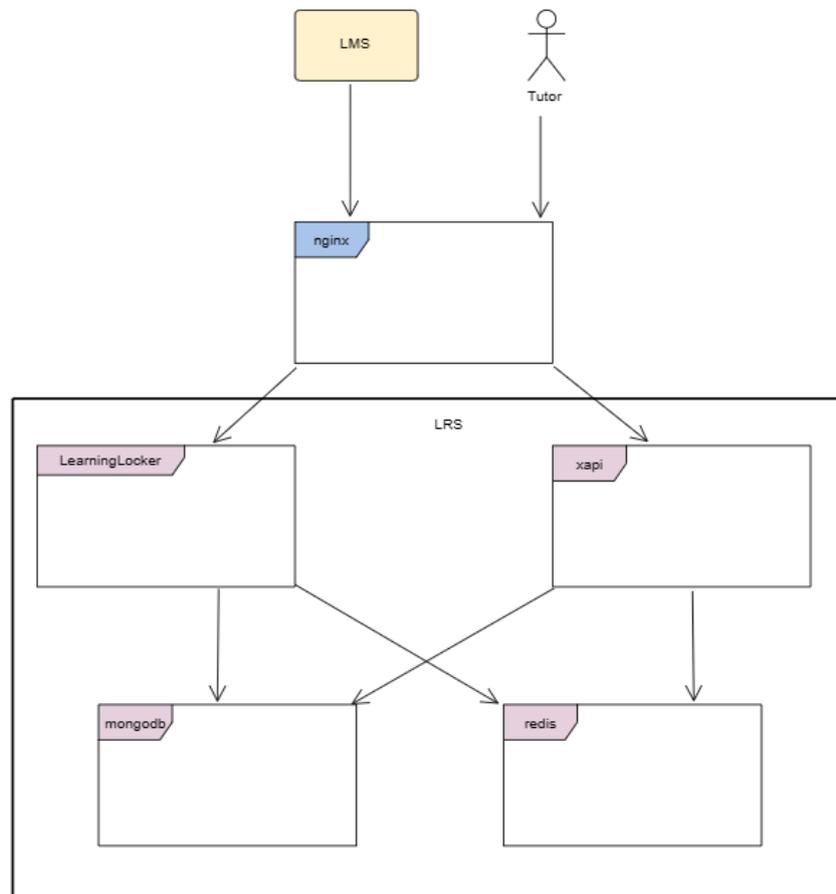


Figura 12. Contenedores del LRS

## 6.2 Módulo Analytics

Se implementará un módulo que como los anteriores será desplegado mediante Docker. Dicho servicio podrá acceder a los datos almacenados en la base de datos MongoDB de nuestra aplicación y se accederá a él a través del servidor NGINX ya desplegado.

Para la creación del módulo Analytics utilizaremos los siguientes recursos:

- **Nodejs:** es un entorno de tiempo de ejecución de JavaScript. Este entorno está especialmente indicado para la creación de aplicaciones de red rápidas, siendo capaz de manejar una gran cantidad de conexiones simultáneas.
- **Handlebars:** motor de plantillas que nos permite generar código HTML utilizando para ello objetos de datos en formato JSON. Nos será especialmente útil a la hora de visualizar los datos y para desarrollar las vistas de nuestra aplicación.
- **Bootstrap:** framework que nos permite y facilita el desarrollo del front-end de una aplicación, optimizando los elementos de una página web.

Dispondremos de una carpeta con los archivos necesarios para crear la imagen Docker, en el mismo directorio que los servicios anteriores.

A este servicio **solo podrán acceder los usuarios o tutores que estén registrados en el LRS.**

El presente módulo dispondrá de varias opciones por cada curso:

- Declaraciones
- Alumnos
- Análisis:
  - Análisis Individual:
    - Info
    - Datos
    - Histórico
  - Análisis General:
    - Info
    - Datos
    - Histórico

Seguidamente en la Figura 13. Opciones del curso se muestra el gráfico que representa las opciones disponibles por cada curso relacionadas anteriormente.

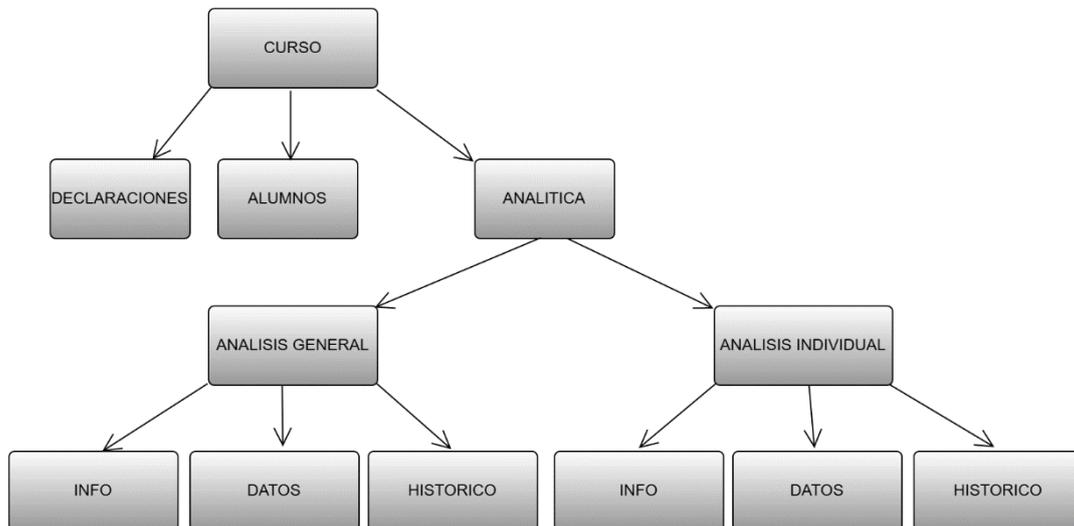


Figura 13. Opciones del curso

Toda la información se mostrará mediante tablas.

### 6.2.1 Declaraciones

En este caso se reflejará una lista de todas las sentencias almacenadas del curso seleccionado en formato ACTOR-VERBO-OBJETO.

El campo VERBO se nos presentara como un enlace al URI del verbo, donde podremos acceder y obtener más información sobre su significado.

Además, se mostrará información como la fecha de almacenamiento de la sentencia, la duración y el resultado numérico en caso de que disponga de dichos campos.

En la Tabla 3. Tabla de Declaraciones podemos visualizar un ejemplo de cómo se verían las sentencias, con el ID o identificador con el que la sentencia se guardó en la base datos, el nombre del alumno, seguido del verbo y el objeto. También se incluye la duración en el caso de disponer de ella, así como la puntuación si fuera una actividad calificable.

#	ID	NOMBRE	VERBO	OBJETO	FECHA	DURACIÓN	PUNTUACIÓN
0	234vfss	Alumno1	Inició	Curso	12/01/2023 15:30:45		
1	4543fdfd	Alumno2	Visualizó	Video1	13/01/2023 14:20:21	PT20S	
2	324fdfs	Alumno3	Finalizó	Prueba1	15/01/2023 12:10:15	PT50S	8.5

Tabla 3. Tabla de Declaraciones

### 6.2.2 Alumnos

Esta opción nos permitirá visualizar una lista de los alumnos que han interactuado con el curso. La información mostrada será el número de alumnos registrados, el nombre y el correo electrónico de cada uno.

En la Tabla 4. Tabla de Alumnos obtenemos un ejemplo de la información detallada anteriormente.

#	NOMBRE	E-MAIL
0	Alumno1	alumno1@example.com
1	Alumno2	alumno2@example.com
2	Alumno3	alumno3@example.com

Tabla 4. Tabla de Alumnos

### 6.2.3 Análisis

Esta alternativa nos desplegará otras opciones, como la posibilidad de realizar un análisis individual y otro general cada uno con tres subfunciones:

- Análisis general:
  - o Info: nos mostrara una pequeña información sobre el análisis.
  - o Histórico: se recuperan todos los análisis realizados, junto a la fecha de realización y el resultado del análisis.

A continuación, en la Tabla 5. Histórico del Análisis General se muestra un ejemplo de la tabla de análisis, donde se nos mostrara la fecha de

creación del análisis, el rango de fechas afectado y los resultados de la media, la mediana, la moda y la desviación.

#	FECHA	RANGO FECHAS	MEDIA	MEDIANA	MODA	DESVIACION
0	12/01/2013	10/12/2012-15/12/2012	8.56	9.02	7	1.51
1	22/01/2013	15/12/2012-25/12/2012	4.23	5.01	3	3.23
1	01/02/2012	25/12/2012-02/01/2013	7.21	6.02	6	1.23

Tabla 5. Histórico del Análisis General

- Datos: podremos elegir mediante la implementación de un calendario un rango de fechas entre las cuales el sistema buscara las sentencias que se añadieron a la base de datos que contienen una nota numérica y que son susceptibles de análisis.

En la Tabla 6. Tabla de Sentencias analizables se visualiza un ejemplo que nos muestra en cada fila el alumno que realizo la sentencia, la puntuación de la actividad, la duración si se dispone de ella y la fecha de ejecución de la prueba. Estos datos son meramente informativos, pero sobre ellos se realizará el análisis general.

#	NOMBRE	PUNTUACION	DURACION	FECHA
0	Alumno1	8.25	PT40S	15/01/2023 12:30:02
1	Alumno2	7	PT50S	16/01/2023 11:25:25
2	Alumno3	4.23	PT24S	17/01/2023 10:30:01
3	Alumno4	2	PT67S	17/01/2023 11:25:03

Tabla 6. Tabla de Sentencias analizables

Una vez ejecutado el análisis el resultado se mostrará como indica la Tabla 7. Resultados Análisis General. Podemos observar que se muestra la fecha del análisis, y el resultado de la media, la mediana, la moda y la desviación típica.

FECHA	MEDIA	MEDIANA	MODA	DESVIACION
15/12/2022	6.52	7.23	5	5.25

Tabla 7. Resultados Análisis General

- Análisis individual:
  - Info: nos mostrará una pequeña información sobre el análisis.
  - Histórico: se recuperan todos los análisis realizados, junto a la fecha de realización y el resultado del análisis. En la Tabla 8. Histórico del Análisis Individual se muestra un ejemplo con la información del nombre del alumno que fue seleccionado para el análisis, la fecha de este y el resultado probable que el alumno obtendrá en su prueba, a partir de sus interacciones con el LMS.

#	ALUMNO	FECHA	Resultado
0	Alumno1	15/01/2023- 20:49:00	4.5
1	Alumno2	20/01/2023- 15:45:00	8.5
2	Alumno3	21/01/2023 09:15:45	1.2

Tabla 8. Histórico del Análisis Individual

- Datos: Podremos elegir entre todos los alumnos que aún no tienen nota en la prueba mediante una tabla. Todas las sentencias que serán analizadas se integrarán en una tabla para su visualización con un resultado binario que indicará si el alumno seleccionado y los alumnos con alguna nota han realizado la actividad, por espacio en la tabla solo se mostrara el verbo de la sentencia. El resultado del análisis será una nota numérica, acompañada de un gráfico que nos permitirá comprobar que el algoritmo ha funcionado correctamente. Dicho gráfico se

corresponderá con el coste de la función del algoritmo cuya implementación se explica en el siguiente punto. En el resultado también se incluirá una tabla con el porcentaje de realización de las actividades que el usuario seleccionado para el análisis no ha realizado.

En la Tabla 9. Selección de alumno se muestra una lista de los alumnos que pueden ser seleccionados para el análisis individual.

Posteriormente en la Tabla 10. Tabla binaria del alumno seleccionado y en la Tabla 11. Tabla binaria de los alumnos con nota se muestran las sentencias realizadas por el alumno sobre el que va a realizarse el análisis y las sentencias realizadas por los alumnos que ya han obtenido una nota en la prueba. Estos se corresponden con todos los datos que formaran parte del análisis. Finalmente, en la Tabla 12. Resultado del Análisis Individual y en la Tabla 13. Porcentaje de actividades realizadas se muestra la información del resultado, con el nombre del alumno y la nota probable que obtuviese si realizara la prueba en base a las actividades que ha desarrollado, además se muestra el porcentaje de realización de las sentencias por los alumnos simultáneamente.

La Figura 14. Gráfica del coste de la función muestra la evolución del coste de la función del análisis que se explicará a continuación y que nos será de utilidad para comprobar la eficiencia del algoritmo.

SELECCION
Alumno1 / alumno1@example.com
Alumno2 / alumno2@example.com

Tabla 9. Selección de alumno

NOMBRE	INICIADO	VISTO	VISUALIZADO	EXPERIMENTADO	PUNTUACION
Alumno1	1	0	0	1	

Tabla 10. Tabla binaria del alumno seleccionado

NOMBRE	INICIADO	VISTO	VISUALIZADO	EXPERIMENTADO	PUNTUACION
Alumno3	1	1	1	0	6
Alumno4	0	1	1	1	5.25
Alumno5	0	0	0	0	2.36
Alumno6	1	1	1	1	8.25
Alumno7	0	0	0	0	0

Tabla 11. Tabla binaria de los alumnos con nota

ALUMNO	RESULTADO PROBABLE
Alumno1	6.7

Tabla 12. Resultado del Análisis Individual

VERBO	OBJETO	PORCENTAJE
INICIADO	ACTIVIDAD	66.58 %
VISTO	VIDEO1	45.25 %
VISUALIZADO	VIDEO2	80 %
EXPERIMENTADO	TAREA1	50 %

Tabla 13. Porcentaje de actividades realizadas

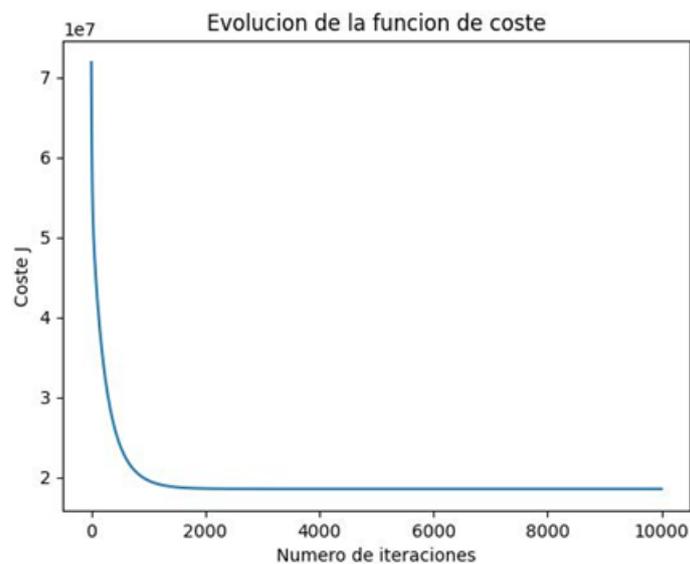


Figura 14. Gráfica del coste de la función

### 6.3 Algoritmo de Regresión Lineal

Se implementa un algoritmo de Regresión Lineal con múltiples variables (University, 2023) en **Python**. Este algoritmo básicamente resuelve un problema de minimización de una función de coste  $J$ .

En la Figura 15. Esquema del algoritmo de Regresión Lineal se esquematiza cómo funciona este algoritmo. Partimos de un conjunto de entrenamiento compuesto, en este caso, por todas las acciones realizadas por los alumnos que ya han sido examinados, al que le aplicamos nuestro algoritmo de aprendizaje cuya misión es ejecutar una función denominada hipótesis que toma como entrada las variables del conjunto de entrenamiento y produce un resultado probable.

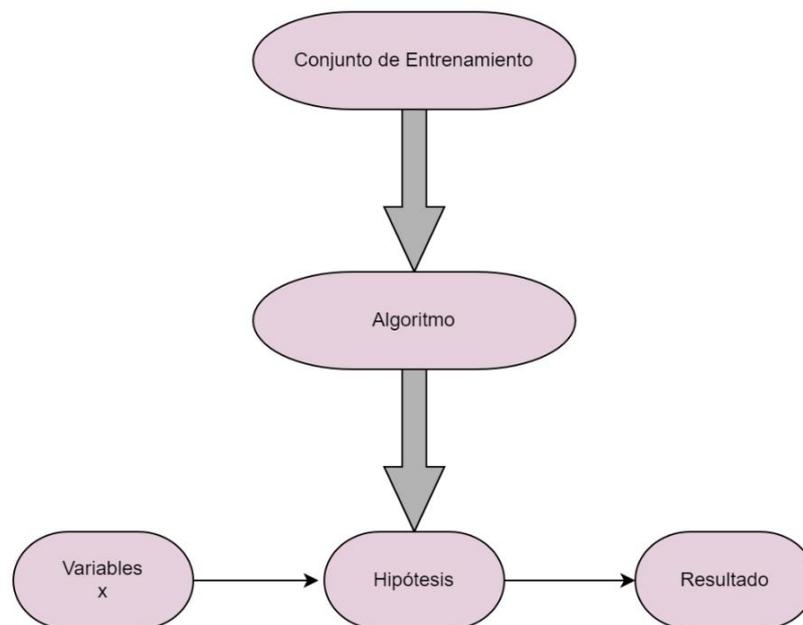


Figura 15. Esquema del algoritmo de Regresión Lineal

Para el desarrollo del algoritmo partimos de un conjunto de entrenamiento formado por todas las interacciones de todos los alumnos que ya han realizado la prueba. Cada interacción constituirá una variable  $x$ . Se formulará la siguiente hipótesis:

$$h_{\theta}(x) = \theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2 + \dots + \theta_n x_n = \theta^T x$$

Donde:

- $x_i$  es el valor de la variable
- $\theta_i$  son los parámetros del modelo, que formaran un vector dimensional  $n+1$ , donde  $n$  es el número de variables

El algoritmo aplicará a nuestra hipótesis la siguiente fórmula para minimizar los parámetros  $\theta_i$ :

$$\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)}) x_j^{(i)}$$

Donde:

- $\alpha$  es el índice de aprendizaje, será configurado en 0,01 como predeterminado.
- $m$  es el número de entradas en el conjunto de entrenamiento.
- $y$  es el valor de los resultados de nuestro conjunto de entrenamiento.
- $x^{(i)}$  es un vector que contiene todos los parámetros de la fila  $i$ . En la Tabla 14. Conjunto de entrenamiento del Algoritmo se correspondería con:

$$x^2 = \begin{matrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{matrix}$$

También se implementará una función de coste  $J$  que será la siguiente:

$$\theta_j := \theta_j - \alpha \frac{1}{2m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)}) x_j^{(i)}$$

Dicha función nos permitirá comprobar la efectividad del algoritmo, ya que mediante su gráfica podremos visualizar que realmente se produce el descenso del coste de dicha función.

Los resultados binarios de cada actividad vendrán dados por 1 que indicara que se ha realizado la actividad y 0 que será lo contrario.

Observamos un ejemplo de conjunto de entrenamiento en la Tabla 14. Conjunto de entrenamiento del Algoritmo con número de variables  $n = 5$ , y número de entradas  $m = 3$ , donde podemos observar las variables nombradas con  $X_n$ , el resultado con  $y$ , además del verbo de las actividades y un código binario con 1 como realizado y 0 como no realizado.

$X_1$	$X_2$	$X_3$	$X_4$	$X_5$	$y$
Leer texto	Ver video	Actividad 1	Actividad 2	Actividad 3	Resultado
1	0	0	0	1	4
1	1	1	1	1	8
0	0	0	1	1	1

Tabla 14. Conjunto de entrenamiento del Algoritmo

Para implementar la función en Python trabajaremos con vectores y matrices utilizando el siguiente método:

Implementar el algoritmo y la función de coste para cada parámetro  $\Theta$  con un bucle por cada iteración (el número de iteraciones será configurado de forma predeterminada en 10000) de la siguiente manera:

Por convención  $x_0 = 1$  por lo que:

$$\Theta_0 = \Theta_0 - \alpha \frac{1}{m} * ((h_{\Theta}(x^{(1)}) - y^{(1)}) x_0^{(1)}) + (h_{\Theta}(x^{(2)}) - y^{(2)}) x_0^{(2)}) + (h_{\Theta}(x^{(3)}) - y^{(3)}) x_0^{(3)})$$

$$\Theta_1 = \Theta_1 - \alpha \frac{1}{m} * ((h_{\Theta}(x^{(1)}) - y^{(1)}) x_1^{(1)}) + (h_{\Theta}(x^{(2)}) - y^{(2)}) x_1^{(2)}) + (h_{\Theta}(x^{(3)}) - y^{(3)}) x_1^{(3)})$$

$$\Theta_2 = \Theta_2 - \alpha \frac{1}{m} * ((h_{\Theta}(x^{(1)}) - y^{(1)}) x_2^{(1)}) + (h_{\Theta}(x^{(2)}) - y^{(2)}) x_2^{(2)}) + (h_{\Theta}(x^{(3)}) - y^{(3)}) x_2^{(3)})$$

$$\Theta_3 = \Theta_3 - \alpha \frac{1}{m} * ((h_{\Theta}(x^{(1)}) - y^{(1)}) x_3^{(1)}) + (h_{\Theta}(x^{(2)}) - y^{(2)}) x_3^{(2)}) + (h_{\Theta}(x^{(3)}) - y^{(3)}) x_3^{(3)})$$

$$\Theta_4 = \Theta_4 - \alpha \frac{1}{m} * ((h_{\Theta}(x^{(1)}) - y^{(1)}) x_4^{(1)}) + (h_{\Theta}(x^{(2)}) - y^{(2)}) x_4^{(2)}) + (h_{\Theta}(x^{(3)}) - y^{(3)}) x_4^{(3)})$$

$$\Theta_5 = \Theta_5 - \alpha \frac{1}{m} * ((h_{\Theta}(x^{(1)}) - y^{(1)}) x_5^{(1)}) + (h_{\Theta}(x^{(2)}) - y^{(2)}) x_5^{(2)}) + (h_{\Theta}(x^{(3)}) - y^{(3)}) x_5^{(3)})$$

Sustituyendo  $h_{\theta}(x^{(i)})$ :

$$\Theta_0 = \Theta_0 - \alpha \frac{1}{m} * (((\Theta^T x^{(1)}) - (y^{(1)})) x_0^{(1)} + ((\Theta^T x^{(2)}) - (y^{(2)})) x_0^{(2)} + ((\Theta^T x^{(3)}) - y^{(3)}) x_0^{(3)})$$

$$\Theta_1 = \Theta_1 - \alpha \frac{1}{m} * (((\Theta^T x^{(1)}) - (y^{(1)})) x_1^{(1)} + ((\Theta^T x^{(2)}) - (y^{(2)})) x_1^{(2)} + ((\Theta^T x^{(3)}) - y^{(3)}) x_1^{(3)})$$

$$\Theta_2 = \Theta_2 - \alpha \frac{1}{m} * (((\Theta^T x^{(1)}) - (y^{(1)})) x_2^{(1)} + ((\Theta^T x^{(2)}) - (y^{(2)})) x_2^{(2)} + ((\Theta^T x^{(3)}) - y^{(3)}) x_2^{(3)})$$

$$\Theta_3 = \Theta_3 - \alpha \frac{1}{m} * (((\Theta^T x^{(1)}) - (y^{(1)})) x_3^{(1)} + ((\Theta^T x^{(2)}) - (y^{(2)})) x_3^{(2)} + ((\Theta^T x^{(3)}) - y^{(3)}) x_3^{(3)})$$

$$\Theta_4 = \Theta_4 - \alpha \frac{1}{m} * (((\Theta^T x^{(1)}) - (y^{(1)})) x_4^{(1)} + ((\Theta^T x^{(2)}) - (y^{(2)})) x_4^{(2)} + ((\Theta^T x^{(3)}) - y^{(3)}) x_4^{(3)})$$

$$\Theta_5 = \Theta_5 - \alpha \frac{1}{m} * (((\Theta^T x^{(1)}) - (y^{(1)})) x_5^{(1)} + ((\Theta^T x^{(2)}) - (y^{(2)})) x_5^{(2)} + ((\Theta^T x^{(3)}) - y^{(3)}) x_5^{(3)})$$

Una vez calculado todas las incógnitas  $\Theta$  calcularemos  $h_{\theta}(x) = \Theta_0 x_0 + \Theta_1 x_1 + \Theta_2 x_2 + \dots + \Theta_n x_n$  sustituyendo los valores  $\Theta$  obtenidos anteriormente y los parámetros  $x_n$  que se compondrán con la lista binaria de actividades realizadas por el alumno seleccionado.

El resultado final será una nota numérica.

Paralelamente y en cada iteración se implementa la función de coste, cuyo resultado se irá guardando en un vector con el que dibujaremos la gráfica posterior, donde comprobaremos la efectividad del algoritmo.

En definitiva, el algoritmo presentado busca un patrón dentro del conjunto de entrenamiento semejante al patrón del usuario seleccionado, mostrándonos la nota probable que obtendría con las interacciones que ha realizado con el sistema.

## 6.4 Arquitectura final

El resultado de la implementación es una arquitectura como la reflejada en la Figura 16. Arquitectura completa donde confirmamos gráficamente que el servidor distribuye el tráfico proveniente del LMS y del tutor entre el LRS y el módulo Analytics que además tiene acceso a los datos almacenados en el LRS.

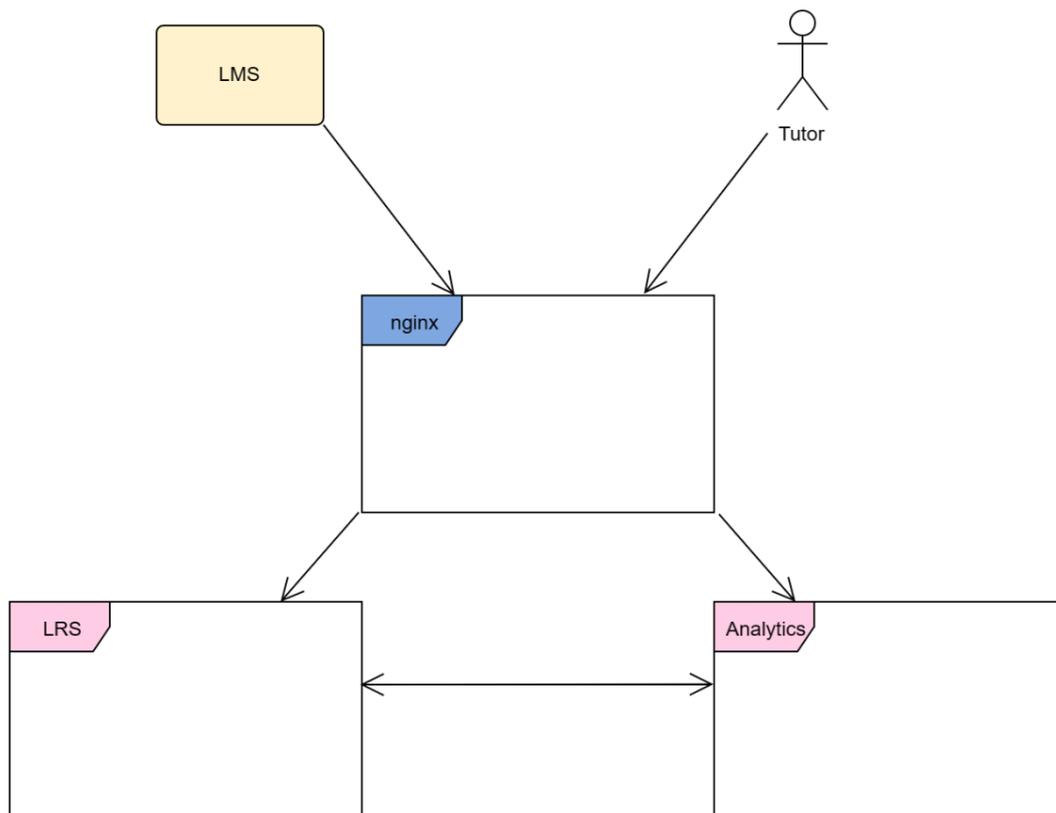


Figura 16. Arquitectura completa

## 6.5 Conclusiones

En conclusión, el proyecto desarrollado para mejorar la eficiencia del aprendizaje en los usuarios de plataformas de aprendizaje interactivo nos permitirá recopilar, almacenar y analizar todos los datos recibidos de dichas plataformas, obteniendo resultados analíticos que nos posibilita modificar y solucionar problemas con el aprendizaje individual y generalmente. Así mismo, nos ha permitido utilizar diversas tecnologías e integrarlas en un único sistema completamente funcional creando una arquitectura simple.



## 7. Pruebas

### 7.1 Introducción

Se realizan pruebas a todos los subsistemas individualmente para determinar el grado de funcionalidad. A continuación, las pruebas de integración nos permitirán medir la compatibilidad entre sistemas, rendimiento de la arquitectura, portabilidad y escalabilidad. Las pruebas de validación consolidarán el diseño conforme a los resultados obtenidos.

### 7.2 Pruebas Unitarias

Primero corroboramos que las imágenes se crean sin ningún tipo de problema, así como la implementación de los contenedores y su puesta en marcha.

Se realizan pruebas a los diferentes subsistemas, una vez creadas las imágenes y el contenedor correspondiente a cada servicio por separado, comprobando que cada unidad funciona como se espera antes de la integración. Se desarrollan **pruebas de accesibilidad, rendimiento y funcionalidad**. Las pruebas de accesibilidad nos permitirán comprobar que podemos acceder a cada subsistema y realizar cambios en los archivos. Las pruebas de rendimiento nos mostraran hasta qué punto el sistema es estable. Las pruebas de funcionalidad permitirán ratificar que el sistema funciona adecuadamente.

Las pruebas se realizan en un Pc doméstico con las siguientes características:

- Procesador: AMD Ryzen 5 3500X 6-Core
- Memoria RAM: 16 GB
- Disco duro: SSD 512 GB
- Sistemas operativos Windows 11 y Linux Mint

#### 7.2.1 Nginx

- Prueba de accesibilidad: se realiza una prueba de accesibilidad al sistema, iniciando el contenedor con VSC y accediendo vía consola a los archivos del sistema, modificando, copiando o eliminando archivos. El resultado es correcto y satisfactorio permitiendo acceder al sistema y ejecutar comandos por consola

dentro del contenedor. También accedemos a los logs del sistema, verificando que no existen errores.

- Prueba de funcionalidad: comprobamos el acceso al servidor a través del navegador y la dirección configurada con un resultado correcto.
- Prueba de rendimiento: ejecutamos varias pruebas de carga online con el software **loader** (SendGrid, 2022).

La primera prueba será la de mantener 250 clientes simultáneos. Podemos observar en la Figura 17. Prueba de carga del servidor 250 usuarios que el tiempo de respuesta se mantiene estable en 100 ms aun cuando tenemos 250 conexiones simultaneas.

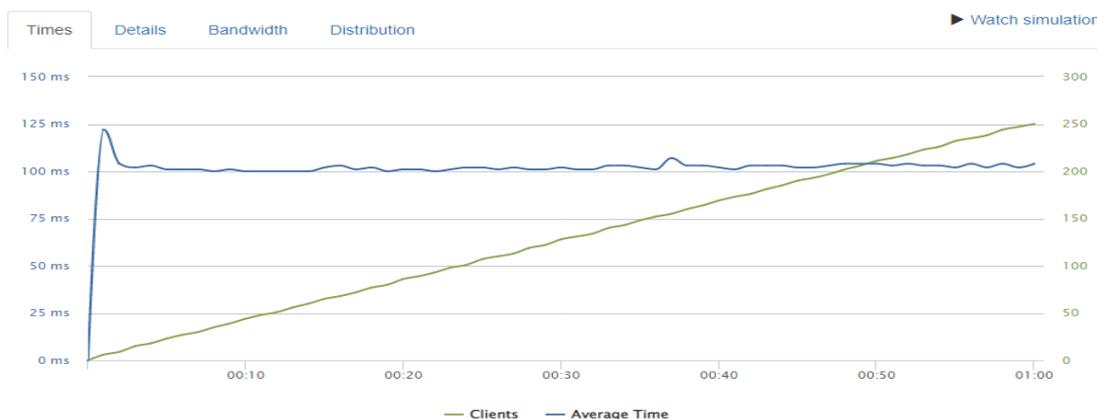


Figura 17. Prueba de carga del servidor 250 usuarios

Realizamos otra segunda prueba de carga con 500 conexiones simultaneas donde podemos comprobar que oscila entre un rango de 100 ms - 125 ms de tiempo de respuesta, más que suficiente para el propósito del proyecto. Se representa el resultado en la Figura 18. Prueba de carga del servidor 500 usuarios.

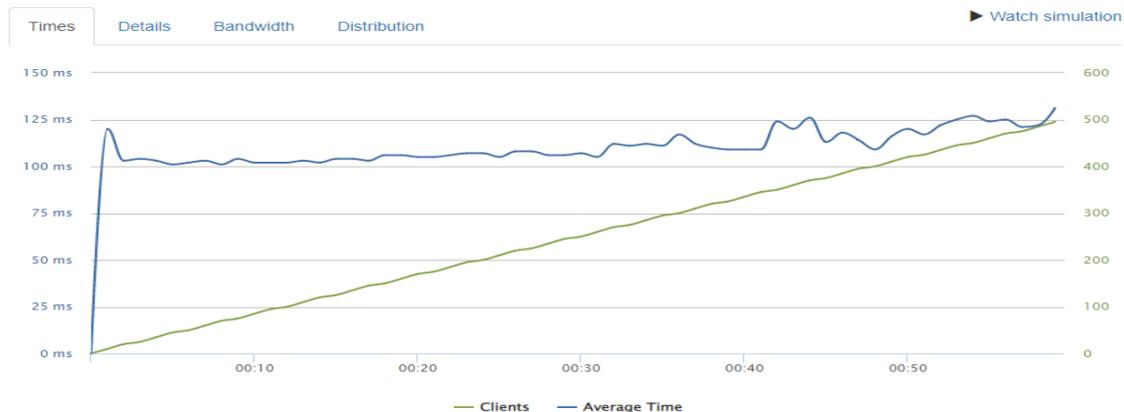


Figura 18. Prueba de carga del servidor 500 usuarios

### 7.2.2 LRS

- Pruebas de accesibilidad: como en el caso anterior realizamos las pruebas de accesibilidad sobre todos los contenedores que conforman el LRS. Accedemos a cada contenedor individualmente verificando que podemos ejecutar comandos por consola. También comprobamos los logs de cada sistema cerciorándonos de la ausencia de errores.
- Pruebas de funcionalidad: para la ejecución de este tipo de pruebas utilizaremos el software Postman (Postman, 2023), que nos permite enviar peticiones vía web en formato JSON y comprobar el correcto funcionamiento. Se realizan varias pruebas (GET, POST, PUT) con el resultado esperado, corroborando la correcta funcionalidad.

### 7.2.3 Módulo de Análisis

- Pruebas de accesibilidad: en este caso confirmamos que podemos acceder al contenedor del módulo sin ningún problema, ejecutando comandos en consola.
- Pruebas de funcionalidad: evidenciamos que podemos acceder vía web a través de la página de confirmación del usuario registrado y ejecutar todas las opciones del módulo sin problemas ni errores.

### 7.3 Pruebas de integración

Una vez iniciado el sistema con todos los contenedores funcionando y el servidor activo, consultamos todos los logs de los diferentes subsistemas verificando que no existen errores de ningún tipo. Se realiza la comprobación tanto en un sistema operativo Windows como en Linux.

También se revisa el tiempo necesario para levantar y/o detener el sistema, no superando los 5 segundos en ningún caso.

### 7.4 Pruebas de validación

Se procede a revisar el funcionamiento global del sistema, en primer lugar, utilizando los prototipos de prueba disponibles en el sitio oficial de xAPI:

[https://xapi.com/prototypes-getting-started/?utm\\_source=google&utm\\_medium=natural\\_search](https://xapi.com/prototypes-getting-started/?utm_source=google&utm_medium=natural_search)

Utilizaremos los 3 prototipos disponibles para verificar que las sentencias creadas por la interacción de los usuarios con éstos se transmiten, se almacenan y se visualizan correctamente en el LRS. A continuación, se detallan las pruebas con cada prototipo:

- Tetris: videojuego que consiste en encajar unas piezas sobre otras. Este recurso ira creando sentencias conforme el usuario juega, especificando las acciones que va realizando. Se ejecutan 3 juegos diferentes, constatando que las sentencias llegan al LRS.
- Golf: un minicurso de golf, con una pequeña prueba de conocimientos al final. Como en el caso anterior ejecutamos 3 pruebas, obteniendo el mismo resultado satisfactorio.
- Localización: un sistema que nos permite almacenar información sobre las ubicaciones que un usuario ha visitado online. Repetimos el proceso de los dos prototipos anteriores con los mismos resultados.

Una vez recabada la información procedemos a entrar en el módulo de análisis. Aquí ratificamos que podemos acceder a las sentencias almacenadas de cada curso, así como al nombre de todos los usuarios que han utilizado los prototipos.

Con respecto al análisis el resultado es correcto, tanto en el análisis individual como en el general.

## 7.5 Conclusiones

A través de las pruebas demostramos que el sistema cumple con las especificaciones y objetivos propuestos con la ausencia de errores.

El sistema recibe, almacena y analiza los datos conforme a las descripciones de este proyecto.

Se recomienda el uso masivo con usuarios del LMS para realizar un seguimiento del comportamiento del sistema, al ser las pruebas de un usuario limitadas en términos de usabilidad, conectividad y rendimiento.

Se debe adaptar el módulo de análisis a la tipología del recurso educativo del LMS y no al revés. El módulo de análisis implementado en este proyecto será útil en cursos que cumplan con la siguiente metodología:

- Un conjunto de recursos educativos, como lecturas, videos, actividades, etc.
- Un examen o prueba final calificado con una nota numérica.

Un ejemplo de recurso serían las asignaturas virtuales de los grados de la UNED (implementados con el estándar xAPI), es decir compuestos por una serie de recursos y un examen final por cuatrimestre, donde se podría aplicar perfectamente este proyecto.

Si la estructura del curso no es como la anterior, habrá que adaptar el módulo de análisis, aunque el LRS recabará la información en cualquier tipo de recurso del LMS.



## 8. Conclusiones y trabajos futuros

### 9.1 Conclusiones

El presente proyecto nos ha permitido realizar una revisión del Learning Analytics o análisis del aprendizaje y del estándar xAPI demostrando que, en combinación, son herramientas fundamentales para mejorar la calidad de la enseñanza, a través del aprendizaje y la tutorización individual, mediante el análisis de los datos obtenidos en los entornos de aprendizaje virtuales o plataformas de aprendizaje interactivo. Se ha profundizado en sus conceptos y fundamentos teóricos, también en sus aplicaciones y beneficios ofreciendo diversa biografía de proyectos e investigaciones que corroboran esta conclusión.

En primer lugar, se han explorado las posibilidades del Learning Analytics y que soluciones puede aportar a los problemas del aprendizaje, utilizando el BigData o las grandes concentraciones de datos que podemos recopilar de la interacción de todos los usuarios, no solo de las citadas plataformas de aprendizaje sino también de juegos o simuladores. Se ha indagado sobre los tipos de datos de los que podemos extraer información útil y de los análisis que podemos realizar dependiendo de las necesidades del recurso educativo en cuestión, permitiendo su corrección o mejora a partir de las interacciones de los usuarios y su aprendizaje.

Posteriormente se estudia y se profundiza en el estándar xAPI, que nos ha permitido recabar todos esos datos necesarios en el Learning Analytics. Un estándar de datos que nace de la necesidad de estructurar el BigData educacional y dotarle de mejores características que sus antecesores como por ejemplo la mejora de las métricas de seguimiento, la posibilidad de registrar cualquier actividad y la desvinculación de una plataforma de aprendizaje en particular entre otras. A través del estudio de este estándar podemos afirmar y demostrar que actualmente es una herramienta indispensable como componente del Learning Analytics para obtener datos masivos y de calidad.

Una vez comentadas las características, objetivos y ventajas del Learning Analytics y del estándar xAPI se realiza un estudio y análisis sobre la integración y explotación de ambos recursos con el objetivo de crear una arquitectura capaz de obtener datos de un LMS y ofrecer un análisis que nos permita mejorar el recurso educativo alojado en el LMS. Se analizan diversas tecnologías sobre servidores, LRSs y bases de datos, además de un estudio sobre inteligencia artificial y conceptos estadísticos, ofreciendo un arquitectura sencilla, portable y escalable que nos permite cumplir con los objetivos y demostrando cómo aplicando la tecnología disponible al servicio del Learning Analytics

podemos demostrar los beneficios de esta ciencia y el impacto significativo que puede tener en la calidad y efectividad de la educación.

Podemos concluir, con la finalización de este proyecto, que disponemos de las herramientas necesarias para influir en el aprendizaje, adaptándolo y mejorándolo general e individualmente, ofreciendo una metodología y unos recursos específicos al tipo de estudiante que accede al conocimiento facilitando así el aprendizaje y asegurando una educación de calidad, cumpliendo así con el objetivo último de la enseñanza y de sus profesionales: **que los alumnos aprendan lo que deben y necesitan aprender.**

## 9.2 Trabajos futuros

En este trabajo se ha explorado como el Learning Analytics en combinación con xAPI facilita y personaliza el aprendizaje asegurando un incremento de la calidad de la educación, pero aún quedan áreas o campos que valdría la pena investigar en el futuro como:

- La integración del estándar xAPI en otros sistemas y herramientas como en plataformas de redes sociales donde analizar toda la información de los usuarios con fines estadísticos psicológicos, plataformas de comercio con un objetivo financiero o comercial, etc.....
- Implementar los análisis en tiempo real, para obtener resultados instantáneos y anticipar medidas a los problemas surgidos instantáneamente.
- Avanzar en el estudio de datos cualitativos e investigar el tipo de análisis que pueden realizarse con este tipo de datos y las soluciones que pueden aportar como por ejemplo comentarios de los estudiantes o discusiones en las redes sociales.
- Analizar la manera de avanzar en la protección y privacidad de los datos y la seguridad de los estudiantes a medida que se recopilan más y más datos.

Estas son algunos de los trabajos futuros que podrían realizarse en línea con este proyecto, pero también se podrían realizar mejoras en los diferentes subsistemas que entran o pueden entrar a formar parte del proyecto como, por ejemplo:

- La implementación de más métodos de análisis que nos permitan aprovechar toda la capacidad de análisis de los datos obtenidos con más tipos de análisis y métricas.
- Generalizar más el módulo de análisis para adaptarse a la estructura de los recursos educativos de más LMS.

- Implementar una capa de seguridad en el servidor cifrando la conexión para asegurar la privacidad y no queden expuestos a terceros los datos.
- Mejorar la interfaz de usuario para que resulte más profesional, incluyendo gráficos que mejoren la legibilidad de los resultados.



---

## Bibliografía

- Amo Filv, D. (2021). *Analtica del aprendizaje* (Segunda Edicin Digital ed.). Recuperado el 08 de Febrero de 2023, de <https://eduliticas.com/analitica-aprendizaje-30-experiencias-datos-aula/pdf>
- Careers, F. N. (2023). <https://www.nginx.com/>. Recuperado el 10 de Febrero de 2023, de nginx: <https://www.nginx.com/>
- Carmen Fernandez-Morante, B. C.-L.-J.-M.-O. (23 de Diciembre de 2021). Adaptive Learning Supported by Learning Analytics for Student Teachers' Personalized Training during in-School Practices. (J. P. Berasaluce, Ed.) *Sustainability*, 124. Recuperado el 09 de Febrero de 2023, de <https://www.mdpi.com/2071-1050/14/1/124>
- Chenoweth, I. A.-G. (Diciembre de 2017). Adoptar nuevas tendencias de elearning xAPI y LRS. *Revista de ciencias de la educacin*, pgs. 63-73. Recuperado el 06 de Febrero de 2023, de [https://www.ecorfan.org/republicofperu/research\\_journals/Revista\\_de\\_Ciencias\\_de\\_la\\_Educacion/vol1num2/Revista\\_de\\_Ciencias\\_de\\_la\\_Educacin\\_V1\\_N2\\_7.pdf](https://www.ecorfan.org/republicofperu/research_journals/Revista_de_Ciencias_de_la_Educacion/vol1num2/Revista_de_Ciencias_de_la_Educacin_V1_N2_7.pdf)
- Cristina Alonso Fernandez, A.-C. M.-O.-M. (Noviembre de 2022). Analytics for Game-Based Learning. *Learning Analytics*. Recuperado el 08 de Febrero de 2023, de [https://pubman.e-ucm.es/drafts/e-UCM\\_draft\\_374.pdf](https://pubman.e-ucm.es/drafts/e-UCM_draft_374.pdf)
- Daniel Amo, R. S. (2017). *Learning Analytics, La narracin del aprendizaje a travs de los datos*. UOC (Oberta UOC Publishing, SL). Recuperado el 02 de Febrero de 2023, de [https://books.google.es/books?hl=es&lr=&id=KJ8tEAAAQBAJ&oi=fnd&pg=PT4&dq=bibliografa+learning+analytics&ots=u7YQdEiAZq&sig=SuNYDDs0HZmZFNl\\_rhFFBXyHC4c#v=onepage&q=bibliografa%20learning%20analytics&f=false](https://books.google.es/books?hl=es&lr=&id=KJ8tEAAAQBAJ&oi=fnd&pg=PT4&dq=bibliografa+learning+analytics&ots=u7YQdEiAZq&sig=SuNYDDs0HZmZFNl_rhFFBXyHC4c#v=onepage&q=bibliografa%20learning%20analytics&f=false)
- Docker. (2023). <https://docs.docker.com/>. Recuperado el 15 de Febrero de 2023, de Docker: <https://www.docker.com/>
- Downes, A. (2016). *Github*. Recuperado el 08 de 02 de 2023, de <https://github.com/adlnet/xAPI-Spec/blob/master/xAPI.md>
- Electrnicos, I. d. (2018). *IEEE*. Recuperado el 06 de Febrero de 2023, de <https://ieeespain.org/quienes-somos/>
- Francisco Jos Garca-Pealvo, . F.-B.-E. (2017). Los MOOC: un anlisis desde una perspectiva desde la innovacin institucional universitaria. (C. U. Universitaria, Ed.) *La cuestin universitaria*(9), pgs. 117-135. Recuperado el 02 de Febrero de 2023, de <http://polired.upm.es/index.php/lacuestionuniversitaria>
- G.Almonte, M. (24 de Agosto de 2021). *Aprendizaje en red.es*. Recuperado el 01 de Febrero de 2023, de <https://aprendizajeenred.es/plataformas-lms-definicion-caracteristicas-tipos-diferencias/>

- 
- GOUIGOUX, J.-P. (2018). *Docker-Primeros pasos y puesta en práctica de una arquitectura basada en microservicios*. Recuperado el 20 de Febrero de 2023
- Government, U. S. (2023). <https://adlnet.gov/research/>. Recuperado el 06 de Febrero de 2023, de ADL: <https://adlnet.gov/research/>
- LLC, R. S. (2023). *xAPI.com*. Recuperado el 05 de Febrero de 2023, de [https://xapi.com/?utm\\_source=google&utm\\_medium=natural\\_search](https://xapi.com/?utm_source=google&utm_medium=natural_search)
- Microsoft. (2023). *Visual Studio Code*. Recuperado el 18 de Abril de 2023, de <https://code.visualstudio.com/>
- Partners, M. (2022). *Moodle*. Recuperado el 01 de Febrero de 2023, de [https://docs.moodle.org/all/es/Acerca\\_de\\_Moodle](https://docs.moodle.org/all/es/Acerca_de_Moodle)
- Phil Long, G. S. (12 de Septiembre de 2011). Penetrating the Fog: Analytics in Learning and Education. *Educause Review*. Recuperado el 01 de Febrero de 2023, de <https://er.educause.edu/-/media/files/article-downloads/erm1151.pdf>
- Pool, L. (2023). *LearningLocker*. Recuperado el 08 de Febrero de 2023, de <https://learninglocker.atlassian.net/wiki/spaces/DOCS/overview>
- Postman, I. (2023). *Postman*. Recuperado el 03 de Junio de 2023, de <https://www.postman.com/>
- SendGrid, I. (2022). *Loader*. Recuperado el 02 de Junio de 2023, de <https://loader.io/>
- Siemens, G. (10 de Octubre de 2013). Learning Analytics: The Emergence of a Discipline. *American Behavioral Scientist*. Recuperado el 01 de Febrero de 2023, de [https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUKEwjWsYGpnc79AhUDolwKHQtuA38QFnoECA4QAQ&url=https%3A%2F%2Fiu.instructure.com%2Ffiles%2F56153619%2Fdownload%3Fdownload\\_frd%3D1&usg=AOvVaw0SoQ89g\\_LH-bfB1N23oaoY](https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUKEwjWsYGpnc79AhUDolwKHQtuA38QFnoECA4QAQ&url=https%3A%2F%2Fiu.instructure.com%2Ffiles%2F56153619%2Fdownload%3Fdownload_frd%3D1&usg=AOvVaw0SoQ89g_LH-bfB1N23oaoY)
- Sommerauer, P. (2021). Augmented Reality in VET: Benefits of a qualitative and quantitative study. *Proceedings of the 54th Hawaii International Conference on System Sciences*. Recuperado el 09 de Febrero de 2022, de [https://digitalcollection.zhaw.ch/bitstream/11475/21195/3/2021\\_Sommerauer\\_Augmented-reality-in-VET.pdf](https://digitalcollection.zhaw.ch/bitstream/11475/21195/3/2021_Sommerauer_Augmented-reality-in-VET.pdf)
- Source, O. (2023). <https://docs.learninglocker.net/welcome/>. Recuperado el 13 de Febrero de 2023, de Confluence: <https://learninglocker.atlassian.net/wiki/spaces/DOCS/overview?homepagelid=360450>
- University, S. (2023). *Machine Learning*. Recuperado el 20 de Abril de 2023, de <https://www.coursera.org/>

---

## Glosario

### A

#### ADL

(Advanced Distributed Learning) es una iniciativa del Departamento de Defensa (DoD) de los Estados Unidos para implementar y desarrollar herramientas y tecnologías de aprendizaje, 16

#### Arquitectura asíncrona

Procesamiento de datos que permite que otros procesos continúen antes de que una transmisión finalice., 23

### B

#### Base de datos

Programa capaz de almacenar gran cantidad de datos, relacionados y estructurados, que pueden ser consultados rápidamente de acuerdo con las características selectivas que se deseen, 20

#### Big Data

Término empleado para definir un gran volumen de datos, 13

#### Bootstrap

Framework front-end utilizado para desarrollar aplicaciones web, 36

### D

#### Desviación típica

Medida que ofrece información sobre la dispersión media de una variable, 32

#### Diagrama de casos de uso

Descripción de las actividades que deberá realizar alguien o algo para llevar a cabo algún proceso, 21

#### Docker

Docker es una plataforma de software que le permite crear, probar e implementar aplicaciones rápidamente, 25

#### Dockerfile

Archivo de texto que contiene las instrucciones necesarias para crear una nueva imagen del contenedor, 34

### H

#### Handlebars

Procesador de plantillas, 36

#### HTML5

Estándar que sirve como referencia del software que conecta con la elaboración de páginas web en sus diferentes versiones, 14

### I

#### Índice de aprendizaje

Número por el que se multiplicarán los parámetros de la recta para realizar pequeñas aproximaciones de la recta al punto, 42

### J

#### JavaScript

Lenguaje de programación que los desarrolladores utilizan para hacer páginas web interactivas, 14

#### JSON

---

Formatación usada para estructurar datos en forma de texto y transmitirlos de un sistema a otro, como en aplicaciones cliente-servidor o en aplicaciones móviles, 19

## **L**

### Latencia

Muestra el tiempo que tardan los datos en transferirse a través de la red, 24

### Learning Analytics

Término que simboliza el análisis del aprendizaje, 13

### LMS

(Learning Manager System), plataformas de aprendizaje interactivo, 13

### LRS

(Learning Record System) , contenedor de datos que sirve como repositorio para registros generados en actividades de aprendizaje creadas con el estándar Tin Can API, 19

## **M**

### Media

Suma de un conjunto de valores dividida entre el número total de sumandos, 32

### Mediana

Número intermedio de un grupo de números, 32

### Memoria caché

Área de almacenamiento dedicada a los datos usados o solicitados con más frecuencia para su recuperación a gran velocidad, 20

### Microservicios

Enfoque arquitectónico y organizativo para el desarrollo de software donde el software está compuesto por pequeños servicios independientes que se comunican a través de API bien definidas, 25

### Moda

Valor que aparece con mayor frecuencia en un conjunto de datos, 32

### Modelo de dominio

Un modelo de dominio describe los tipos de dominio que admite una organización y sus restricciones, 20

### MongoDB

Base de datos de documentos que ofrece una gran escalabilidad y flexibilidad, así como un modelo de consultas e indexación avanzado, 24

### MOOC

(Massive Online Open Courses), cursos online masivos o abiertos, 13

### Moodle

Plataforma de aprendizaje gratuita, diseñada para crear contenidos educativos, 13

## **N**

### Nginx

Servidor web de código abierto, 23

### Nodejs

Entorno de tiempo de ejecución de JavaScript, 35

## **P**

### Python

Lenguaje de programación ampliamente utilizado en las aplicaciones web, el desarrollo de software, la ciencia de datos y el machine learning (ML, 14

## **R**

### Redis

---

(Remote Dictionary Server), es un rápido almacén de datos clave-valor en memoria de código abierto, 24

Regresión Lineal

Técnica de análisis de datos que predice el valor de datos desconocidos mediante el uso de otro valor de datos relacionado y conocido, 14

## **S**

SCORM

(Sharable Content Object Referenfe Model),trata de la creación de unidades de entrenamiento online que puedes compartir entre sistemas, 16

Servidor

Aparato informático que almacena, distribuye y suministra información, 20

Sistema operativo

Un sistema operativo es el software o programa más importante que se ejecuta en un computador, nos permite usarlo y darle órdenes para que haga lo que necesitamos, 25

## **T**

Tecnología de virtualización

El concepto de virtualización hace referencia a una tecnología que permite la ejecución de varias máquinas virtuales sobre una máquina física con el objetivo de aprovechar al máximo los recursos de un sistema y que su rendimiento sea mayor, 25

Tin Can

Estándar para las aplicaciones (software) de e-learning que estructura y permite almacenar las interacciones de una persona ante distintas actividades de aprendizaje en línea, 16

## **X**

xAPI

Especificación de software de aprendizaje electrónico que permite que el contenido de aprendizaje y los sistemas de aprendizaje se comuniquen entre si de manera que se registra y se hace el seguimiento de toda clase de experiencias de aprendizaje, 14



---

## Anexos

### I. Manual de la aplicación

#### I.I Requisitos de LMS para su uso y compatibilidad con el módulo de análisis

- La arquitectura del módulo de análisis está diseñada para obtener los datos que proporciona el LMS sobre un recurso o curso determinado. Partimos de la idea de un curso con una serie de actividades, como ver un video, leer un texto, realizar alguna actividad..., y un examen o control final. Dicha prueba se deberá puntuar numéricamente. Si la estructura del curso cambiara se deberá modificar el código del sistema para adaptarlo al LMS en concreto. Se ha probado y diseñado el sistema para un tipo de curso como el del prototipo proporcionado en la página oficial de xAPI <https://xapi.com/download-prototypes/>, en concreto un minicurso de golf, con un examen final tipo test.
- LearningLocker funcionará con cualquier tipo de LMS que cumpla con el estándar xAPI, pero **el módulo análisis habrá que adaptarlo conforme a la estructura del curso**, por lo que habrá que conocer la estructura del curso y el tipo de sentencias que enviará al LRS.
- Un buen ejemplo de uso sería con los cursos virtuales de las asignaturas de los grados de la UNED, constituidos por una serie de recursos, actividades y prácticas, finalizando con un examen final del cuatrimestre.

#### I.II Requisitos del sistema

Para poder crear todos los componentes de la arquitectura e iniciarla necesitamos:

- Una conexión de internet.
- Tener instalado el cliente Docker en el sistema, se podrá descargar e instalar de la página oficial de **Docker** <https://docs.docker.com/engine/install/> donde se encuentra disponible para varios sistemas operativos.
- Trabajaremos con **Visual Studio Code** (VSC) para el control, administración y creación del sistema, para instalar VSC se podrá descargar de su página oficial: <https://code.visualstudio.com/download>
- El siguiente paso será instalar la extensión Docker en VSC.

### I.III Creación de las imágenes

- Una vez iniciado el cliente Docker, descargamos y descomprimos la carpeta del proyecto LRS-ANALYTICS.zip.
- Procedemos a abrir la carpeta descomprimida con VSC, donde nos aparecerán las 7 subcarpetas del proyecto como se nos muestra en la Figura 19. Subcarpetas del proyecto.
- Cada servicio consta de sus respectivos archivos de configuración, si se desean modificar se recomienda visitar la página web oficial de cada uno.
- Dentro de cada carpeta dispondremos del archivo Dockerfile correspondiente que nos permitirá crear las imágenes de cada servicio. Para crear la imagen presionamos con el botón derecho en cada archivo y seleccionamos Build Image, las imágenes tardaran un tiempo en crearse dependiendo de la capacidad del equipo. El nombre de las imágenes debe coincidir con los del archivo Docker-Compose, serán:
  - o analytics: an:1.0
  - o learninglocker: ll:1.0
  - o mongo: mongo:1.0
  - o nginx: nginx:1.0
  - o redis: redis:1.0
  - o xapi: xapi:1.0

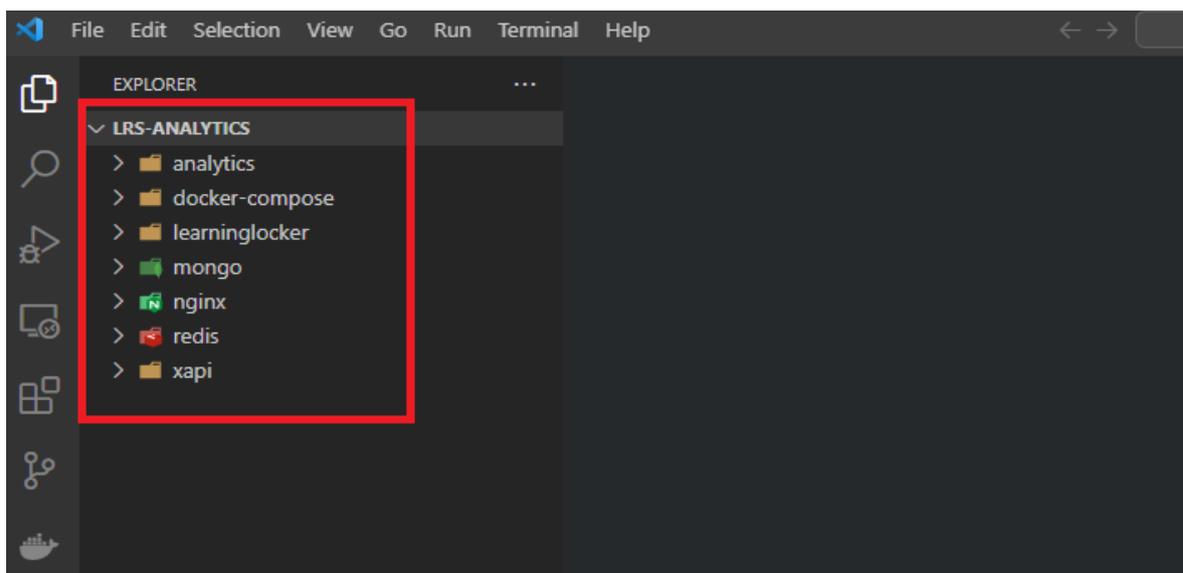


Figura 19. Subcarpetas del proyecto

- Una vez creadas las imágenes, podremos levantar el sistema con el archivo dockercompose (dentro de la carpeta docker-compose), **no funcionará si los nombres no coinciden** con los del archivo docker-compose.yml. Para el inicio del sistema seleccionaremos el archivo docker-compose.yml con el botón derecho y presionamos sobre la opción Compose Up, se crearán los contenedores establecidos y el sistema comenzara a funcionar. Podremos visualizar los contenedores activos como vemos en la Figura 20. Contenedores activos.

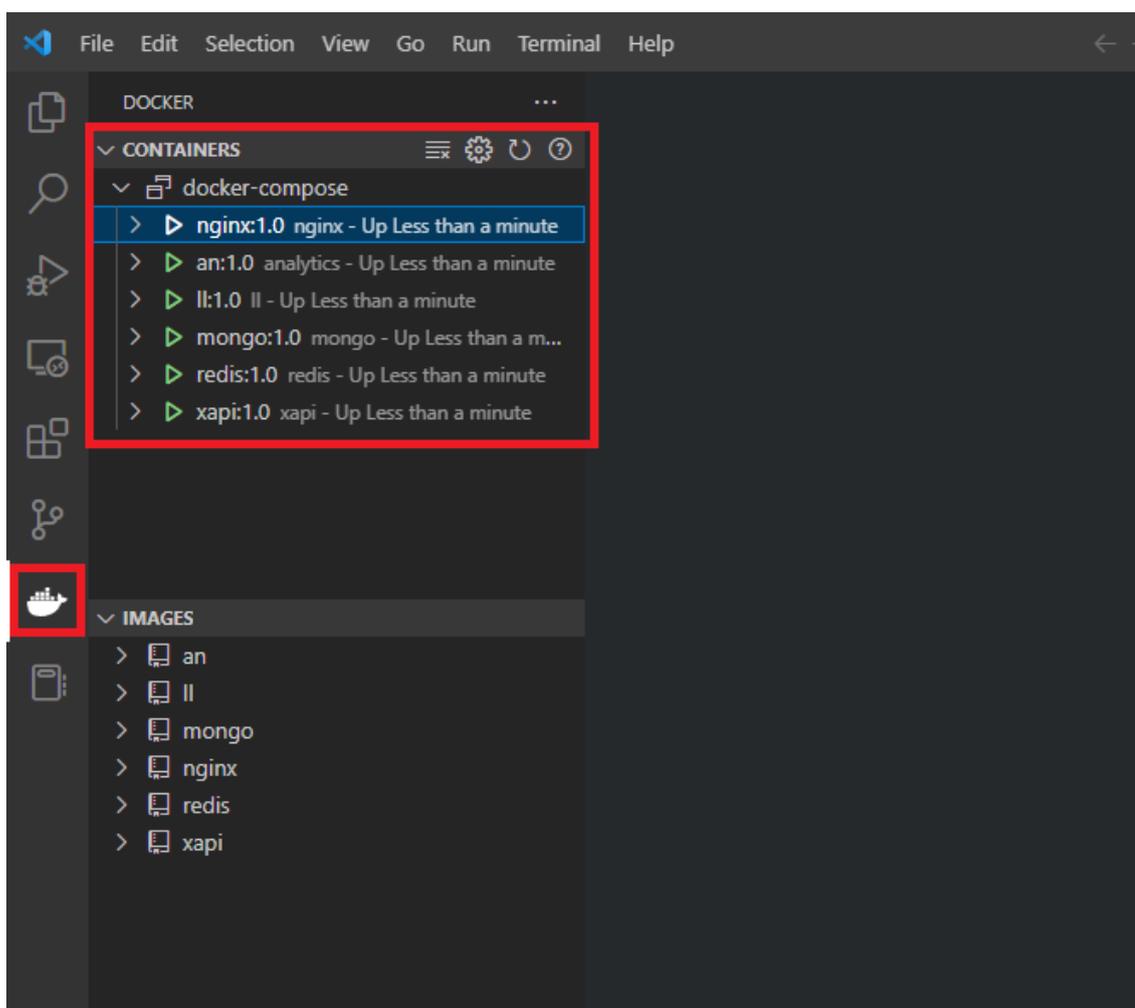


Figura 20. Contenedores activos

#### I.IV Estructura del LRS

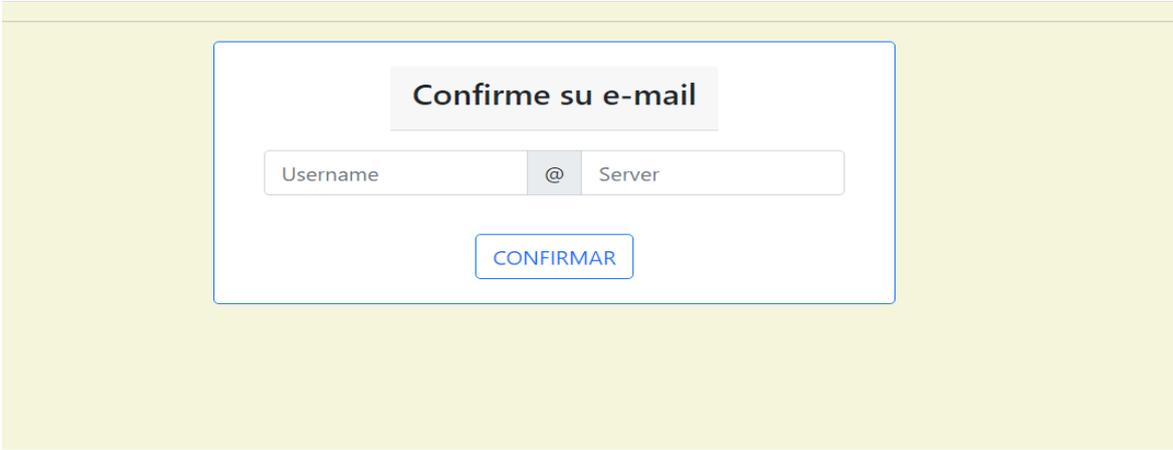
- Cada tutor tendrá una cuenta en LearningLocker donde podrá obtener un LRS por cada curso de que disponga.
- Podrá acceder a su cuenta mediante su correo electrónico y una contraseña seleccionada por el o por el administrador que por supuesto deberá conocer.
- El administrador del sistema creará cada usuario.
- Cada usuario creará los almacenes de sentencias necesarios por curso.
- El acceso al módulo Analytics se hará con el correo electrónico de la cuenta de cada usuario.
- Para la creación de un nuevo usuario en el LRS, accederemos al contenedor de LearningLocker. Una vez dentro del sistema entramos en la carpeta analytics-app con el comando **cd**. Seguidamente crearemos el usuario y su contraseña de acceso mediante el comando:  
***node cli/dist/server createSiteAdmin [email] [organisation] [password]***
- Toda la documentación de LearningLocker podemos encontrarla en el siguiente enlace:

<https://learninglocker.atlassian.net/wiki/spaces/DOCS/overview?homepageId=360450>

#### I.V Módulo Analytics

- El acceso al módulo de análisis será a través del puerto 81, del equipo donde se encuentre instalada la aplicación, con la configuración predeterminada.
- En la página principal se nos pedirá introducir el e-mail, no se podrá acceder si el usuario no se encuentra en la base de datos. Podemos observar el aspecto de esta página en la Figura 21. Página de inicio del módulo de análisis.

## ANALYTICS

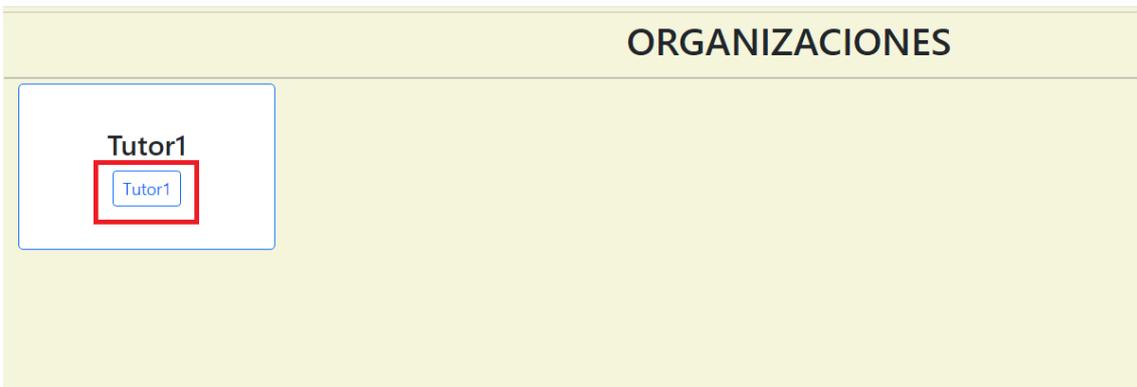


The screenshot shows a confirmation page with a light green background. At the top center, there is a grey box with the text "Confirme su e-mail". Below this, there is a white input field divided into three sections: "Username", "@", and "Server". Below the input field is a blue button with the text "CONFIRMAR".

Figura 21. Página de inicio del módulo de análisis

- Si el usuario es correcto se nos redirigirá a la siguiente vista donde podremos pulsar sobre el nombre de usuario para acceder a los cursos., como podemos observar en la Figura 22. Vista del usuario y en la Figura 23. Vista de los cursos del usuario.

## ANALYTICS



The screenshot shows a user view page with a light green background. At the top center, there is a header "ORGANIZACIONES". Below this, there is a white box containing the text "Tutor1" and a blue button with the text "Tutor1". The button is highlighted with a red border.

Figura 22. Vista del usuario

## ANALYTICS

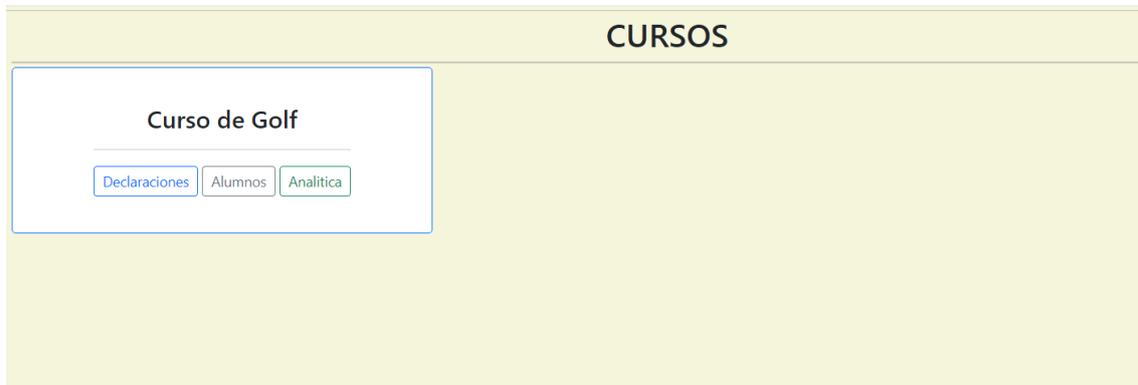


Figura 23. Vista de los cursos del usuario

- En la Figura 23. Vista de los cursos del usuario visualizamos que nos ofrecen 3 opciones:
  - o Declaraciones: donde se nos muestra, en otra vista, la información de todas las declaraciones recibidas en ese curso.
  - o Alumnos: se listarán todos los alumnos pertenecientes o matriculados en ese curso.
  - o Analítica: se visualiza una nueva vista con los dos tipos de análisis, como muestra la Figura 24. Vista de los tipos de análisis.

## ANALYTICS

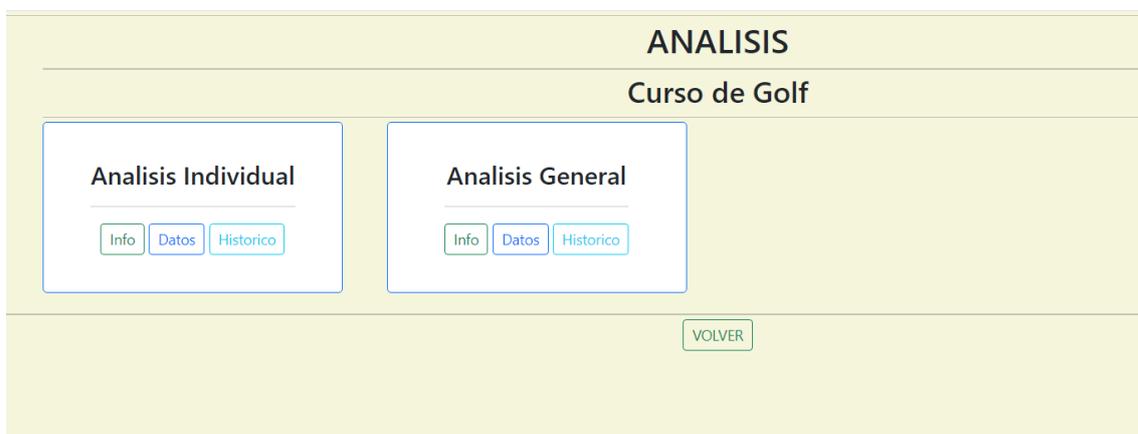


Figura 24. Vista de los tipos de análisis

- En la nueva vista se nos ofrecen tres opciones por el tipo de análisis:
  - Info: donde se nos muestra una breve información sobre el tipo de análisis.
  - Histórico: se visualizan todos los análisis realizados e información sobre éstos.
  - Datos: en el caso del análisis individual podremos elegir sobre que alumno ejecutar el análisis como se muestra en la Figura 25. Vista de la selección de alumnos para el análisis individual . En el análisis general podremos seleccionar un rango de fechas mediante un calendario, el análisis solo se realizará sobre aquellas sentencias que se hallan recibido entre esas fechas. Podemos ver un ejemplo de la vista en la Figura 26. Vista de la selección del rango de fechas en el análisis general.

## ANALYTICS

DATOS DEL ANALISIS INDIVIDUAL	
Curso de Golf	
Debe seleccionar un alumno	
Numero de Alumnos	
1	
Seleccion	
<input checked="" type="radio"/>	Alumno2 / alumno2@example.com
<input type="button" value="VOLVER"/> <input type="button" value="DATOS"/>	

Figura 25. Vista de la selección de alumnos para el análisis individual

**DATOS DEL ANALISIS GENERAL**

**Curso de Golf**

Debe seleccionar un rango de fechas

<b>Fecha mas antigua</b>	<b>Fecha mas reciente</b>
23/11/2022, 12:40:27	30/11/2022, 12:29:14

23/11/2022 | 30/11/2022 | Continuar

Oct 2022 Nov 2022

Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa	
25	26	27	28	29	30	1	30	31	1	2	3	4	5	
2	3	4	5	6	7	8	6	7	8	9	10	11	12	
9	10	11	12	13	14	15	13	14	15	16	17	18	19	
16	17	18	19	20	21	22	19	20	21	22	23	24	25	26
23	24	25	26	27	28	29	27	28	29	30	1	2	3	
30	31	1	2	3	4	5	4	5	6	7	8	9	10	

23/11/2022 - 30/11/2022    Cancel    Apply

Figura 26. Vista de la selección del rango de fechas en el análisis general

- Una vez seleccionados los datos (el alumno en el análisis individual o el rango de fechas en el general) visualizaremos los datos que serán analizados y podremos ejecutar el análisis, produciéndose otra vista con los resultados. Observamos los ejemplos de resultados en la Figura 27. Vista del resultado del análisis general y en la Figura 28. Vista del resultado del análisis individual.

## ANALYTICS

**ANALISIS GENERAL**

**Curso de Golf**

Fecha	Media	Mediana	Moda	Desviacion Típica
24/2/2023, 7:56:29	60.857142857142854	73	73	20.12816079881262

VOLVER

Figura 27. Vista del resultado del análisis general

## ANALYTICS

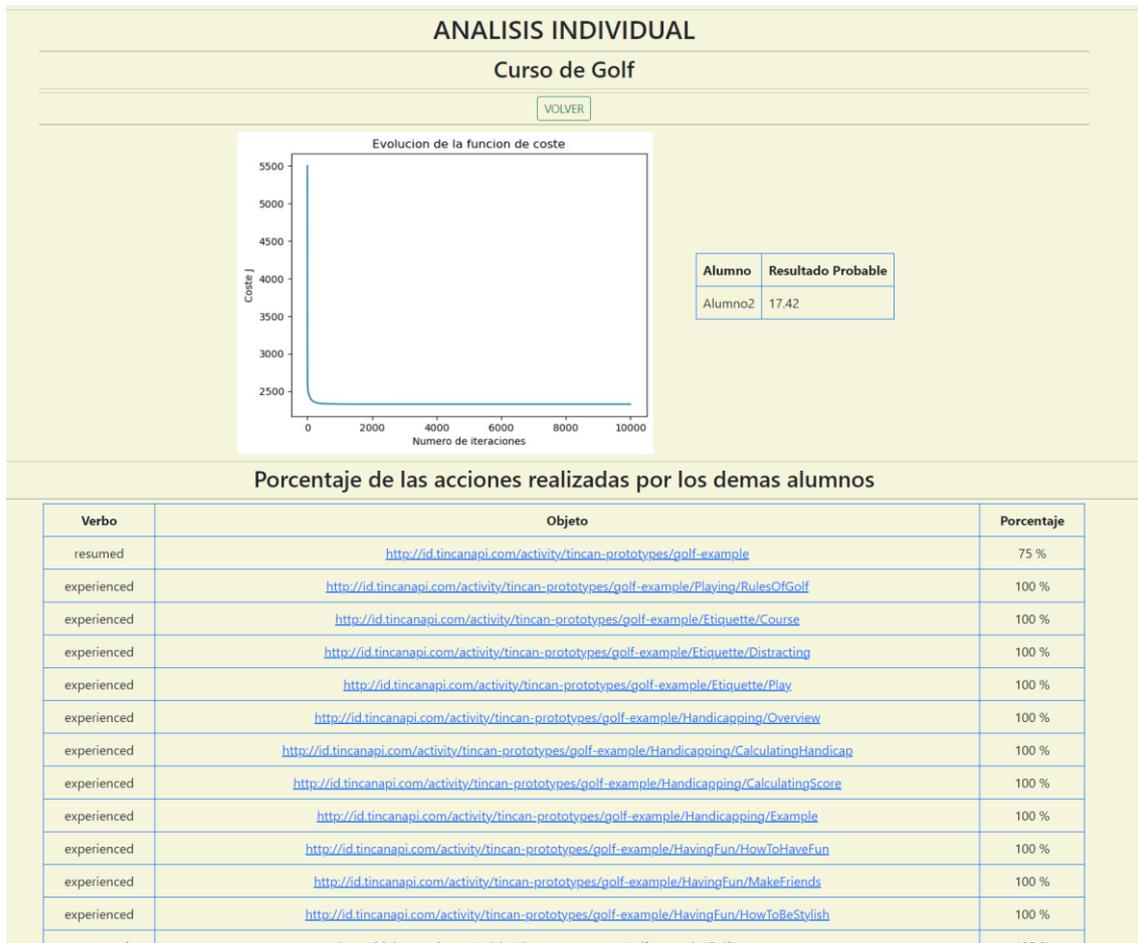


Figura 28. Vista del resultado del análisis individual

## II. Código Fuente

- El código está organizado en carpetas, cada una correspondiente a un microservicio y una carpeta para el archivo docker-compose.yml.
- La estructura del código del módulo de análisis podemos visualizarla en la Figura 29. Estructura del módulo de análisis.

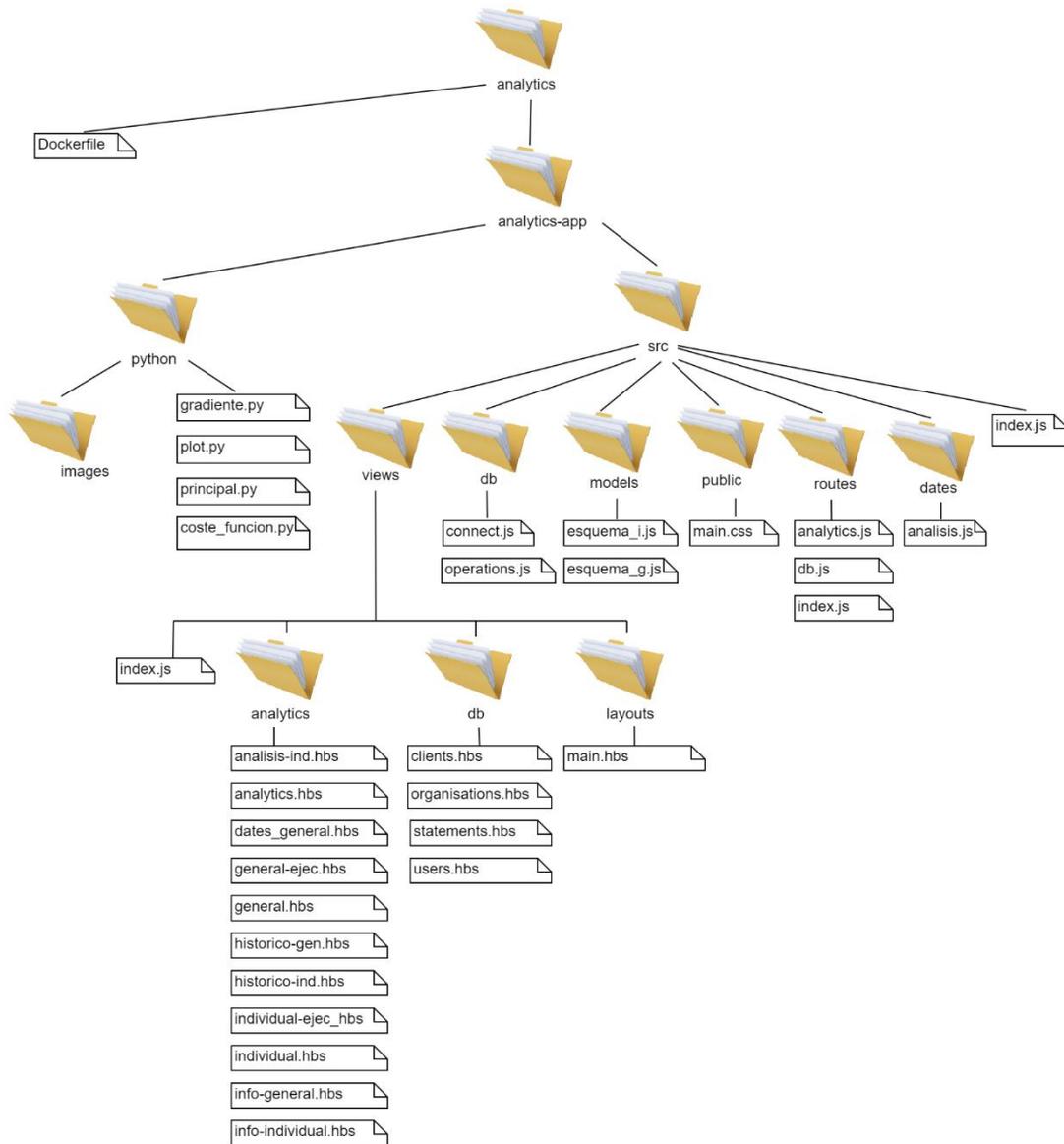


Figura 29. Estructura del módulo de análisis

- A continuación, en la Figura 30. Estructura de las carpetas de servicios, podemos ver la estructura de los archivos de los demás microservicios.

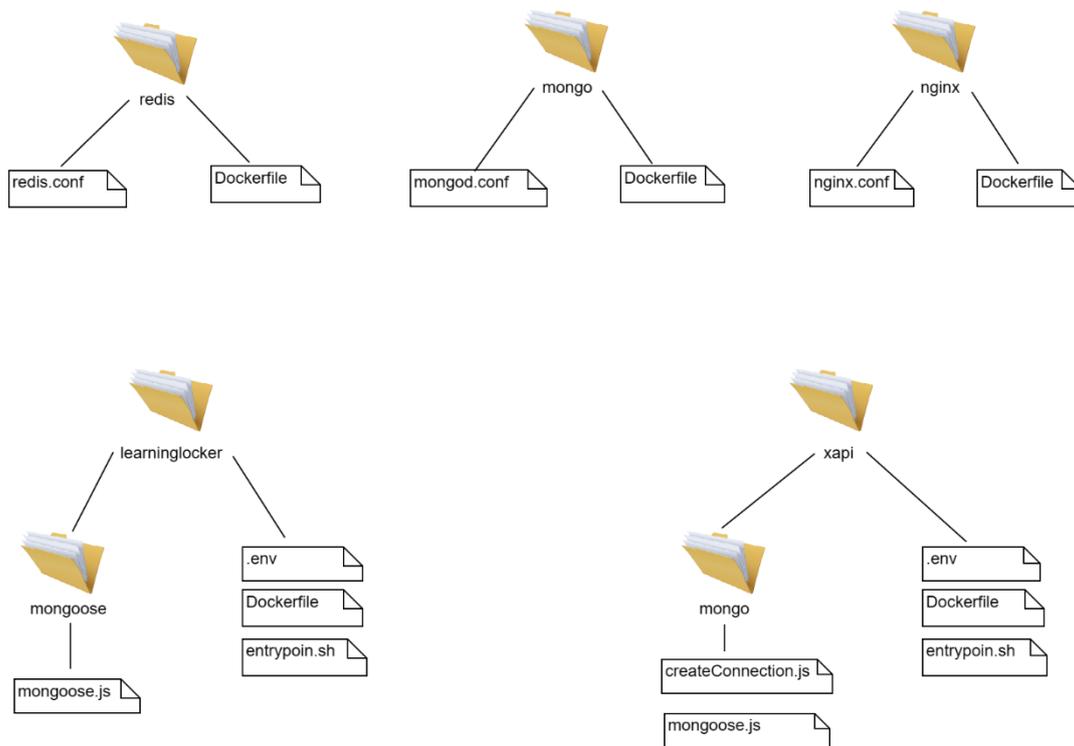


Figura 30. Estructura de las carpetas de servicios

- En la carpeta docker-compose, únicamente se encuentra el archivo docker-compose.yml necesario para lanzar la aplicación.