Máster Universitario de Investigación en Ingeniería de Software y Sistemas Informáticos

(31105151) Trabajo fin de máster en ingeniería de software y sistemas informáticos

Arquitectura software para analíticas de aprendizaje en laboratorios online

Alvaro Danilo Uyaguari Uyaguari

Ingeniero de Sistemas Informáticos por la Universidad del Azuay

Directores:

Ph.D. Rubén Heradio Gil

Catedrático de Universidad del Departamento de Ingeniería de Software y Sistemas Informáticos de la Universidad Nacional de Educación a Distancia

Ing. Daniel Galán Vicente

Catedrático de Universidad del Departamento de Ingeniería de Software y Sistemas Informáticos de la Universidad Nacional de Educación a Distancia

(2018-2019) Convocatoria extraordinaria con finalización en febrero

Máster Universitario de Investigación en Ingeniería de Software y Sistemas Informáticos

(31105151) Trabajo fin de máster en ingeniería de software y sistemas informáticos

Arquitectura software para analíticas de aprendizaje en laboratorios online

Tipo A: Trabajo específico propuesto por un profesor

Alvaro Danilo Uyaguari Uyaguari

Ingeniero de Sistemas Informáticos por la Universidad del Azuay

Directores:

Ph.D. Rubén Heradio Gil

Catedrático de Universidad del Departamento de Ingeniería de Software y Sistemas Informáticos de la Universidad Nacional de Educación a Distancia

Ing. Daniel Galán Vicente

Catedrático de Universidad del Departamento de Ingeniería de Software y Sistemas Informáticos de la Universidad Nacional de Educación a Distancia

DECLARACIÓN JURADA DE AUTORÍA DEL TRABAJO CIENTÍFICO, PARA LA DEFENSA DEL TRABAJO FIN DE MASTER

Fecha: 20/02/2019

Quién suscribe:

Autor(a): Alvaro Danilo Uyaguari Uyaguari
D.N.I/N.I.E/Pasaporte.: 0103411112

Hace constar que es la autor(a) del trabajo:

Título completo del trabajo.
Arquitectura software para analíticas de aprendizaje en laboratorios online.

En tal sentido, manifiesto la originalidad de la conceptualización del trabajo, interpretación de datos y la elaboración de las conclusiones, dejando establecido que aquellos aportes intelectuales de otros autores, se han referenciado debidamente en el texto de dicho trabajo.

DECLARACIÓN:

- ✓ Garantizo que el trabajo que remito es un documento original y no ha sido publicado, total ni parcialmente por otros autores, en soporte papel ni en formato digital.
- ✓ Certifico que he contribuido directamente al contenido intelectual de este manuscrito, a la génesis y análisis de sus datos, por lo cual estoy en condiciones de hacerme públicamente responsable de él.
- ✓ No he incurrido en fraude científico, plagio o vicios de autoría; en caso contrario, aceptaré las medidas disciplinarias sancionadoras que correspondan.

Fdo.

Impreso TFdM05_AutorPbl. Autorización de publicación y difusión del TFM para fines académicos

Autorización

Autorizo/amos a la Universidad Nacional de Educación a Distancia a difundir y utilizar, con fines académicos, no comerciales y mencionando expresamente a sus autores, tanto la memoria de este Trabajo Fin de Máster, como el código, la documentación y/o el prototipo desarrollado.

Firma del/los Autor/es

Juan del Rosal, 16
28040, Madrid

Tel: 91 398 89 10
Fax: 91 398 89 09
www.issi.uned.es

AGRADECIMIENTO

Los resultados de este proyecto, están dedicados a todas aquellas personas que, gracias a su apoyo contribuyeron para la culminación de este trabajo. Por esto agradezco muy especialmente a Rubén Heradio, Daniel Galán y Luis de la Torre, quienes a lo largo de este tiempo me brindaron todo su contingente para el desarrollo de este trabajo, el cual ha finalizado de la mejor forma. A mis padres quienes a lo largo de toda mi vida han apoyado y motivado mi formación académica y muy especialmente a mi esposa e hija quienes me han acompañado durante en este reto.

Alvaro Danilo Uyaguari Uyaguari

RESUMEN

En el presente trabajo se expone una arquitectura para la elaboración de una solución informática, la cual permite predecir el comportamiento académico de un estudiante, mediante el uso de modelos y técnicas de aprendizaje de máquina. Esta predicción se basa en la información proveniente de la interacción entre el usuario y un entorno simulado, generado por la herramienta Easy Java/JavaScript Simulations y publicado en el proyecto UNILabs de la Universidad Nacional de Educación a Distancia (UNED). La arquitectura utiliza la especificación xAPI para la recolección y el almacenamiento de los eventos y acciones realizadas en el laboratorio virtual. Este diseño utiliza también componentes de Big Data para el almacenamiento y el análisis de los flujos de datos generados. Logrando así una solución informática escalable y de bajo acoplamiento.

Palabras clave: Arquitectura de software, Software architecture, Easy Java/JavaScript Simulations , EJS, Big Data, xAPI, Business Intelligence, BI, Machine Learning, aprendizaje de máquina, e-learning.

1. Índice

Lista de figuras

Lista de tablas

1. Introducción

La utilización de servicios inteligentes para mejorar la usabilidad de una solución informática, mediante la predicción de comportamientos identificados en el usuario, se está aplicando en muchas áreas (Salud, finanzas, educación, comercio, etc.). En el campo de la educación en línea (e-learning) existen múltiples investigaciones, las cuales se enfocan en predecir la navegación, el desempeño y otros comportamientos del estudiante, en base a la información generada en los cursos en línea (Test, videos, foros, etc.).

El aporte del presente trabajo, es la creación del esquema de una solución informática para la búsqueda de patrones de comportamiento en los laboratorios virtuales; los cuales, generalmente están embebidos en un curso virtual. Estos laboratorios son utilizados para realizar prácticas en una diversidad de campos. A diferencia de un curso en línea, estos generan información fina (clic en un botón, deslizar un slider, deslizarse en un panel, etc.), lo que vuelve más complejo el análisis y el almacenamiento de dicha información, debido a la granularidad y diversidad de la información generada en estos entornos virtuales.

El objetivo generar de esta arquitectura es bosquejar esta solución informática, para identificar patrones de comportamiento y de progreso de los estudiantes que realizan prácticas en los laboratorios virtuales del proyecto UNILabs.

Para alcanzar este objetivo se requiere identificar los requisitos funcionales y no funcionales de la solución informática, establecer el esquema base del sistema y determinar las herramientas Open Source adecuadas. También para el diseño se usará estándares internacionales que permitan la interoperabilidad entre los componentes. El análisis para la selección de la arquitectura, componentes y estándares se basará en estudios exploratorios de la literatura científica, los cuales nos permiten sustentar las opciones elegidas a través de experiencias probadas y desarrolladas. De igual forma existen análisis propios que no pudieron ser localizados en la revisión de la literatura, pero que aportan significativamente a la temática desarrollada en el presente trabajo.

Nuestro trabajo consta de 3 bloques principales:

- Capítulo 2. Marco teórico. Incluye conceptos generales y necesarios para un mejor entendimiento del presente trabajo.
- Capítulo 2. Trabajos relacionados: En este capítulo se explora la literatura científica con la finalidad de identificar buenas prácticas, estándares, arquitecturas y herramientas usadas en trabajos relacionados. Esto con el objetivo de aplicar los conocimientos en el diseño planteado.
- Capítulo 3. Solución informática: Se establece el esquema de la arquitectura, los componentes y los estándares que se usarán en el diseño. Esto en base a los trabajos relacionados y a los análisis desarrollados en el presente trabajo. Además, en esta sección se establecen los requisitos funcionales y no funcionales.

- Capítulo 4. Evaluación: Se valida la arquitectura con un prototipo y con las experiencias probadas en la literatura científica.

El diseño busca una alta cohesión, bajo acoplamiento y escalabilidad. Los datos de la solución informática parten del proyecto UNILabs y de otras fuentes de datos de la Universidad Nacional de Educación a Distancia.

"UNILabs es un proyecto colaborativo, realizado entre varias universidades que comparten sus recursos de laboratorio con propósitos educativos. Estos laboratorios interactivos pueden o ser virtuales (simulaciones basadas en modelos matemáticos) o remotos, los cuales utilizan dispositivos y equipamiento de verdad para realizar experimentos reales." [23]

La metodología para diseñar la arquitectura es la siguiente:

1. Levantamiento de requisitos funcionales y no funcionales.
2. Análisis de estándares usados en entornos e-learning para la recolección de acciones o eventos generados en los entornos digitales.
3. Determinar los patrones a predecir y las variables requeridas para el o los modelos que se apliquen.
4. Diseñar la arquitectura en una representación en capas.
5. Seleccionar los componentes de software que se utilizará en cada capa.
6. Desarrollar un prototipo. [26]

Para obtener la estructura final de la solución informática. En este trabajo se contempla con el análisis de las siguientes áreas de la ingeniería de software:

- Lenguajes específicos de dominio (DSL)
  Herramienta Easy Java/JavaScript Simulations. Me permite crear un laboratorio simulado en base a un modelo matemático.
- Arquitectura orientada a servicios (SOA).
  Análisis de los estándares de interoperabilidad en entornos e-learning xAPI y Caliper (Sección 3 y 4)
- Arquitectura y componentes de un sistema de Inteligencia de negocios (Sección 3).
- Arquitectura y componentes de un sistema de Big Data.
- Arquitectura y funcionamiento de algoritmos y técnicas de aprendizaje de máquina (Machine Learning).
- También para comprender mejor el funcionamiento de Easy Java/JavaScript Simulations y los algoritmos de machine learning, es necesario conocimientos básicos de estadística, algebra lineal, ecuaciones diferencias y calculo integral.

2. Marco teórico

2.1. Laboratorios virtuales.

Un laboratorio virtual es un entorno en el que se simula, por medio de un sistema computacional, los recursos pedagógicos y tecnológicos necesarios para que el estudiante puede realizar sus actividades o prácticas, como si estuviera usando o interactuando con recursos reales.

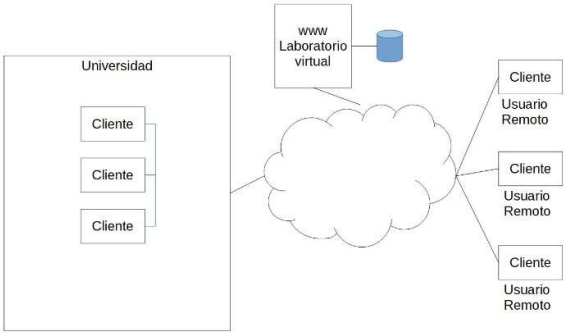

Figura 1: Laboratorio virtual en la nube.

En la figura 1 se simboliza un laboratorio virtual que puede ser accedido tanto por usuarios de la Universidad y por usuarios remotos.

La UNED en conjunto con una red de universidades formaron un proyecto de laboratorios virtuales compartidos con fines académicos (UNILabs). Estos basan su virtualización en un modelo matemático, el cual, tiene unas variables de entrada y un modelo matemático que determina el comportamiento de las variables. En la siguiente figura tenemos un ejemplo de un laboratorio virtual de UNILabs.

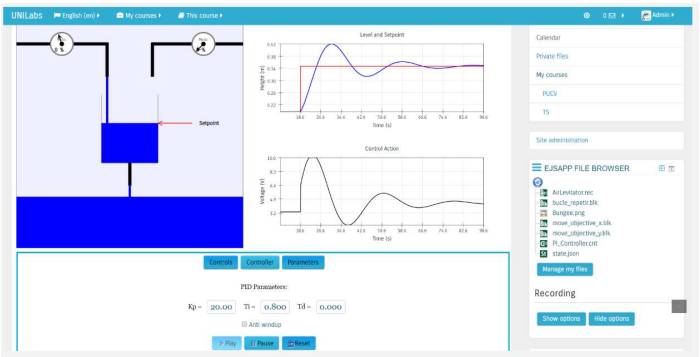

Figura 2. Laboratorio virtual del proyecto UNILabs.

2.2. Machine Learning

Machine learning es una rama de la inteligencia artificial y un mecanismo para buscar patrones y construir un algoritmo para la simulación de inteligencia en base a un proceso de aprendizaje. Este aprendizaje puede mejorar en forma proporcional a su experiencia (Ingreso de nuevos datos)[24].

El principal objetivo de machine learning es la implementación de un algoritmo de propósito general y que solvente un problema específico. Para obtener los resultados deseados, es importante que en este proceso incluya los datos requeridos por el algoritmo.

En un entorno de Big Data y machine learning, es fundamental tener siempre en cuenta el volumen de los datos, la diversidad de los tipos de datos y determinar cuan cambiantes serán los mismos, para saber aplicar las herramientas adecuadas para su procesamiento.

En forma genérica el aprendizaje automático involucra 3 procesos principales:

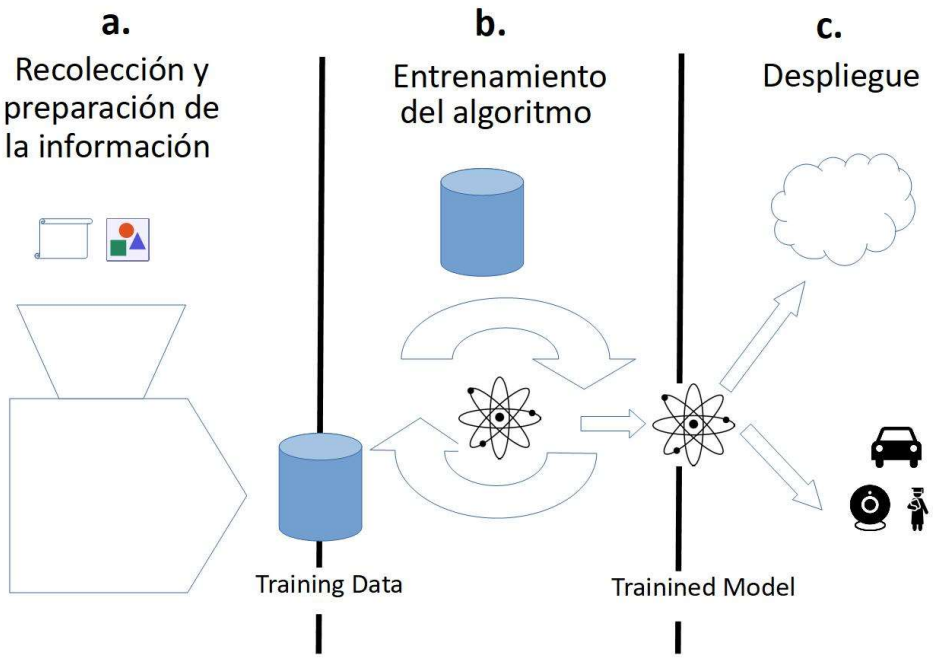

Figura 3. Procesos genéricos para un sistema de Machine Learning.

a. Recolección y preparación de la información

   En los modelos de aprendizaje de máquina supervisada se requiere variables:

   - Independientes. Son variables causales o predictoras.
   - Dependiente. Es una variable de tipo consecuencia o variable a predecir.

En este paso se realiza el proceso de filtro, preparación y selección de variables dependientes e independientes.

b. Entrenamiento del algoritmo.

   En este paso se realiza el entrenamiento del algoritmo en base a un modelo de predicción establecido.

c. Despliegue

   Despliegue de información pronosticada.

En el proceso de recolección, preparación y selección de variables predictores y a predecir, se involucra una senda conocida como Minería de datos (Datamining).

2.3. Datamining

El Datamining (DM) puede tener muchas definiciones, pero podemos resumirlas esencialmente como una tecnología que intenta ayudar a comprender el contenido de una base de datos. Básicamente, es un proceso de análisis de datos brutos con la finalidad de hacerlos comprensibles para un usuario final. Esto en términos de información relevante sobre la fuente, patrones y naturaleza de dichos datos. [28] .

Machine learning y la minería de datos usan modelos matemáticos para sus modelos; pero en minería de datos no se incluye un factor de experiencia que afina y mejora el modelo.

“Datamining is defined as the process of discovering patterns in data. The process must be automatic or (more usually) semiautomatic. The patterns discovered must be meaningful in that they lead to some advantage, usually an economic advantage. The data is invariably present in substantial quantities.” [28].

2.4. Big Data

Dig Data es definido como una colección de un gran volumen de datos que se hacen complejos de procesar a través técnicas y plataformas convencionales. En otras palabras, un grupo de datos puede ser llamado Big Data si es difícil de almacenar, procesar y visualizar usando tecnologías comunes. Es estos días, las fuentes de generación de datos están incrementándose dramáticamente, tales como los log files, streaming data, datos de sensores y otros datos que por su naturaleza pueden crecer rápidamente [25].

En los años recientes, Big Data ha estado jugando un rol vital en muchos entornos como la administración pública, investigación científica, salud, redes sociales y manejo de recursos naturales.

2.4.1. Retos de Big Data.

Vamos a revisar cuatro de los principales retos en el manejo de grandes volúmenes de información [10]:

Procesamiento. No es fácil procesar exabytes de forma completa porque resultaría demasiado largo. Una solución correcta es usar plataformas de procesamiento paralelo y algoritmos tales como MapReduce, Spark streaming o Apache Storm para obtener información oportuna y procesable.

Almacenamiento de información. Las bases de datos tradicionales no son capaces de almacenar una enorme cantidad de datos. No solamente en tamaño, sino también en variedad de tipos, tales como audio, video, texto y otros formatos. Las bases de datos NoSql son usadas para almacenar datos no estructurados y datos no relacionales. Cuando comparamos los sistemas de gestión de bases de datos relacionales RDBMS, las bases de datos NoSql son más escalables y proveen un mejor desempeño en el manejo de grandes volúmenes de datos semi estructurados y no estructurados.

Adiciona y por su naturaleza las bases de datos noSql mantienen un diseño para un crecimiento horizontal. Esto quiere decir que pueden distribuirse en un cluster de computadores, con la finalidad de aumentar su capacidad de almacenamiento y procesamiento.

Velocidad: En las recientes décadas se ha incrementado considerablemente la velocidad de las plataformas de procesamiento. Debido a que, las organizaciones no están interesadas solamente en que se busque e investigue los datos relevantes que necesitan ellos. Sino también se requiere, que esta información sea provista de forma rápida y oportuna.

Visualización de datos: La visualización de una enorme cantidad de datos implica el uso de herramientas no convencionales, debido al volumen, variedad y a la diversidad de formatos de la información. Al respecto, existe nuevas plataformas en desarrollo que tienen la capacidad de trasformar grandes cantidades de datos, en intuitivas imágenes que abstraen de forma eficiente la información, con fines de apoyo a la toma de decisiones o a visualizar patrones de comportamiento.

3. Trabajos relacionados

En la biblioteca científica existen abundantes investigaciones aplicadas en entornos reales en el contexto de:

- Arquitecturas de software para el manejo de flujos elevados de información.
- Modelos de machine learning para la predicción del desempeño, progreso y comportamiento del estudiante en entornos e-learning.
- Fuentes de información de Big Data y algoritmos de minería de datos.

Esta información recolectada es fundamental para realizar los análisis necesarios para la definición del esquema de la solución informática y también para, identificar los componentes de software y los estándares internacionales que serán usados.

Para realizar la búsqueda de artículos científicos se estableció los siguientes parámetros:

- Buscadores: Scopus, IEEE, ACM, ScienceDirect.
- Idioma: Inglés.
- La búsqueda se realizó sobre los metadatos de los artículos.
- Se estableció el siguiente contexto para la búsqueda:
    - Investigaciones de artículos que apliquen modelos de machine learning en entornos e-learning.
    - Investigaciones sobre arquitecturas y fuentes de datos para el almacenamiento y gestión de grandes volúmenes de información.
    - Algoritmos de predicción de comportamientos para grandes volúmenes de información.

La información extraída y analizada se describe en las siguientes sub secciones.

3.1. Modelos de Machine Learning en entornos e-learning.

El objetivo de esta revisión es explorar investigaciones que apliquen modelos de machine learning en procesos de predicción de comportamientos en entornos e-learning.

Los artículos localizados y seleccionados fueron los siguientes:

Nro	Artículo	Resumen de la predicción
1	The Faces of Engagement: Automatic Recognition of Student Engagement from Facial Expressions [11]	Reconocer el compromiso de un estudiante a través de sus expresiones faciales.
2	Modeling How Students Learn to Program [12]	Realizar un clúster para identificar patrones que determinan problemas en las fases del proceso de aprender a programar.

3	MLTutor: An Application of Machine Learning Algorithms for an Adaptive Web-based Information System [13]	Realiza un proceso de clasificación con el algoritmo ID3 para sugerir la navegación o hipertexto sugerido (Encuentra patrones de navegación - navegación adaptativa)
4	Prediction of Student Dropout in E-Learning Program Through the Use of Machine Learning Method [14]	Análisis de factores que ocasionan la deserción de estudiantes y el retiro de los mismos.
5	Preventing Student Dropout in Distance Learning Using Machine Learning Techniques [15]	Predicen si un estudiante va a abandonar su curso en educación a distancia e-learning.
6	Comparison of machine learning methods for intelligent tutoring systems [16]	Analiza diferentes algoritmos para aplicar machine learning en la educación (Considerando la poca cantidad de información que existe en estos entornos), predicen si va a terminar o retirarse.
7	Exploring Machine Learning Methods to Automatically Identify Students in Need of Assistance [17]	Estudiantes de programación (Predecir si necesitan asistencia y su desempeño en la primera semana)
8	WHAT MAKES A GREAT MOOC? AN INTERDISCIPLINARY ANALYSIS OF STUDENT RETENTION IN ONLINE COURSES [18]	Analiza un MOOC con diferentes técnicas, entre ellas machine learning (se recomienda métodos para evitar la salida de estudiantes)
9	Can Log Files Analysis Estimate Learners' Level of Motivation? [19]	Predecir a través de los logs la motivación de los estudiantes.
10	Use of machine learning techniques for educational proposes: a decision support system for forecasting students' grades [20]	Revisión de la literatura.
11	A Reference Model for Learning Analytics [21]	Mapeo sistemático de ambientes, interesados, objetivos y métodos en análisis de aprendizaje.
12	Automatic Detection of Learning-Centered Affective States in the Wild [22]	Reconocimiento de expresiones faciales, como por ejemplo el nivel de concentración, aburrimiento, confusión, frustración, felicidad, ansiedad

Tabla 1. Artículos que aplican modelos de machine learning para predecir comportamientos en entornos e-learning.

Se evidencio que existen investigaciones para predecir los siguientes comportamientos:

- Si el estudiante va a terminar exitosamente o a abandonar un curso.
- Si está comprometido.
- Si está aprendiendo.
- Para predecir la navegación en el entorno e-learning.
- Entre otros comportamientos.

Las fuentes de datos también fueron diversas:

- Datos de admisión y pre admisión.
- Datos históricos y actuales del desempeño del estudiante.
- Imágenes faciales.
- Logs de información (acciones realizadas en el entorno e-learning).
- Otros datos relacionados a habilidades del estudiante y datos generales.

Esta exploración nos permitió identificar los tipos de predicciones que se podría realizarse sobre un laboratorio virtual y los tipos de datos fuente que podrían generar estas predicciones.

3.2. Arquitecturas para el manejo de flujos de gran volumen de información.

En la bibliografía científica existen varias arquitecturas, muchas de ellas enmarcadas en requisitos específicos orientados a una temática y a tipos de datos específico.

El objetivo de esta revisión de la literatura es explorar las arquitecturas para el manejo de un flujo grande de datos con la finalidad de adaptar o tomar parte de estas, en el diseño de la solución informática.

Las arquitecturas exploradas son:

- Lambda architecture.
- NIST Big Data reference architecture.
- Big data for remote sensing.
- The service-on-line-data (SOLID) architecture.
- Semantic-based architecture for heterogeneus multimedial retrieval.
- Large-scale security monitoring architecture.
- Modular software architecture.
- MongoDB-based healthcare data management architecture.
- Scalable and distributed architecture for sensor data collection, storage, and analysis.
- Distributed parallel architecture for Big Data.

Cada una de estas arquitecturas se adapta a la naturaleza del problema a solucionar. Por ejemplo. Se adapta a la temática, al volumen de datos, a los tipos de datos y también al tipo de procesamiento (tiempo real o en paquetes).

Nro.	Nombre de la arquitectura	Aplicación	Ventajas	Desventajas
1	Lambda architecture	Redes sociales	Permite manejar flujos de datos online y offline.	Cuando existe mucha variabilidad en la naturaleza de los datos, no es posible el procesamiento (Tipos de datos muy variados).
2	NIST Big Data reference architecture	Aplicable a sistemas empresariales estrechamente integrados o sistemas industriales verticales acoplados libremente.	Pueden ser procesados una variedad de datos y con fuentes de datos muy variada.	Debido a que emergen rápidamente nuevas técnicas de Big Data, el proveedor del framework necesita ser actualizado constantemente.
3	Big Data architecture for remote sensing	Sensores remotos en aplicaciones satelitales.	Testeado con datos de la vida real.	Herramientas para el manejo de flujos de datos grandes no son usadas en la fase de recolección de datos y con flujos de datos en formato estructurado.
4	The Service-On-Line-Index-Data (SOLID) architecture	Modelar datos climáticos	Direcciona los principales requisitos de un Big Data semántico, en tiempo real.	Unidad gráfica de procesamiento no disponible.
5	Semantic- based architecture for heterogeneus multimedia retrieval	Procesamiento de datos multimedia heterogéneos.	Bases de datos NoSql y frameworks MapReduce son usados para mejorar la escalabilidad.	No ha sido testeado con entornos de internet reales.
6	Large-scale security monitoring architecture	Prevención y detección de intrusiones.		
7	Modular software architecture.	Procesamiento de datos geoespaciales heterogéneos grandes.	Soporta múltiples algoritmos diseñados para paradigmas como MapReduce, in memory computing or programación basada en agentes.	Solamente aplicable a un dominio específico.
8	MongoDB-based healthcare data management architecture.	Procesamiento de un gran volumen de datos de pacientes a través de sistemas distribuidos.		No se considera flujos de datos.

9	Scalable and distributed architecture for sensor data collection, storage, and analysis	Descubrir información oculta e interesante utilizando datos de la ubicación de GPS en vehículos.	Altos desempeños son logrados cuando trabajamos con sensores de datos.	Solamente el algoritmo kmeans es discutido.
10	Distributed parallel architecture for 'Big Data'	Procesar grupos de datos financieros grandes	La unidad de procesamiento gráficos posee una interfaz amigable.	No considera flujos de datos.

Tabla 2. Arquitecturas para Big Data. [10]

En la tabla 2 se sintetizan 10 arquitecturas y se describe:

- Su aplicación en la vida real.
- Ventajas.
- Desventajas.

Este análisis exploratorio me permite identificar una arquitectura adecuada a los requisitos de mi solución informática.

3.2.1. Lampda architecture

La arquitectura Lampda es una arquitectura genérica. Esto quiere decir, que no está enfocada a solucionar un problema en específico. Por lo que, esta puede ser aplicada a cualquier ámbito en que se requiera un procesamiento en lote y en tiempo real.

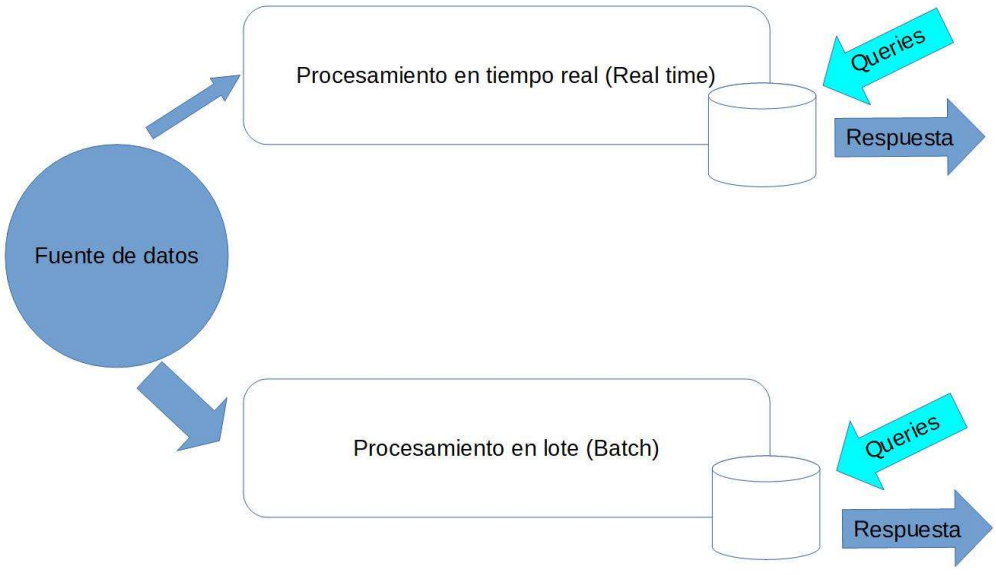

Figura 4. Esquema de la Arquitectura Lampda.

En la Fig. 4 se visualizan las tres capas de la arquitectura Lampda:

- Batch.
- Speed
- Serving layer.

Batch layer

Batch layer realiza las dos funciones principales. Llama al gestor de datos y pre computa los resultados usando un sistema de procesamiento distribuido. El gestor de datos almacena una enorme cantidad de registros del sistema. Mientras que la función de pre computo establece que el grupo de datos maestros se ejecute en paquetes de datos mediante queries con baja latencia. Las dos funciones mencionadas, mantienen un balance continuo entre lo pre computado y lo que será computado durante el tiempo de ejecución, para completar la consulta o el query.

Speed layer

La batch layer le puede tomar un considerable tiempo computar las vistas por lotes (batch view). Estas vistas podrían en algunos casos estar siendo poco oportunas; debido a que, en ocasiones, ésta tiene que esperar a que se termine el periodo de escaneo de un registro de gran tamaño. Sin embargo, el usuario continúa generando información sin ninguna pausa.

Para computar la información más reciente del usuario, el archivo de transacciones debería ser conjugado y almacenado en vistas en tiempo real (real-time view). La capa de vista en tiempo real, es usado para computar los resultados de los flujos de datos de entrada más recientes. Después, las consultas son procesadas y sus resultados son almacenados para responder a las consultas de los usuarios más recientes.

Serving layer

La Serving layer está siempre conectada con la batch layer para almacenar las batch views. En general, debido a la alta latencia de las batch views (fuera de tiempo). Esta latencia puede ser resuelta por medio de la Speed layer, porque la speed layer está siempre disponible para cualquier dato que todavía no esté en la serving layer. Las batch views deberían estar almacenadas en una base de datos distribuida que almacena batch views y hace más eficientes las consultas a las batch views.

Las batch views pueden ser modificadas bajo un control de versiones y siempre las modificaciones son provistas por la batch layer.

Este flujo analizado en los párrafos anteriores está presentado en la siguiente figura:

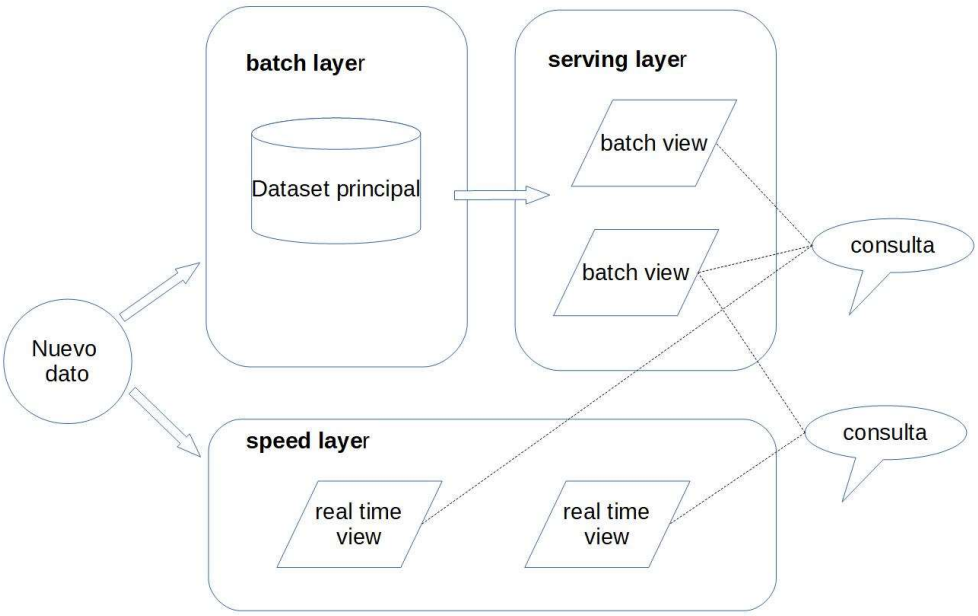

Figura 5. Flujo de datos de la arquitectura Lampda.

La arquitectura lampda puede ser una buena alternativa para aportar a la arquitectura de la solución informática, debido a que procesa en tiempo real y en paquetes.

3.3. Fuentes de información para Big Data

Esta revisión de la literatura tiene el objetivo de identificar técnicas de análisis de datos de acuerdo a la naturaleza de los datos o al tipo de datos.

Las principales fuentes de datos para Big Data localizadas en la literatura son las siguientes:

Nro.	Fuentes de Big Data	Tipo de datos	Técnica / metodología de minería de datos
1	Cuidado de la salud	Historia clínica electrónica	Procesamiento del lenguaje natural (NLP)
		Datos de imágenes médicas	Sistema de recuperación de imágenes basado en su contenido.
		Datos genéticos	Penalised logistic regression
2	Redes sociales	Datos de texto	Análisis sentimental
		Grafos	Detección de la comunidad
			Análisis de influencia social
			Investigación colaborativa

3	Circuito cerrado de televisión (CCTV) y vigilancia	Video	Labour-based surveillance system.
4	Sensores de datos	Data no estructurados	Detección de anomalías contextuales.
5	Datos generados por la máquina	Archivos log	Minería de patrones frecuentes

Tabla 3. Fuentes más comunes de Big Data. [10]

Existe una diversidad de fuentes de información que pueden crecer en volumen y variedad de tipos de datos y así ser consideradas como Big Data. Las fuentes descritas en la tabla 3, son las fuentes de datos más habituales en Big Data.

Entre las fuentes más recurrentes, se encuentra la fuente número cinco de la tabla 1 "Datos generados por la máquina". La cual hace referencia a los log files, cuyo objetivo es realizar un tracking de las acciones realizadas por los usuarios en un sistema informático. En el contexto de la gestión de conocimientos de entornos virtuales, los log files se utilizan para realizar un tracking de la interacción entre el usuario y un laboratorio virtual. La información resultante del rastreo o tracking puede ser analizada con una técnica de minería de datos o machine learning (Tabla 3, columna: técnica / metodología de minería de datos) para localizar patrones de frecuencia. Por ejemplo:

- Número de configuraciones en el laboratorio virtual.
- Número de intentos y aciertos.
- Etc.

3.4. Algoritmos para análisis con Big Data.

El Big Data es tan grande que no se almacena en la memoria principal de un sistema; al contrario, necesita procesar grandes cantidades de datos en algoritmos que se ejecutan en forma distribuida en varios nodos (Servidores).

En esta revisión de la literatura revisaremos algunos algoritmos disponibles para el análisis de grandes volúmenes de información.

- Algoritmos de clustering.
- Algoritmos de clasificación.

3.4.1. Algoritmos de clustering para Big Data.

Son usados para analizar grandes cantidades de información generada en tiempo real u offline. El principal objetivo de un algoritmo de clustering es categorizar los datos en grupos similares de acuerdo a métricas establecidas.

Los principales algoritmos de cluster pueden se clasificados en:

- Basados en particionar.
- Basados en jerarquías.
- Basados en redes.
- Basados en la densidad.
- Basados en un modelo.

Basados en particiones

Divide un grupo de datos en particiones usando una distancia para clasificar puntos, y se basa en sus similitudes.

La principal desventaja de los agrupamientos basados en particiones es que se tiene que definir un valor K (Numero de grupos). Los algoritmos más conocidos son k-means, k-modes, k-medoids.

Agrupamiento basado en jerarquías para Big Data.

También es conocido como un clustering basado en conectividad. Este no tiene un buen desempeño con grandes grupos de datos, debido a la gran cantidad de interacciones que requiere; por lo que, puede generar tiempos de respuesta demasiado elevados. Para su implementación en Big Data se debe considerar el procesamiento en paralelo.

El algoritmo tiene dos métodos principales:

- Selección de características basados en co ocurrencia.
- Modificación del lote.

Selección de la característica basado en co ocurrencia. Es usado para reducir el número de vectores de características.

Modificación del lote. Es usado para reducir el costo computacional mediante la disminución del número de iteraciones.

Normalmente usa un framework Mapreduce para el procesamiento en paralelo. Este algoritmo mejora la precisión del agrupamiento en comparación al tradicional k-means.

Agrupamiento basado en densidad.

Es usado para buscar agrupamiento de forma aleatoria, donde los grupos son marcados como regiones densamente pobladas y divididas a través de áreas de baja densidad. Este algoritmo no es adaptable para grandes cantidades de datos, pero puede aplicarse usando un framework Mapreduce.

Algunos algoritmos usados son DBSCAN, OPTICAL, DBCLASD, y DECLUDE y una de los principales usos de este algoritmo es para agrupar información geográfica.

Agrupamiento de alta dimensionalidad

Este algoritmo divide los datos localizando dimensiones por el número elevado de recurrencia de las mismas. En un documento de texto, el número de dimensiones es igual al tamaño del vocabulario del texto.

3.4.2. Algoritmos de clasificación para Big Data.

La clasificación es una de las técnicas de minería de datos que clasifica datos no estructurados en clases estructuradas y grupos. Lo cual ayuda a identificar patrones útiles para los usuarios finales.

Los principales algoritmos de clasificación son k-Nearest Neighbour (kNN), Support Vector Machine (SVM), Randow Forest and Naïve Bayes.

Árboles de decisions para Big Data

Los árboles de decisión son ampliamente usados para problemas de clasificación y regresión. Son usados para dividir grandes datos mediante la búsqueda de patrones significativos en un contexto dato.

En estos algoritmos, los datos con ruido (Incompletos, anómalos) pueden disminuir la precisión en los árboles de decisiones. Lo cual es un factor crítico en Big Data, debido a la dificultad existente para limpiar un gran volumen de datos.

Algortimo de clasificación del vecino más cercano para Big Data.

Clasifica si un elemento A pertenece a la clase B en base a un valor que está en función de densidad de probabilidad o directamente de su probabilidad.

Naive Bayes for Big Data.

Es un algoritmo supervisado de machine learning para clasificación. El teorema de Bayers se aplica en la clasificación de las características de las variables. Naive Bayes no es aplicable para grandes volúmenes de datos, pero se está investigando para usarlo con el framework Mapreduce.

A diferencia de otra clasificación en términos simples, esta clasificación "asume que la presencia o ausencia de una característica particular no está relacionada con la presencia o ausencia de cualquier otra característica, dada la clase variable". [33]

Random Forest

Este es un método de aprendizaje basado en randomizar árboles de decisiones. El algoritmo Random Forest utiliza un mayor número de árboles. Luego busca el mayormente votado mediante las clases identificadas en los árboles individuales. Este se puede usar con el apoyo de un framework que use Mapreduce y lograr un buen resultado.

Suport Vector Machine

Está basado en planos de decisión que definen límites de decisión. Un plano de decisión es usado para separar grupos de objetos que tiene diferentes miembros de una clase.

Actualmente es uno de los algoritmos de clasificación más usados para procesamiento de Big Data.

3.5. Resultados de la revisión de trabajos relacionados

Las revisiones de la literatura científica nos han ayudado a explorar información relevante, para el desarrollo de la arquitectura en lo que respecta a:

- Arquitecturas de software aplicadas para el análisis de datos y para la gestión de grandes volúmenes de información.
- Comportamientos investigados y las variables usadas para los modelos.
- Algoritmos de minería de datos y de machine learning usados en casos reales.

Esta exploración me aporta en un mejor entendimiento de las áreas estudiadas y a explorar opciones usadas en casos reales. Esto en base a investigaciones publicadas en las bibliotecas científicas.

4. Solución informática

La solución informática realizará la predicción de comportamientos en base a la recolección y al análisis del flujo de datos (log files). Estos son generados a partir de la interacción entre el usuario y los entornos virtuales.

Para realizar el diseño del esquema arquitectónico, componentes y estándares se realiza los siguientes pasos:

- Levantamiento de requisitos.
  Se establece los requisitos funcionales y no funcionales.
- Se analiza la interoperabilidad (SOA).
  Se determina el estándar internacional que se usará para el envío y recepción de los logs de datos. Existen dos estándares internaciones para el envío de logs en un entorno e-learning (xAPI y Caliper). Se analiza los dos estándares y se determina el que se adapta mejor a la solución informática.
- Comportamientos a predecir.
  Se determina el o los comportamientos y las variables necesarias para poder realizar las predicciones. Estas variables seleccionadas en base a un análisis, me permiten identificar las fuentes de datos o bases de datos que se van a requerir en la arquitectura.
- Escalabilidad de la solución.
  Se establece la arquitectura y componentes necesarios para que la solución sea escalable en dos aspectos:
    - Almacenamiento: La solución informática almacenará logs de datos y otra información relacionada al estudiante. Los logs por su naturaleza tienden a crecer de forma acelerada; por lo que se plantea una arquitectura que inicialmente pueda sostener el almacenamiento en un solo servidor. Pero de requerirse, se puede escalar a un almacenamiento distribuido (Escalabilidad horizontal).

    - Procesamiento: El procesamiento también puede ser inicialmente ejecutado en un solo servidor. Pero se plantea una solución que permita escalar a un procesamiento distribuido.

  Esta escalabilidad horizontal en almacenamiento y procesamiento se consigue con la utilización de componentes de Big Data.

- Arquitectura. Se establece un diseño en base a una arquitectura de un sistema de Inteligencia de Negocios, pero potenciado con el uso de componentes de una arquitectura de Big Data. Todo esto validado con la información recolectada y extraída en la sección de trabajos relacionados y también con los análisis previos.

4.1. Requisitos de la solución informática

4.1.1. Objetivo General de levantamiento de requisitos

Levantar los requisitos funcionales y no funcionales de la arquitectura orientada a componentes, para gestionar la recolección y el análisis de la información (log files).

4.1.2. Requisitos funcionales

Los requisitos funcionales son:

Nro	Requisito	Descripción	Prioridad (1…5)
RF1	Recolectar las interacciones entre el usuario y el laboratorio virtual.	- Los eventos de iteración pueden ser, por ejemplo: o Iniciar una simulación. o Detener una simulación. o Configurar un parámetro de la simulación. o Etc.	5
RF2	Almacenar las interacciones en una base de datos sql o no sql .	Analizar el almacenamiento en PostgreSQL	4
RF3	Generar reportes dinámicos de las interacciones entre el usuario y el laboratorio.	Los reportes dinámicos pueden ser generados por cubos de información.	3
RF4	Analizar la información, producto de la interacción entre el usuario y el laboratorio virtual.	El análisis tiene el objetivo de identificar correlaciones entre las actividades y eventos realizados por el usuario.	5
RF5	Predecir posibles comportamientos del usuario en base a la información recolectada en la plataforma tecnológica.	El proceso de predicción se basará en técnicas de aprendizaje de máquina.	3

Tabla 4. Requisitos funcionales.

4.1.3. Requisitos no funcionales

Los requisitos no funcionales son:

Nro	Requisito	Descripción	Prioridad (1…5)
RNF1	Seguridad de la información.	La información debe estar protegida contra ataques informáticos.	4
RNF2	Privacidad	La información es sensible y debe ser visible únicamente al personal autorizado	4
RNF3	Alta transaccionalidad.	El número de transacciones por segundo puede ser superior a 1000.	5
RNF3	Interoperabilidad	El sistema debe interoperar con los componentes internos y/ o sistemas externos en base a estándares de uso común en entornos e-learning.	5
RNF 5	Bajo acoplamiento	La arquitectura de la aplicación tiene que ser de bajo acoplamiento, para permitirme realizar cambios con facilidad, en los diferentes componentes.	5
RNF6	Escalabilidad	El software debe ser fácilmente escalable.	5

Tabla 5. Requisitos no funcionales.

4.2. Interoperabilidad

La aplicación de estándares de interoperabilidad y de comunicación entre componentes de software de sectores de la sociedad afines, se ha diversificado a muchas áreas. Por ejemplo. El área de la salud con HL7, finanzas, educación con xAPI, etc.

Las soluciones informáticas modernas usan arquitecturas orientadas a servicios (SOA), mediante el uso de servicios web y de un lenguaje interno de comunicación (Json o XML). Esto con la finalidad de compartir información o lograr interoperabilidad entre organizaciones afines (Ejm. Empresas de Salud, educación, etc.). En el campo de los entornos de aprendizaje e-learning, y más específicamente en la gestión de eventos generados durante la interacción con un entorno e-learning. Existen dos estándares que norman el envío de mensajes:

- xAPI.
- Caliper

4.2.1. xAPI

xAPI, es un estándar desarrollado por la empresa “The Advanced Distributed Learning Initiative”. Esta empresa se orienta al uso de mejores prácticas y el manejo de información distribuida [6]. Su objetivo es permitirnos realizar el tracking de las actividades de aprendizaje en entornos online u offline. Tiene dos componentes fundamentales:

- Statements. Actividades de aprendizaje que tienen una estructura “[actor] [verb] [object]”.
    - Actor. Agente que inicia el evento o actividad.
    - Verb. Acción realizada por el Agente.
    - Objeto. Agente o actividad que recibe la acción.
    - Adicional en el statement puede incluirse información que complemente el evento o la actividad.
- El LRS (Learning Record Store), almacén donde los statements son guardados.

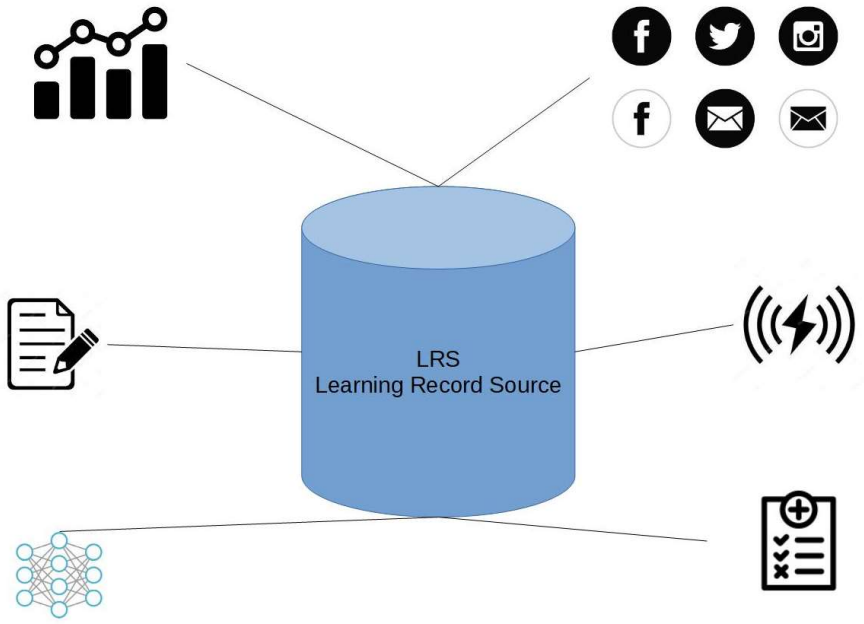

Figura 6. Learning Record Source (LRS)

En la figura 6 se representa las posibles fuentes de datos de aprendizaje almacenadas en un LRS.

El enfoque de la especificación es realizar el tracking de cualquier contenido de aprendizaje o experiencia que pueda ser entregada por la red (Figura 6). Además, para realizar la comunicación de sistemas de alerta e intervención y para obtener la información base para un modelo de predicción de comportamientos [6]. xAPI permite la interoperabilidad entre herramientas LMS, sistemas de almacenamiento LRS y otras herramientas en un entorno e-learning, a través de la especificación. La especificación contempla sentencias de lectura y de escritura en el LRS. Lo cual permite tener un feedback hacia el estudiante.

El lenguaje y vocabulario de la especificación son:

- Activity types. Catálogo de tipos de actividades.
- Attachment usages. Catálogo de tipo de archivos adjuntos.

- Extensions. Catálogo de propiedades que pueden complementar información de un statement.
- Verbs. Catálogo de palabras con las que se expresan las acciones.

Adicional la especificación permite crear:

- Profiles. Permite especificar el uso de xAPI en un dominio específico (Ejm. Acciones en la iteración con un video)
- Recipes. Especifica como xAPI soporta un caso de uso en un dominio.

Adicional xAPI soporta el uso de JavaScript como lenguaje de frontend.

4.2.2. Caliper

Caliper es una especificación elaborada por la empresa "IMS Global Learning Consortium". Este consorcio se enfoca en la creación de estándares orientados a brindar soporte a la gestión del aprendizaje [33]. La especificación se enfoca en la recolección de recursos en entornos digitales, para mejorar la visualización y entendimiento de las actividades de aprendizaje. El objetivo principal de la especificación es el permitir la creación de métricas cuantitativas de aprendizaje. Caliper se base en los "metric profiles". Los cuales modelan una actividad de aprendizaje o a su vez especifican las actividades que ayudan a facilitar el aprendizaje de dicha actividad.

Los "metric profiles" especificados son:

- Annotation Profile
- Assessment Profile
- Assignable Profile
- Forum Profile
- Grading Profile
- Media Profile
- Reading Profile
- Session Profile
- Tool Use Profile
- Basic Profile

Cada uno de estos "metric profiles" tiene un dominio de términos y conceptos.

IMS puede certificar que una aplicación informática cumple con uno o varios "metric profiles". Luego de que el interesado pase un proceso de certificación. Cada uno de las iteraciones enmarcadas en un "metric profile" tiene la siguiente estructura:

- Actor. Agente que inicializa o realiza una actividad.
- Action. evento que define una o más acciones que se pueden realizar en un dominio de una actividad.
- Object. una entidad o alguna cosa que participa en una actividad relacionada al proceso de aprendizaje.

En el siguiente cuadro se sintetiza las principales características de las especificaciones xAPI y Caliper.

Características	xAPI	Caliper
Empresa fabricante	The Advanced Distributed Learning Initiative	IMS Global Learning Consortium
Enfoque de empresa	Mejores prácticas y el manejo de información distribuida (Empresa apegada a la industria)	Estandarización
Store	Learning record store	Event store
Modelos de datos	Actor, Verb, Object	Actor, Action, Object
Enfoque	Tracking a cualquier contenido de aprendizaje o experiencia que puede ser entregado por la red. Sistemas de alerta e intervención. Modelo de predicción de datos.	Crea métricas cuantitativas de aprendizaje.
Lenguaje	Activity Scripting Language	Event Scripting Language
Interoperabilidad	Si	Si
Endpoint	Read/Write	Read
Vocabularies	Profiles and Recipes Profile: Como usar xAPI en un dominio específico Recipe: Como xAPI soporta un especificado caso de uso en un dominio Ejm. El número de interacciones Área de conocimiento (conceptos, relaciones y reglas)	Metric profile Metric profile: Describe una actividad de aprendizaje o una actividad que facilita el aprendizaje Los perfiles están compuestos de uno o más eventos Cada evento especifica un vocabulario

Certificaciones	Certifica herramientas y logros	Certifica herramientas y logros
Soporta java script	Si	Si
Adopters	https://xapi.com/adopters/	

Tabla 6. Características de xAPI y Caliper

Como parte del análisis también se investigó también los componentes de software disponibles para cada una de las especificaciones.

Especificación	Tipo de almacén de datos	App	Requisitos	Fecha del último commit	Observación
Caliper	Event store	Cube	MongoDB	hace 4 años	No esta soportado
Caliper	Event store	MongoDB		ago-18	
Caliper, xAPI	Even Store, Learning Record Store, Werehouse	OpenLRW	MongoDB	5 meses	
Caliper, xAPI	Iteroperabilty	BadgesCoP		hace 4 años	
Caliper	Extensiones			Abril de 2018	
xAPI	Learning Resource Store	SCORM Cloud		No especifica	Tril free
xAPI	Learning Resource Store	Learning Locker	MongoDB	hace 1 mes	Open Source

Tabla 7. Software y librerías para xAPI y Caliper.

En conclusión y de acuerdo a el objetivo de la solución informática, la especificación que más se adapta es xAPI. Esto debido a que, la especificación xAPI permite realizar un tracking de todo tipo de eventos y actividades, sin discriminar que estos eventos estén relacionados o no a un Metric Profile. Como es el caso de Caliper, el cual determina que todos sus eventos tienen necesariamente que pertenecer a una acción de aprendizaje.

La flexibilidad de eventos y actividades que pueden ser rastreadas de acuerdo a la especificación xAPI, se adapta a una recolección de comportamientos e interacciones muy variadas. Lo cual permitirá tener una fuente amplia de datos para realizar un proceso de análisis y predicción de comportamientos. Adicional a lo anterior, también existe una amplia variedad de herramientas que se adaptan la especificación xAPI, las cuales cubren varios campos, como son:

- Diversidad de clientes para la recolección de información (JavaScript, Python, Java, etc)
- Librerías para manejo de información offline
- Desarrollos externos para el almacenamiento de registros (Learning Record Store).
- Herramientas para la generación dinámica de reportes.

4.3. Definición de comportamientos a predecir.

En las secciones anteriores se ha especificado los requisitos funcionales, no funcionales, y los estándares que se utilizarán.

El objetivo final de la solución informática, es la predicción de comportamientos de los estudiantes en base al análisis de los eventos y actividades realizadas en los laboratorios virtuales, por parte del estudiante. En este proceso de análisis se aplicará modelos de machine learning, con el objetivo de generalizar los comportamientos en base a la información suministrada por la recolección de eventos y acciones de los estudiantes.

En esta sección se detallan los pasos que se establecieron para realizar la selección de los comportamientos a predecir y las variables predictoras. Este análisis parte de la revisión de la literatura “3.1. Modelos de Machine Learning en entornos e-learning.”

4.3.1. Comportamientos a predecir

En la revisión de la literatura se estableció que existe una gran cantidad de investigaciones en la aplicación de modelos de predicción en entornos e-learning. En dos contextos:

- Mejorar el proceso de aprendizaje.
  Predecir si un estudiante va a terminar el curso, si va a aprender a programar, si necesita ayuda, etc.
- Mejorar la usabilidad de la herramienta.
  Predecir la navegación del estudiante.

En el presente trabajo nos vamos a enfocar en mejorar el proceso de aprendizaje. Por lo tanto, necesitamos determinar qué factores pueden ser determinantes para mejorar el aprendizaje. En los artículos recolectados en la sección “3. Revisión de trabajos relacionados” se evidencia que, muchas de las predicciones se basan en forma directa o indirecta en identificar dos factores fundamentales:

- El progreso del estudiante.
- El compromiso del estudiante.

La identificación de un grado o índice de ocurrencia de estos dos factores nos permite deducir si el estudiante probablemente:

- Termine la práctica.
- Si le interesa el tema.
- Si necesita ayuda, etc.

Una solución puede ser presentada, asignando un índica de compromiso y de progreso del estudiante como se puede ver en la figura 4.

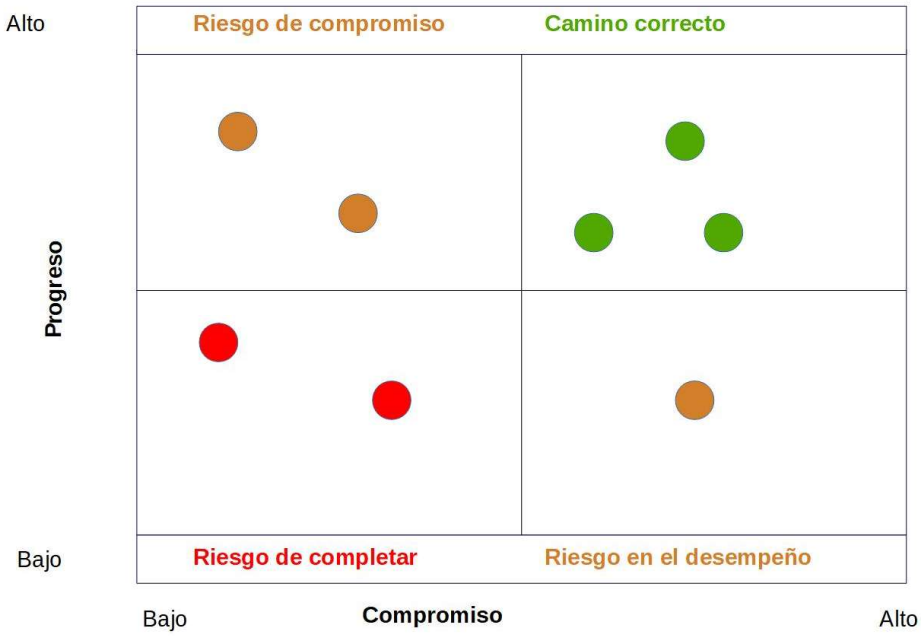

Figura 7. Nivel de compromiso y progreso del estudiante. [32]

En la Fig. 7, cada punto representa a un estudiante y su ubicación depende de su puntuación alcanzada en progreso y compromiso. En el eje de las "X" tenemos el compromiso y en el eje de las "Y" el progreso. El color asignado a cada estudiante (Punto) depende del cuadrante en el que se encuentre ubicado.

- Rojo, tiene riesgo de completar y riesgo de compromiso.
- Naranja, tiene solo riesgo de compromiso o solo riesgo de desempeño.
- Verde, no tiene riesgo de compromiso, ni riesgo de desempeño.

Figura 8. Posibles acciones a tomar de acuerdo al nivel de progreso y compromiso. [32]

En complemento a la Figura 8, se especifica las posibles acciones que se pueden requerir de acuerdo al progreso y compromiso del estudiante.

En síntesis, en esta subsección se determina que los factores que se van a predecir son:

- El compromiso del estudiante.
- El progreso del estudiante.

4.3.2. Variables independientes (Predictoras)

En un modelo de entrenamiento de aprendizaje de máquina es necesario básicamente que se agregue al modelo, un número de variables representativas y relacionadas al comportamiento o a la variable a predecir.

Existen una gran variedad de factores o variables que podría deducir o estar relacionados al desempeño y compromiso del estudiante. En la siguiente figura se detalla un grupo de factores que podría ayudar a determinar si el estudiante va a finalizar o no el curso.

- Factores internos.
- Factores externos.
- Características del estudiante.
- Habilidades del estudiante.

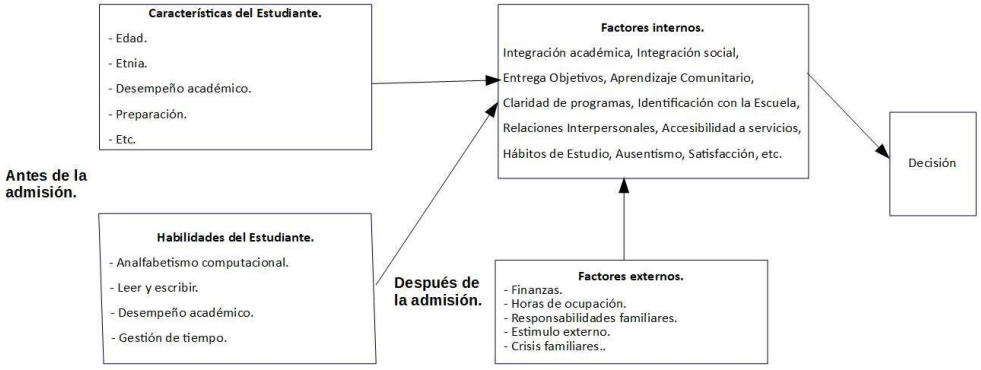

Figura 9. Modelo para la retención de un estudiante [14].

En la figura 9 se determinan variables o información relacionada al estudiante antes de la admisión y después de la admisión; las cuales pueden influir en una decisión de abandonar o no un curso.

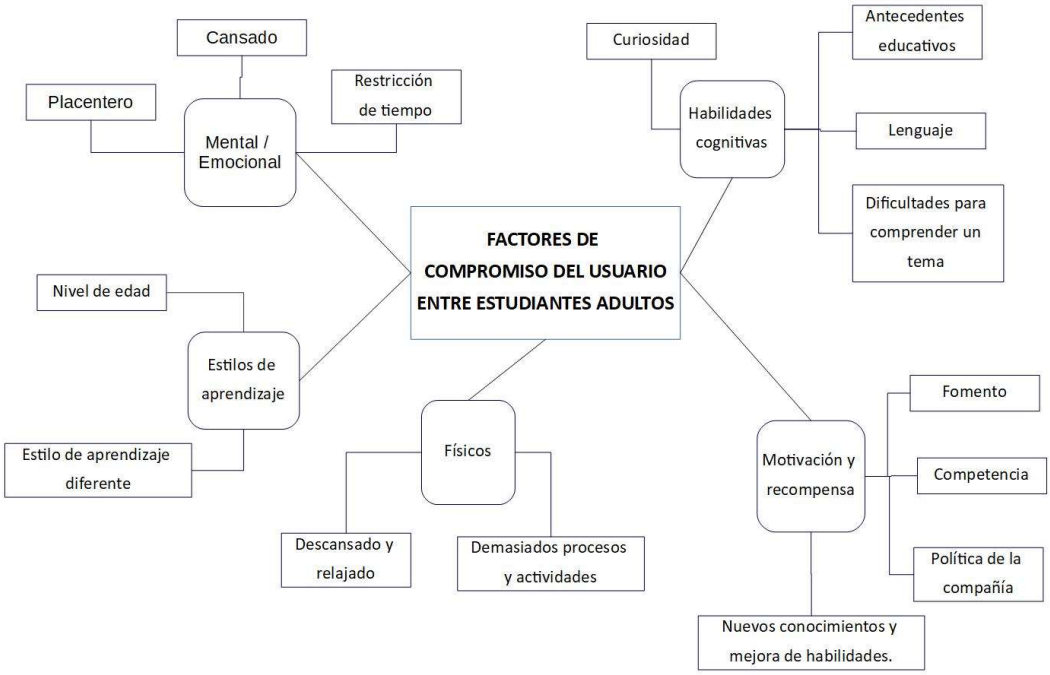

Figura 10. Red de análisis para factores de compromiso en entornos e-learning. [15]

En la Figura 10 se especifican factores que pueden influir en el compromiso de un estudiante adulto. Agrupados por factores emocionales, físicos, habilidades cognitivas,

etc. Adicional a los factores expuestos en la Figura 9 y 10, también existen otros factores más finos que están relacionados a la interacción del estudiante con su entorno virtual:

- Parametriza el laboratorio.
- Inicia y termina la práctica.
- Manipula controles (Sliders, labels, etc)
- Etc.

Esta información fina, la podemos considerar como un tracking de los usuarios, durante el uso de los laboratorios virtuales.

Para definir qué información será rastreada, se toma como fuente de investigación un estudio de predicción de la motivación de los estudiantes en base a sus Logs [19].

En esta investigación se evalúan los siguientes atributos:

Atributo	Descripción
Id del usuario	Un identificador único para cada usuario.
Desempeño	Porcentaje de test resueltos correctamente (Test correctos sobre el número total de test).
Tiempo dedicado a la lectura	Tiempo de espera en páginas (Calcular como la suma de tiempo de espera en cada página accedida).
Número de páginas	Número de página accedidas.
Tiempo en realizar un test	Tiempo de espera para realizar un test (Calculado como tiempo de espera en cada test).
Motivación	Compromiso / sin compromiso

Tabla 8. Atributos que podrían ser incluidos en el análisis. [19]

En el cuadro anterior se especifican factores que pueden ser identificados en una interacción con el entorno e-learning, los cuales son fácilmente cuantificables y podría ayudar a definir si existe o no motivación del estudiante en base a su log de acciones.

Sin compromiso	Compromiso
Acceso a páginas mediante un clic (Consecutivos eventos de acceso a páginas) con una permanencia corta en	Acceso a páginas mediante un clic (Consecutivos eventos de acceso a páginas) con un promedio de al menos 60 segundos en cada página.

cada página (menos de 20 segundos).	
Mucho tiempo de espera en cada página / test (sobre los 10 minutos).	Tiempo de espera razonable por cada página / test (entre 1 y 10 minutos).
Logouts automáticos de parte del sistema debido a la inactividad (30 minutos).	Ausencia de logouts automáticos.

Tabla 9. Reglas para ayudar a determinar la motivación del estudiante [19].

En el cuadro anterior se identifica comportamientos que podría determinar si el estudiante está comprometido con la práctica o únicamente asiste por cumplir una actividad y probablemente no la termine.

En el entorno que nos desempeñamos (laboratorios virtuales), generalmente se realizarán prácticas. Esto disminuye la oportunidad de rastrear comportamientos en lo que respecta a interacciones que evidencien de manera certera ciertos comportamientos del estudiante (Ejm. No se tiene test o foros, sino netamente una interacción con componentes generados por JavaScript).

La poca información certera para predecir el compromiso o el progreso del estudiante aumenta la complejidad del análisis. Esto debido a que es necesario tener más variables independientes para mejorar la predicción. Sin embargo, se puede aumentar el número de variables independientes mediante la inclusión de información que se encuentra en otras bases de datos de la UNED. Estas bases de datos son el Sistema académico y el Sistema de Gestión del Aprendizaje. La información a complementar es la siguiente:

- Características de los estudiantes.
- Habilidades de los estudiantes.
- Algunos otros factores relacionados a los entornos virtuales.

A continuación, se describen los factores que probablemente tengan mayor incidencia para determinar un nivel de progreso y compromiso del estudiante al realizar una práctica en un laboratorio virtual.

Métrica	Eventos a monitorear	Factor al que puede influir (Progreso/Compromiso)	Detalle de compromiso
Acierta la simulación (Si/No).	Login simulación Acertar simulación	Progreso/Compromiso	Behavioral engagement
Termina la simulación (Termina correcta o incorrectamente)	Login simulación Inicia simulación Termina la simulación	Progreso/Compromiso	Behavioral engagement
Número de veces que acierta la simulación.	Login simulación Acertar simulación	Progreso/Compromiso	Behavioral engagement
Configura el laboratorio (20 % de configuraciones posibles)	Configurar un elemento de la simulación.	Progreso/Compromiso	Behavioral engagement
Inicia Pausa la simulación	Login simulación Inicia la simulación Pausa la simulación	Compromiso	Behavioral engagement
Permanece activo al menos 4 minutos en el laboratorio, si no está en ejecución (Inactivo o revisando el progreso)	Login simulación Movimiento de mouse en el laboratorio. (Penalizar)	Compromiso	Behavioral engagement
Guarda datos del experimento	Almacenar datos (Parámetros, gráficas, resultados, etc)	Compromiso/Progreso	Behavioral engagement
Información inicial para el perfil	**Detalle**	**Factor al que puede influir (Progreso/Compromiso)**	**Detalle de compromiso**
Ubicación geográfica	Ubicación geográfica	Compromiso	Tipo de estudiante
Temática de laboratorio (Física, Electricidad, Electromecánica, Calculo)	Metadatos del curso	Compromiso	Tipo de estudiante

Carrera (Electrónica, Físico, Electromecánico, etc.)	Extrae de una base de datos relacional	Compromiso	Tipo de estudiante
Primera o segunda matrícula		Compromiso	Emotional engagement
Edad		Compromiso	Tipo de estudiante
Conoce el lenguaje de interface del laboratorio	Lenguaje(Inglés, español, etc)	Compromiso	Cognitive engagement
Tiene recompensa	Existe una motivación para llevar a cabo el laboratorio	Compromiso	Emotional engagement
Etnia		Compromiso	Tipo de estudiante
Género		Compromiso	Tipo de estudiante
Desempeño académico en la materia		Compromiso	Cognitive engagement
Maneja sistemas informáticos		Compromiso	Cognitive engagement
Estado civil		Compromiso	Tipo de estudiante
Trabaja		Compromiso	Tipo de estudiante

Tabla 10. Eventos y acciones a ser rastreadas.

La tabla contiene las siguientes columnas:

- Métrica. Especifica el factor que puede ser medido para determinar un grado de compromiso o/y progreso.
- Eventos a monitorear. Especifica los eventos de la interfaz de usuario que serán registrados en la base de datos.
- Factor al que puede influir. Determina si la métrica influye en el progreso o/y en el compromiso del estudiante.

- Detalle del compromiso. Especifica el tipo de compromiso que puede ser medido:
    - Compromiso de comportamiento. Es determinado de acuerdo a las acciones realizadas por el estudiante.
    - Compromiso cognitivo. Es determinado de acuerdo al interés o predisposición a adquirir nuevos conocimientos.
    - Compromiso emocional. Es determinado por eventos que pueden motivar al estudiante (Ejm. Si fuera su última matrícula).

Hemos determinado factores relacionados al estudiante en su pre y pos admisión, factores finos mediante el rastreo de las acciones realizadas por el estudiante durante la ejecución de una práctica en el laboratorio virtual. Finalmente, es necesario sumar todas estas variables correlacionadas a un proceso de compromiso y/o progreso de un estudiante al realizar la práctica de laboratorio, como se puede apreciar en la Tabla 10.

Como se mencionó, en un proceso de análisis de información con machine laerning, es importante contar con una considerable cantidad de información y variables. Cada uno de los factores o variables de la Tabla 10 podría ser considerado como una variable independiente en un análisis de predicción. En el proceso de machine learning existe el pre procesamiento "2. Marco teórico", en el que se mantendrá o eliminará las variables de acuerdo su relación existente con la variable a predecir. Por lo que es importante incluir variables y no descartarlas sin un previo análisis.

4.4. Escalabilidad de la solución informática

Conforme la educación se va automatizando y digitalizando, nuestras actividades online generan una gran cantidad de datos. El volumen de los datos puede ser tan masivo, que puede sobrepasar las capacidades de un servidor debido a su crecimiento. Por lo tanto, puede ser necesario usar un cluster de servidores para paralelizar el procesamiento y/o almacenamiento.

El rastreo de los log files en un entorno educativo e-learning, conlleva el tracking de todas las interacciones del usuario y en todos los dispositivos. Por ejemplo. Iniciar la simulación, pausar la simulación, configurar un parámetro del laboratorio, pulsar un botón, subir o bajar una slider, etc.

Por su naturaleza, los log files tienden a crecer en forma vertiginosa, debido al flujo de datos(logs) y a la necesidad de mantener un histórico de dichos eventos. Esto, con la finalidad de poder identificar patrones de comportamiento, navegación o interacción con la plataforma tecnológica, y así mejorar la experiencia de navegación, la usabilidad de la solución o para identificar de forma oportuna los posibles problemas en el proceso de aprendizaje.

En esta subsección se analizan dos aspectos:

- El almacenamiento.

El almacenamiento debe permitirme una escalabilidad horizontal. Esto considerando que los logs de información pueden sobrepasar la capacidad de almacenamiento de un servidor y podría a futuro, ser necesario almacenarlos en un cluster de servidores (Escalabilidad horizontal).

- El procesamiento.

  La escalabilidad horizontal en el procesamiento implica que, la arquitectura seleccionada pueda en este momento o a futuro distribuir el procesamiento en un cluster de servidores.

4.4.1. Almacenamiento (log files)

El almacenamiento de acciones y eventos durante el uso de los laboratorios virtuales y remotos, implica un crecimiento acelerado de datos resultantes del tracking de las acciones realizadas por los estudiantes en el entorno virtual.

Los eventos y las acciones son almacenadas en formato Json, de acuerdo a la especificación de xAPI. En base a este requisito de la especificación, se evaluó dos herramientas para el almacenamiento de los logs de información en un Learning Record Store (LRS).

- TimescaleDB. Para adaptarlo como un Learning Record Store(LRS).
- LearningLocker. Que es un LRS y usa MongoDB como almacén de datos

LearningLocker se seleccionó como una opción, debido a que utiliza como base de datos MongoDB; la cual es una base de datos NoSQL que de forma nativa utiliza el almacenamiento de archivos en formato Json y permite además una escalabilidad horizontal.

TimescaleDB fue seleccionada como una opción. Debido que usa como base de datos PostgreSQL (Un requisito del usuario - RF2) y TimescaleDB como framework adaptado a PostgreSQL. El cual también permite un crecimiento horizontal.

4.4.1.1. TimescaleDB y PostgreSQL (Library)

TimescaleDB es una herramienta Open Source que se extiende del gestor de base de datos open-source PostgreSQL, para crear una tabla transaccional, dividida en pedazos pequeños de tablas organizados por una serie de tiempo (Hipertablas). TimescaleDB se encarga en forma transparente de gestionar la creación, modificación y consultas sobre las hipertablas. [30]

El proceso de partición de una tabla grande en tablas pequeñas organizadas por una serie de tiempo, optimiza el rendimiento de la base de datos, debido a que la base de datos nunca va a tener una tabla demasiado grande. Adicional a esto, el software optimiza los algoritmos para el acceso y la escritura en los pedazos de tablas.

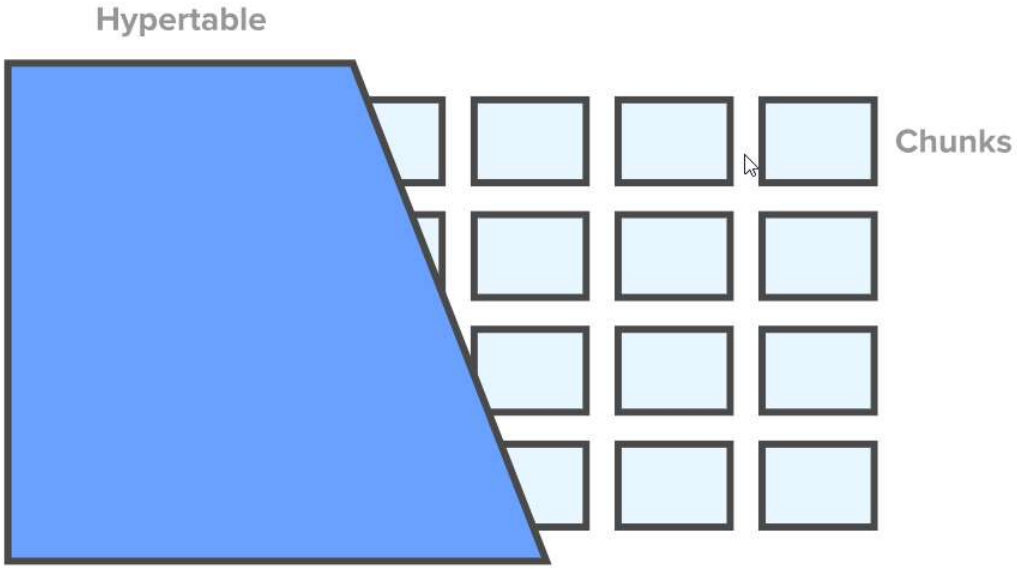

Figura 11. Particiones de una hipertabla. [30]

En la Fig. 11 se exhibe un ejemplo de la división de una tabla grande (HiperTable) en tablas pequeñas (Ckunks).

El componente TimescaleDB se instala como una librería adjunta a la base de datos PostgreSQL. Con respecto a los archivos Json de la especificación xAPI, PostgreSQL permite el almacenamiento de archivos Json en forma nativa. Sin embargo, el manejo de los archivos en formato Json es complejo y no muy intuitivo (No sigue una notación estándar ni intuitiva de creación, inserción, modificación y eliminación). El desempeño en lo que concierne al manejo de consultas sobre archivos Json (Insert, update, select), la librería tiene un muy buen desempeño.

En relación al LRS, este tiene que ser integrado a el TimescaleDB, como una librería o a través de un desarrollo personalizado. Actualmente existe una librería para crear un LRS integrado a PostgreSQL (ADL LRS) [35]. Pero este no tiene actualizaciones recientes en su repositorio (Última actualización hace 2 años). Por lo que probablemente no pueda funcionar correctamente con versiones actuales de PostgreSQL.

TimescaleDB es una excelente herramienta para el manejo de información estructurada y con escalabilidad horizontal. Sin embargo, no contiene de forma nativa un LRS.

4.4.1.2. Learning locker con Mongo DB

Learning locker es una herramienta informática que funciona como un servicio. Trabaja como un LRS, y adicional a esto, permite realizar un análisis de la información almacenada a través de reportes dinámicos y dashboards. [32] LearningLocker usa MongoDB para almacenar la información no estructurada en formato Json. Puede escalar en forma horizontal de forma nativa. Esto mediante la partición de colecciones

(Equivalente a tablas en bases de datos estructuradas) en tamaños predeterminados. [31]

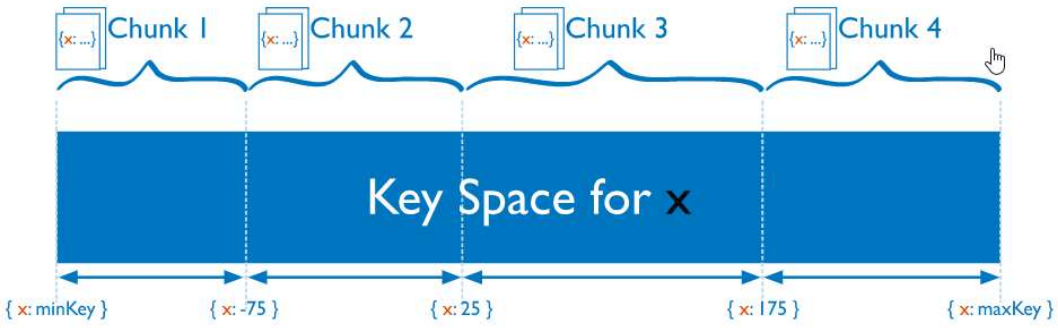

Figura 12. Particiones en una base de datos MongoDB. [31]

La gestión automática de particiones de MongoDB le permite manejar colecciones muy grandes (Fig. 12), mediante la división de las colecciones en segmentos pequeños (Chunks). Los mismos que pueden ser distribuidos y gestionados de forma trasparente para el usuario. Esto le brinda al usuario la impresión de que está trabajando sobre una sola colección.

Respecto al manejo de archivos Json, MongoDB de forma nativa utiliza este formato estándar para el almacenamiento. Para la gestión de estos archivos, el gestor brinda al usuario un lenguaje para realizar transacciones y consultas sobre la información almacenada, de una forma fácil e intuitiva. El rendimiento de estas consultas depende de la correcta configuración de la base de datos. Además, esta base también permite un crecimiento horizontal. Lo que quiere decir que la base de datos puede ser distribuida en un cluster de servidores y aumentarse equipos al cluster de acuerdo a la demanda.

El componente LearningLocker integra el LRS para almacenar los datos (archivos Json), y adicional un módulo de reportaría dinámica y de generación de dashboards.

A continuación, se sintetiza las principales características de TimescaleDB y LearningLocker:

Características importantes con respecto a el manejo de Json y pensando en un crecimiento horizontal	TimescaleDB – PostgreSQL	LearningLocker-MongoDB
Autonomía	Librería, si se tiene ya instalado PostgreSQL	Componente / servicio
Desempeño del funcionando como clúster	El desempeño depende de una correcta configuración.	El desempeño depende de una correcta configuración.
Escalabilidad horizontal	Si	Si
Almacena archivos Json	Si	Si
Desempeño con la gestión de archivos Json	- "PostgreSQL has por performance out of the box Requires a decent amount of tuning to get good performance out of it - Does not scale well with large number of connections pgBouncer is a must - Combines ACID compliance with schemaless JSON - Queries not really intuitive" [34]	- "MongoDB has decent performance out of the box. - Unstable throughput and latency - Scale well with large number of connections - Strong horizontal scalability - Throughput bug is annoying - MongoDB rolling upgrades are ridiculously easy - Developer friendly - easy to use!" [34]
LRS	Si existe un software que funcione como LRS sobre PostgreSQL (ADL LRS).	Incluido en el componente
Licencia	Apache (Open Source)	GPL V3 (Open Source)

Tabla 11. Características de TimescaleDB y Learning Locker

La sección de "Desempeño con la gestión de archivos Json", de la Tabla 11, se sustenta en un análisis realizado por una empresa consultora Alemana "High Performance JSON PostgreSQL vs. MongoDB" [34], en la que se analiza el desempeño de las bases de datos MongoDB y PostgreSQL para la gestión de archivos Json.

Ventajas y desventajas	TimescaleDB	LearningLocker-MongoDB
Ventajas	Se adapta a los componentes actuales (PostgreSQL), como una librería, sin ser necesario extender o incorporar nuevos componentes,	Se agregar un componente que funciona como data warehouse y reportería dinámica a la arquitectura.

	pensando en un proceso de distribución.	
	Combina datos estructurados y en formato Json.	Facilita el manejo de Queries dinámicos (Facilidad de uso)
	Más velocidad en Insert, Update y Select	Incorpora un LRS
	Escalabilidad horizontal	Escalabilidad horizontal
Desventajas	Para el análisis de datos, en base a una reportaría dinámica, se tendría que implementar una solución informática.	Se incorporaría como un componente externo a la solución informática actual.
	Es necesario incorporar un LRS.	No se recomienda si se requiere un uso transaccional.

Tabla 12. Ventajas y desventajas de TimescaleDB y Learning Locker.

En base a las características, ventajas y desventajas (Tabla 11 y 12) se concluye que las dos herramientas alcanzan un buen desempeño con la gestión de grandes volúmenes de información. También pueden escalar horizontalmente en base la partición de sus tablas y/o colecciones. Pero la herramienta LearningLocker es la que mejor se adapta a la arquitectura de la solución informática, debido a que:

- LearningLocker de forma nativa integra un LRS.
  TimescaleDB por su lado, necesita incorporar un LRS. Pero el mismo esta descontinuado desde hace 2 años.
- LearningLocker además incorpora un gestor de reportaría dinámica, un gestor de dashboards y la gestión de seguridades.
- Learninglocker usa MongoDB para el almacenamiento de archivos Json según la especificación xAPI, pero a diferencia de PostgreSQL. MongoDB utiliza un lenguaje intuitivo y fácil para realizar transacciones y consultas sobre estos archivos Json almacenados.

4.4.2. Procesamiento

En esta sección buscaremos alternativas arquitectónicas y herramientas para que el o los procesamientos pueda ser ejecutado en una solo máquina o de forma distribuida. El procesamiento distribuido involucra que una tarea global pueda ser cumplida a través de varias tareas hijas, ejecutadas en máquinas separadas.

En la siguiente figura se puede identificar el proceso de creación de un almacén de datos:

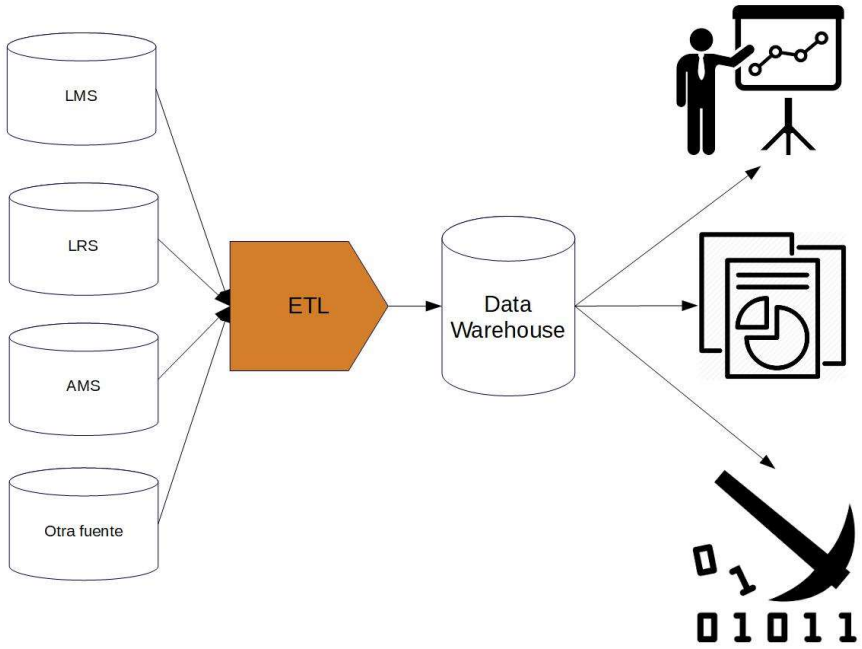

Figura 13. Proceso de creación de un Data Warehouse.

La Fig. 13 presenta el proceso de creación de un almacén de datos y consta de los siguientes componentes:

- Learning Management System (LMS)
  Sistema para la gestión de cursos en entornos e-learning.
- Academic Manager System (AMS)
  Sistema para la gestión de recursos académicos (Curriculum, alumnos, etc.)
- Learning Record Store (LRS)
  Almacén de datos de acciones capturadas con la especificación xAPI
- Extract, Transform and Load (ETL)
  Permite extraer información de múltiples fuentes de datos, limpiar, afinar o transformar esta información y finalmente cargarla en otra base de datos.
- Data Warehouse

Es un almacén de datos que integra información proveniente de múltiples fuentes en un modelo estrella o copo de nieve.

En el proceso de creación de un almacén de datos, el componente principal es el ETL. Este es el responsable de conectarse a las múltiples fuentes de información, procesar la información mediante Jobs (trabajos) y finalmente construir el almacén de datos en otra base de datos. Para la creación del ETL se puede utilizar herramientas de procesamiento en lote o en tiempo real. Para desarrollar el presente trabajo necesitamos un ETL de procesamiento en tiempo real y con una arquitectura escalable como se puede apreciar en el siguiente esquema.

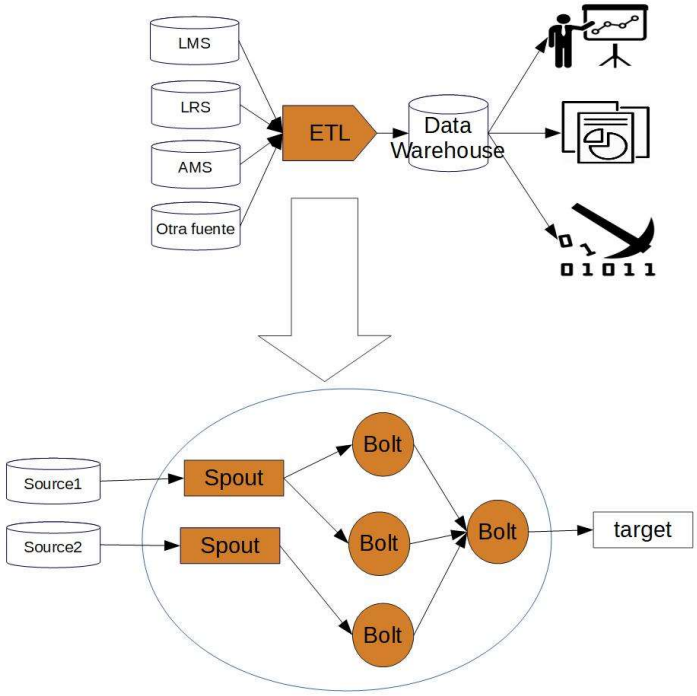

Figura 14. Componente ETL escalable por medio de Apache Storm.

En la Fig. 14 se incorpora el componente Apache Storm, el cual está compuesto de:

- Spout: Conectores a fuentes de información externas.
- Bolt: Componentes que realizar tareas programadas en forma individual o en colaboración con otros componentes Bolt. Estos realizan una tarea colaborativa formando una topología de red.

En nuestra arquitectura, el Apache Storm realiza el proceso de Extraer, Transformar y Cargar (ETL). Este es escalable debido a que cada Bolt puede procesar su tarea en máquinas separadas físicamente. Esto se conoce como procesamiento distribuido.

4.5. Arquitectura en capas de una solución Big Data

El esquema general de la solución informática se representará por medio de capas. Estas encapsulan sus componentes y funcionan de manera autónoma y desacoplada. Lo cual permite obtener una solución escalable y altamente desacoplada.

El esquema de la solución informática parte del análisis de una arquitectura para Inteligencia de Negocios (BI) y otra para el manejo de grandes volúmenes de información (Big Data). Esto debido a que, la arquitectura requiere incorporar componentes y capas de las dos arquitecturas mencionadas. Luego de esto, se procederá a incorporar los componentes de software que formarán parte de cada una de estas capas sugeridas.

4.5.1. Arquitectura de una solución informática para Inteligencia de Negocios.

Figura 15. Arquitectura genérica de una solución de Inteligencia de Negocios.

Las principales capas en esta arquitectura son la siguientes:

Fuentes de datos.

Aquí se encuentra el conjunto de almacenes de datos que contienen la información relevante para el negocio. Estos pueden contener información estructurada o no estructura y además pueden pertenecer a fuentes de datos internas y o externas.

Extracción de datos

Esta capa contiene el software para extraer datos de cualquier fuente de información estructurada o no estructurada. Luego de extraer los datos, los transforma y los carga en otra base de datos multidimensional.

Almacén de datos (Data Warehouse)

Aquí se encuentra el conocido como (On-Line Analytical Processing) OLAP Server. El cual mantiene los datos de una forma estructurada y multidimensional para facilitar el proceso de exploración y análisis de los datos. Este modelo multidimensional puede ser usualmente un modelo estrella o un modelo copo de nieve.

Visualización, análisis y cubos

Es la capa en la que el cliente o usuario final usa para realizar reportes, consultas o minería de datos sobre la información almacenada en el Data Warehouse.

Seguridad y gobierno

El gobierno de los datos, gestiona de forma efectiva los riegos, visibilidad, confianza y control de la información de los clientes y de la empresa.

4.5.2. Arquitectura de una solución informática para la gestión de grandes volúmenes de información (Big Data).

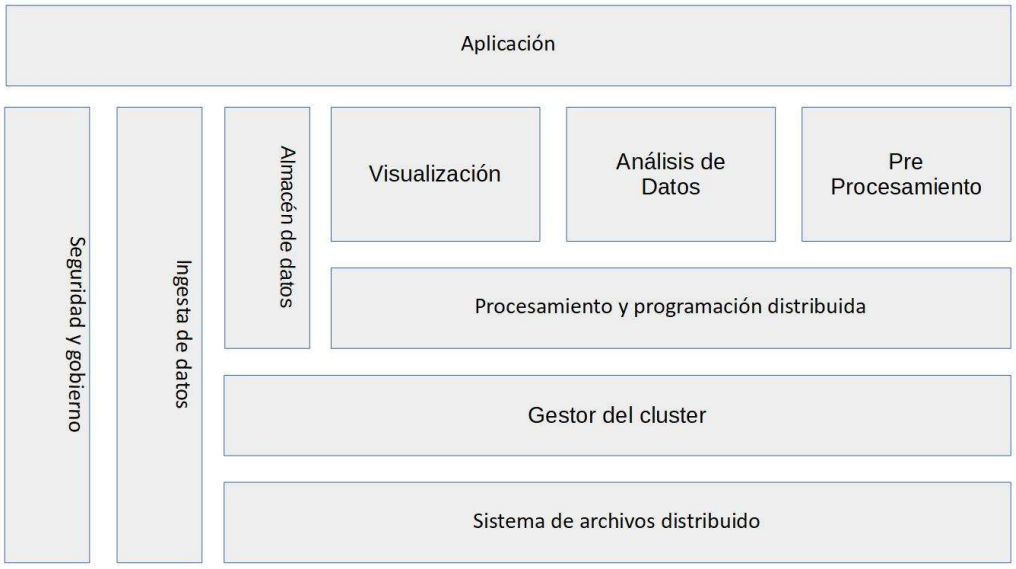

Figura 16. Arquitectura genérica de una solución Big Data. [9]

A continuación, se detallan los principales componentes de una arquitectura genérica Big Data:

Sistema de archivos distribuidos.

En la capa de sistemas de archivos distribuidos (DFS) reside el nivel más bajo de la arquitectura, con el fin de almacenar y gestionar grandes volúmenes de datos mediante múltiples nodos hijos. Los DFS son diseñados para cumplir un trabajo mediante la distribución a sus nodos esclavos.

Cluster management

Este componente es el responsable de desplegar, calendarizar y orquestar los trabajos (jobs) en una red de nodos. Esta distribución la realiza para construir una arquitectura escalable y confiable.

Programación y procesamiento de datos distribuidos

Big Data necesita procesar una significativa cantidad de lotes de datos en tiempo real; por lo que es necesario constituir un eficiente, escalable y distribuido modelo de programación y procesamiento soportado en varios nodos, de una forma oportuna y resistente a fallos.

Almacén de Datos

En Big Data es necesario el almacenamiento de un enorme volumen de datos que se encuentran en una diversidad de formatos; por lo que estos sistemas de almacenamiento deben ser veloces y tolerantes a errores. Este requisito de almacenamiento es solventando por las bases de datos No Sql, las cuales existen de diferentes tipos como por ejemplo basadas en documentos, grafos, clave valor, etc.

Visualización

Las herramientas de visualización tienen la habilidad de presentar una cantidad alta de datos en ilustraciones o gráficos y ayudan a entender conceptos difíciles de identificar.

Análisis de datos

En esta capa se encuentran las herramientas para realizar el proceso de desarrollar un modelo analítico. Este se realiza mediante la revisión de datos crudos y con la finalidad de inferir conocimiento o buscar patrones de comportamiento. Esto mediante el soporte de herramientas y algoritmos de minería de datos o aprendizaje de máquina.

Pre procesamiento de datos

El objetivo de esta capa es localizar ruidos en los datos, datos perdidos, datos incompletos en grandes volúmenes de datos. Este proceso de limpieza de datos determina el éxito de los análisis de datos.

Seguridad y gobierno

El gobierno de los datos, gestiona de forma efectiva los riegos, visibilidad, confianza y control de la información de los clientes y de la empresa.

Data digestión

Las herramientas de Data ingestión ayuda en la transferencia de datos de varias fuentes de datos externas hacia sistemas internos. Esto lo realiza de una manera eficiente y efectiva. También proveen un método confiable y tolerante a fallos de distribución de datos a través de una arquitectura orientada a componentes.

Aplicación

Esta capa provee un alto nivel de abstracción para implementar aplicaciones específicas de Big Data; o, presentar el análisis de resultados producidos en capas inferiores hacia el usuario final.

En la siguiente tabla se presentan las alternativas de herramientas para cada una de las capas:

Nro.	Capa	Software
1	Sistema de archivos distribuidos.	Hadoop Distributed File System (HDFS), Baidu File System, Gluster File System, Tachyon.
2	Gestor del Cluster.	Apache Mesos, Apache Aurora, Genie- Netflix y Apache Helix.
3	Programación y procesamiento de datos distribuidos.	Apache Spark y Hadoop usa el modelo de programación MapReduce y Apache Storm, Heron Gora.
4	Almacén de Datos.	MongoDB, Tera, RethinkDB, HBase, Voldemort, HiperTable, y RQLite…
5	Visualización	Kibana, Airpal…
6	Análisis de datos	Apache Calcite, Apache Drill, Tensor flow, PhoyonML, Cascalog y Scalding.
7	Pre procesamiento de datos.	CKAN, Apache Griffin y Data Cleaner.
8	Seguridad y gobierno.	Apache Atlas, Apache Ranger, HiBench y Apache Zookeeper.
9	Data digestión	Apache Kafka, Sqoop, Pulsar, Gobblin y Suro.
10	Aplicación	Apache Spark, Apache Cassandra, Apache Kafka y Akka.
11	Herramientas de soporte.	Apache OpenWhisk, Apache River y Apache Solr

Tabla 13. Software para los diferentes componentes de una arquitectura Big Data.

Para la implementación de la arquitectura se podría aplicar cualquiera de estas herramientas. Pero con un análisis previo que determine el cumplimiento de requisitos y la afinidad tecnológica con los demás componentes de la solución informática.

4.5.3. Arquitectura resultante de la combinación entre BI y Big Data.

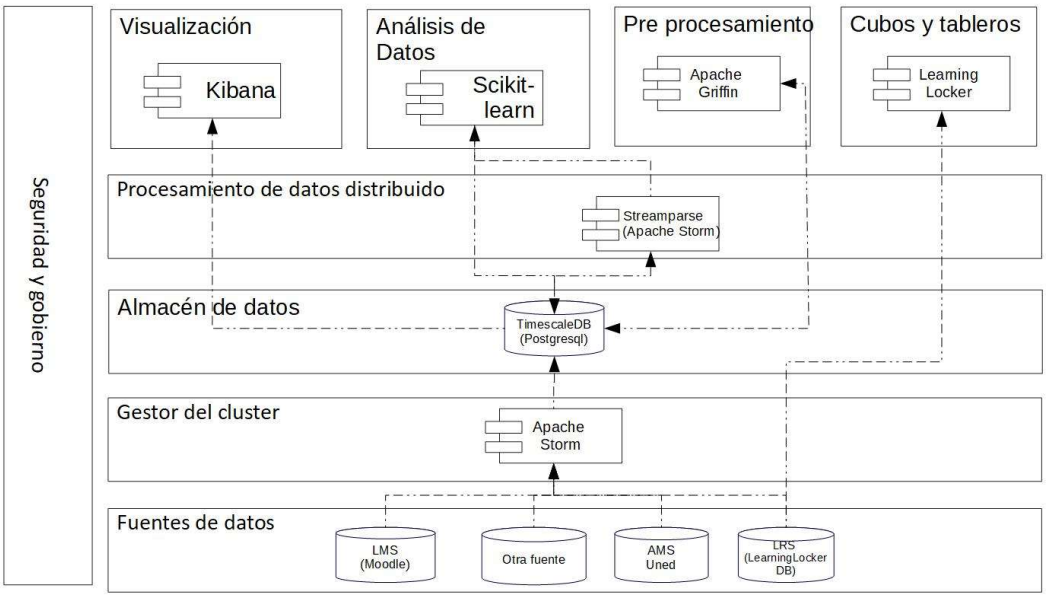

Figura 17. Arquitectura de la solución informática.

En la figura 17 se presentan los siguientes componentes:

4.5.3.1. Fuentes de datos

En esta capa se encuentran las bases de datos que contienen la información identificada en la tabla número 10. En esta se encuentran todas variables de entrada del modelo de predicción. Y las mismas pueden ser ubicadas en las siguientes bases de datos:

- El Learning Record System (LRS). Es la base de datos que almacena los eventos de los entornos virtuales. Estos son generados en base a la interacción entre el usuario y los laboratorios virtuales. El formato en el que se transmiten los eventos está normado de acuerdo a la especificación xAPI.
- El Learning Manager Systems (LMS). Es la base de datos de la plataforma virtual e-learning (Moodle), la cual está personalizada y adaptada a los requisitos del proyecto UNILabs [32]. El mismo que almacena todos los recursos y resultados del aprendizaje impartidos en este LMS.
- Academic Manager Systems (LMS). Es la base de datos que almacena la información académica de la UNED, en la que encontramos datos del docente, de los alumnos, las calificaciones, los períodos, etc.
- Otra fuente. Representa alguna fuente o fuentes de datos que se requieran para completar información que no pudo ser localizada en las bases de datos mencionadas.

4.5.2.2. Gestor del cluster

La herramienta Apache Storm realiza el proceso de una herramienta ETL. Extraer los datos de las fuentes de información, transformar los datos a un modelo multidimensional y cargar esta data en una base de datos (TimescaleDB). Además, Apache Storm puede ejecutar los Jobs de transformación en un simple nodo o en nodos distribuidos (Procesamiento distribuido). Esto le convierte en componente que puede escalar su procesamiento de acuerdo a la demanda existente.

4.5.2.3. Almacén de datos.

La capa almacén de datos tiene la herramienta TimescaleDB. Esta es una librería incluida en la base de datos PostgreSQL, con la finalidad de que, las tablas de grandes dimensiones puedan ser divididas en pedazos pequeños de acuerdo a series de tiempo. Estos pedazos de una misma tabla pueden estar distribuidos en varios nodos (Escalabilidad horizontal de almacenamiento). Pero son gestionados por TimescaleDB como una única tabla de una base de datos PostgreSQL.

TimescaleDB tiene dos funciones:

- Almacenar uno o varios modelos estrella o copo de nieve.
- Almacenar tablas que contendrán todas las variables necesarias para la ejecución de modelo de predicción.

4.5.2.4. Procesamiento de datos distribuido.

Esta capa es la responsable de localizar patrones de comportamiento. Esta tarea la realiza por medio de Jobs ubicados en la herramienta Apache Storm y la librería Streamparse. Se adiciona esta librería para que Apache Storm pueda procesar segmentos de código escritos en el lenguaje Python. Estos scripts tienen que estar escritos en lenguaje Python. Debido a que los Jobs que se tiene que ejecutar en esta capa, tienen que entrenar datos en un modelo de machine learning extraído de la herramienta Scikit-learn, la cual está escrita en el lenguaje Python.

Como ya se mencionó, Apache Storm puede ejecutar los Jobs de procesamiento en un simple nodo o en nodos distribuidos (Procesamiento distribuido).

Análisis de datos

Se utiliza las librerías de la herramienta Scikit-learn, debido a que, contiene casi todos los modelos de predicción descritos de la sección de trabajos relacionados y estos modelos ya han sido probados en proceso de predicción sobre entornos e-learning.

Cubos y tableros

En esta sub capa se utilizará la herramienta LearningLocker, que ya tiene un cubo de información embebido. El mismo que trabajará directamente sobre el LRS de la solución informática.

Visualización

Esta sub capa me permite identificar gráficamente patrones que son muy difíciles de identificarlos de otra manera. Esto lo realizará con la herramienta Kibana, misma que permite la visualización de grandes volúmenes de información. Esta pertenece al paquete de software Elastic. Un conjunto de herramientas construidas para la gestión, análisis y visualización grandes volúmenes de datos.

Seguridad y gobierno

En esta capa se gestiona la administración de procesos. Se recomienda usar la herramienta Apache Zookeeper. Esta permite realizar un tracking de los eventos realizados por la herramienta Apache Storm, tanto en su funcionamiento como ETL y para el entrenamiento de los modelos de machine learning.

5. Validación

Las áreas de la ingeniería de software que son estudiadas en el presente trabajo son:

- Lenguajes específicos de dominio (DSL).
  Herramienta Easy Java/JavaScript Simulations: Me permite crear un laboratorio simulado en base a un modelo matemático.
- Arquitectura orientada a servicios (SOA).
  Análisis de los estándares de interoperabilidad en entornos e-learning xAPI y Caliper (Sección 3 y 4).
- Arquitectura y componentes de un sistema de Inteligencia de negocios (Sección 3).
- Arquitectura y componentes de un sistema de Big Data.
- Arquitectura y funcionamiento de algoritmos y técnicas de aprendizaje de máquina (Machine Learning).
- También para comprender mejor el funcionamiento de Easy Java/JavaScript Simulations y los algoritmos de machine learning, es necesario conocimientos básicos de estadística, algebra lineal, ecuaciones diferencias y calculo integral.

Como se puede apreciar, la solución informática abarca una extensa cantidad de temáticas de la ingeniería de software. Cada una estos temas pueden ser abordado por un trabajo de fin de master por separado. Sin embargo, este diseño se sustenta en:

- Un prototipo de un segmento de la solución informática.
- Las referencias bibliográficas de investigaciones realizadas en el contexto estudiado.
- 13 años de experiencia en la implementación y desarrollo de soluciones informáticas. De las cuales, al menos 7 años he ejercido el rol de Arquitecto de Software. Lo que, me ha permitido diseñar e implementar soluciones en temáticas afines al presente trabajo.

5.1. Prototipo de la solución informática

Este prototipo se enfoca en el proceso de recolección de datos mediante el estándar xAPI, a través de un entorno virtual desarrollado por la herramienta Easy Java/JavaScript Simulations. Para realizar dicho prototipo se realizó lo siguiente:

a. Definir el laboratorio virtual.
b. Construir un laboratorio virtual.
c. Integrar las librerías del cliente de xAPI en el laboratorio virtual.
d. Implementar un Learning Record Store (LSR) para el almacenamiento de los logs de información generados en el laboratorio virtual.
e. Probar el funcionamiento.
f. Conclusiones.

5.1.1. Definir el laboratorio virtual.

Los laboratorios virtuales del proyecto UNILabs están construidos con la herramienta Easy Java/JavaScript Simulations. Esta herramienta utiliza un modelo matemático y sus variables para generar un paquete de código JavaScript. El cual es embebido en un Learning Manager System (Moodle) y de esta manera es integrado al entorno de aprendizaje de la UNED.

La interfaz de un modelo simulado tiene una estructura variable, de acuerdo a la naturaleza del problema que se esté simulando. Pero podemos identificar 3 secciones:

- La simulación. Aquí se encuentran todos los componentes gráficos que se representan de acuerdo al modelo establecido.
- Un panel de configuraciones. Se establecen los parámetros que definen el comportamiento de la simulación.
- Un panel de controles. Son los controles de inicio, pausa y stop.

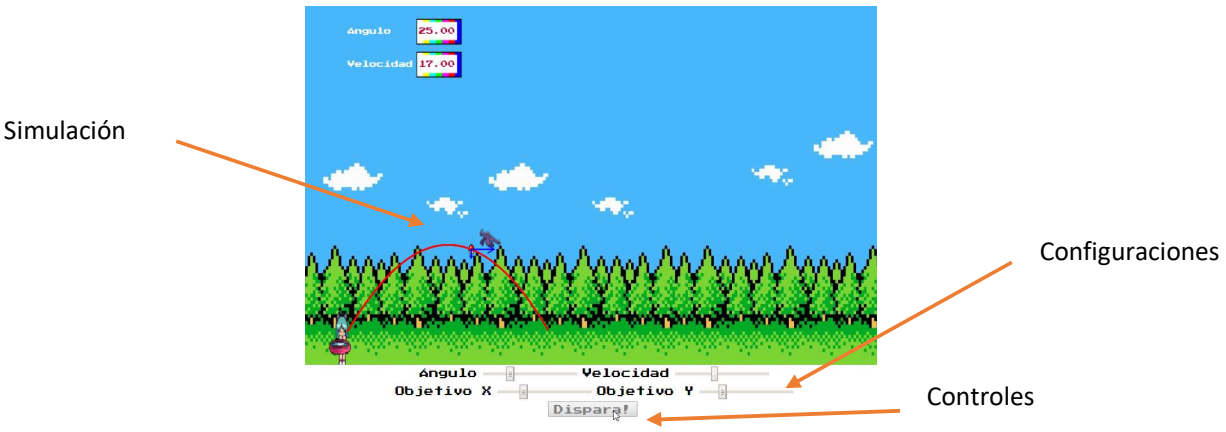

Figura 18. Estructura de la interfaz de la simulación.

El modelo que se utilizará para el prototipo es el **"Lanzamiento parabólico de proyectiles"**. El mismo que, le permitirá al estudiante validar en un entorno virtual el alcance y la trayectoria del lanzamiento de un proyectil, de acuerdo a un ángulo y a una velocidad inicial. Adicional, en el modelo se colocó una figura de una "ave" (Objetivo), con una ubicación geográfica variable. Esto para que, el estudiante intente acertar al objetivo y también para crear un medio dinámico de interacción.

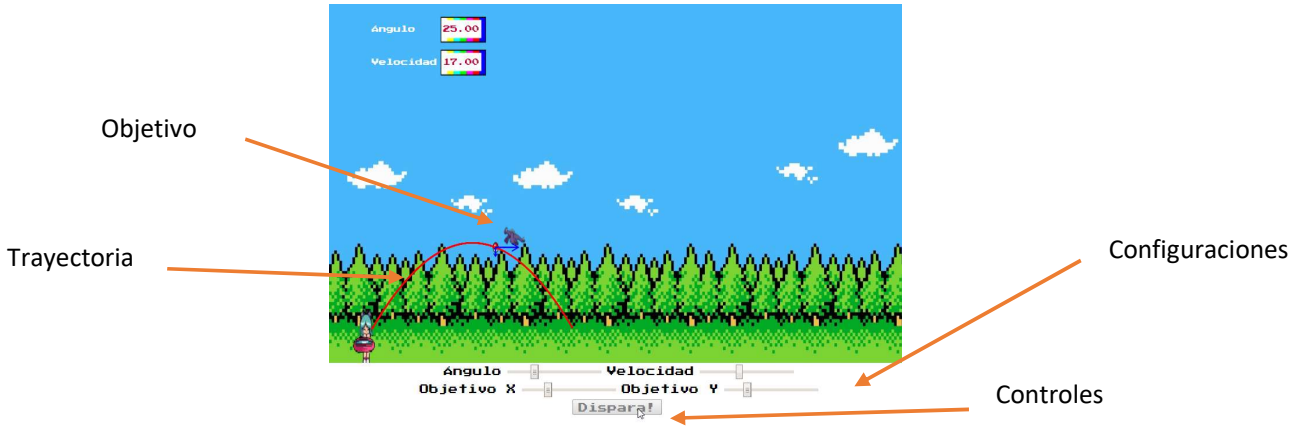

Figura 19. Interface del modelo del lanzamiento parabólico creado con el EJS.

En la figura anterior se puede visualizar el alcance y la trayectoria de un lanzamiento parabólico con una velocidad inicial de 17 km/h y un ángulo de 25 grados. También se puede visualizar en la figura:

- El panel de configuraciones. Se utiliza para configurar el ángulo y la velocidad inicial del lanzamiento y la ubicación "X" y "Y" del objetivo (Figura de un ave).
- Controles. Únicamente se tiene el control para iniciar la animación. En este panel no se estableció un botón para detener la animación. Debido a que, la animación se detiene automáticamente cuando el proyectil llega a tierra.
- Animación. Se puede visualizar la trayectoria del proyectil y en el caso de acertar al objetivo, se ejecuta una actividad indicando este acierto.

5.1.2. Construir un laboratorio virtual

Easy Java/JavaScript Simulations es una herramienta gratuita que permite fácilmente crear simulaciones en base a un modelo matemático. Está orientado a programadores con y sin experiencia. El despliegue puede generar componentes JavaScript y que pueden ser incrustados en una plataforma e-learning web como por ejemplo Moodle [27].

Tiene tres componentes principales:

- Descripción. Describe de forma narrativa el modelo a ser utilizado para la simulación.
- Modelo. Describe las variables del modelo, su inicialización y evolución (Puede ser una ecuación diferencial).
- Vista. Se especifica la interfaz gráfica del usuario (Botones, paneles, sliders, etc).

Para la construcción de un laboratorio virtual en la herramienta EJS se tiene que:

- Construir un modelo.

- Construir la interfaz gráfica.

5.1.2.1. Modelo

El modelo determina el comportamiento de la simulación. En EJS la opción “Modelo” tiene las siguientes pestañas:

- Variables.
- Initialization.
- Evolution.
- Fixed relations.
- Custom
- Elements

A continuación se describe únicamente las pestañas que fueron utilizadas para la creación del modelo:

5.1.2.2. Variables.

Aquí se establecen las variables matemáticas que se utilizarán en el modelo de la simulación, sin embargo, de requerirse otras variables para la animación, la interfaz o algún otro aspecto. Estas variables también tienen que definirse en esta misma pestaña. No está claro.

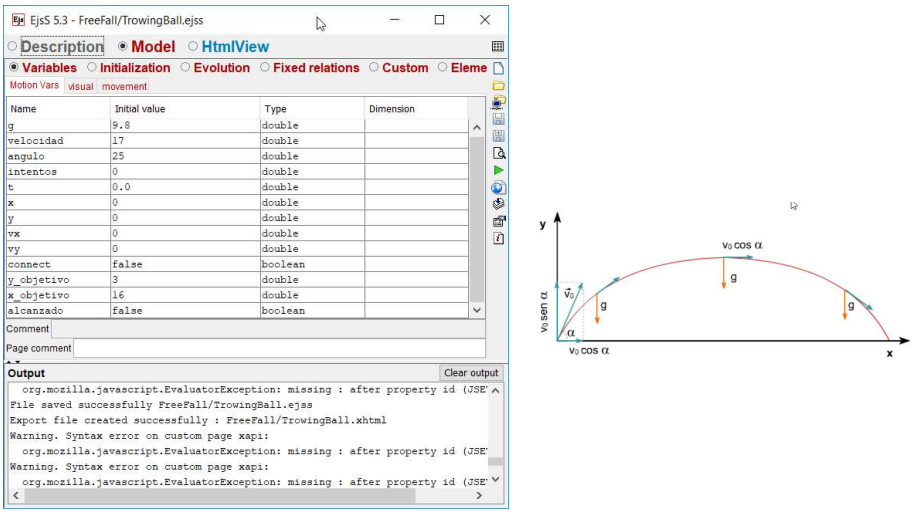

Figura 20. Variables del modelo para el lanzamiento parabólico de un proyectil

Las variables del modelo matemático son:

- g. Gravedad.
- velocidad. Velocidad inicial.
- ángulo. Angulo del lanzamiento.
- t. Tiempo.
- vx. Velocidad en el vector X.
- vy. Velocidad en el vector Y.
- x. Ubicación en el eje X.
- y. Ubicación en el eje Y.

También aquí se definen otras variables como, por ejemplo:

- alcanzado. Especifica si el objetivo fue alcanzado.
- y_objetivo. Ubicación en el eje Y del objetivo.
- x_objetivo Ubicación en el eje X del objetivo.

5.1.2.3. Evolución

La evolución define el comportamiento de las variables con respecto a una variable independiente. EJS le permite dos opciones:

- Una expresión matemática.
- Una ecuación diferencial.

En el prototipo utilizamos las ecuaciones diferenciales que se describen en la siguiente figura.

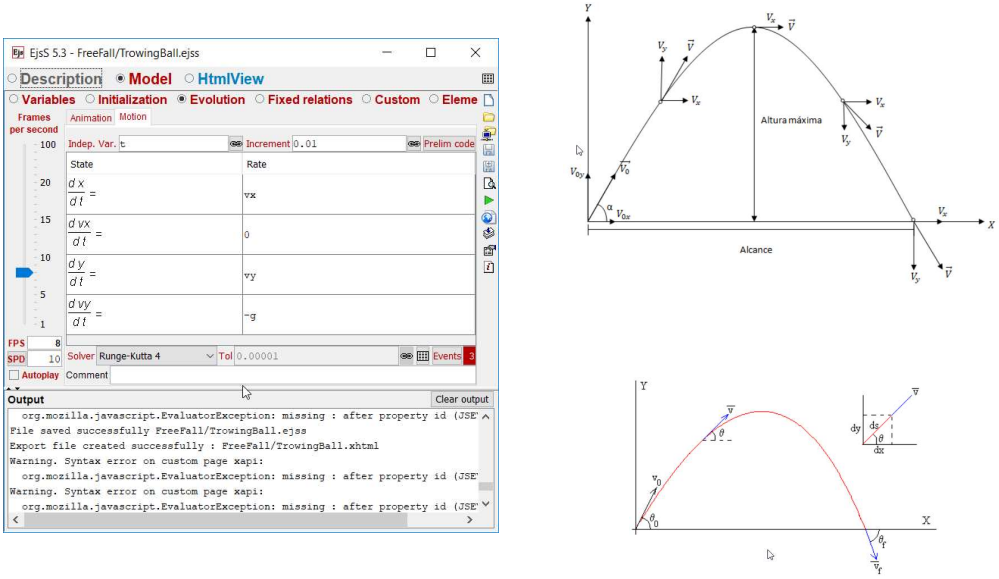

Figura 21. Variables del modelo para el lanzamiento parabólico de un proyectil.

En la figura 21 se puede visualizar las ecuaciones a la izquierda y una representación geométrica de las diferenciales para un mejor entendimiento del modelo de evolución.

5.1.3. Interfaz gráfica (HtmlView)

Una vez que se ha establecido las variables y las ecuaciones diferenciales, definimos la interfaz gráfica de usuario. En esta pestaña se diseña la interfaz gráfica en base a la incorporación de paneles, controles, etc. En la siguiente figura se visualiza los componentes definidos para el prototipo:

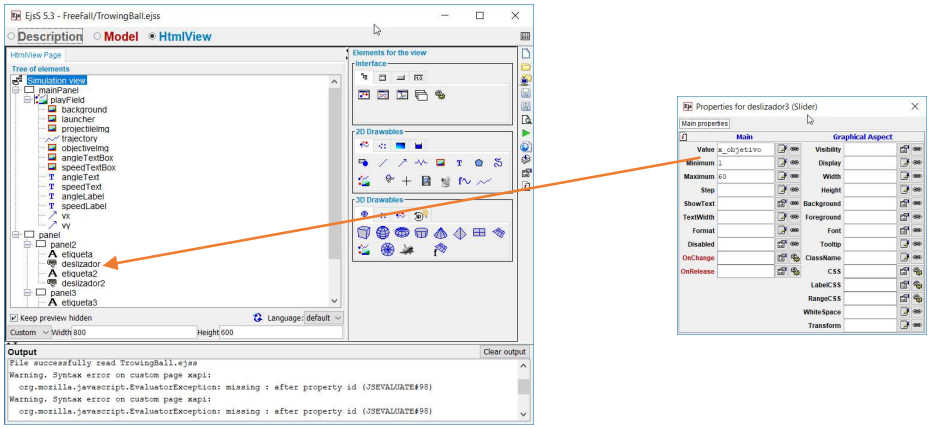

Figura 22. Interfaz gráfica de usuario del prototipo.

En la figura 22 se representa parte de los controles establecidos para la interfaz gráfica de usuario del prototipo. Además, también se especifica la forma en la que se relacionan las variables del modelo con los controles de la interfaz. Posterior a esto se ejecuta la simulación y se despliega el laboratorio virtual en el Browser.

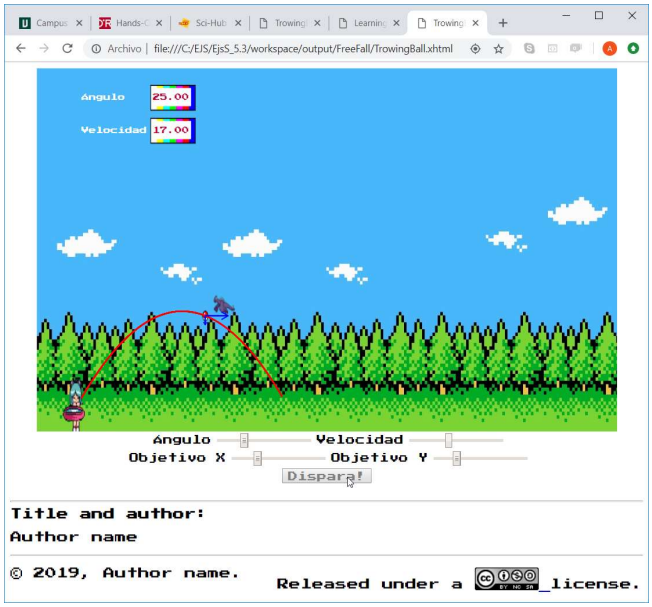

Figura 23. Interfaz generado del prototipo del lanzamiento parabólico de un proyectil.

5.1.4. Integrar xAPI

La especificación xAPI estandariza un formato para el envío de eventos y acciones generadas en entornos e-learning. Estos eventos son almacenados en un repositorio que se conoce como Learning Record System. Para el cual se han creado varias opciones Open Source que funcionan como un LRS y permiten el almacenamiento de mensajes en formato Json que siguen la especificación.

El LRS que integraremos en la arquitectura es la herramienta OpenSource LearningLocker. Misma que incluye un LRS, un módulo de reportes dinámicos y un módulo para la creación de tableros de control.

Para la implementación de xAPI tenemos que:

- Establecer un diccionario de acciones que podemos ejecutar siguiendo la especificación xAPI.
- Instalar el repositorio LRS LearningLocker
- Integrar xAPI en el EJS.
- Realizar las pruebas del prototipo.

5.1.4.1. Diccionario de xAPI

En base al análisis de eventos y acciones que pueden incidir en mayor magnitud a la predicción del progreso y compromiso (Análisis realizado en la sección 3. Solución), se ha definido el diccionario a ser usado para la implementación de xAPI. La tabla tiene las siguientes columnas:

- Nombre. Especifica el nombre de la actividad, verbo o extensión según la especificación xAPI.
- Tipo. Determina si se trata de una actividad, verbo o extensión.
- Uri. Especifica la referencia a los metadatos del componente según la especificación xAPI.
- Observación. incluye aspectos en los que se puede aplicar la especificación de xAPI en un caso real del laboratorio virtual.

Nombre	Tipo	Uri	Observación
Simulación	Activity Type	http://adlnet.gov/expapi/activities/simulation	
Cheklist	Activity Type	http://id.tincanapi.com/activitytype/checklist	
Checklist-item	Activity Type	http://id.tincanapi.com/activitytype/checklist-item	
Event	Activity Type	http://activitystrea.ms/schema/1.0/event	
Started	Verb	http://shindig2.epfl.ch/xapiextension.html#started	
Paused	Verb	http://id.tincanapi.com/verb/paused	
Resumed	Verb	http://adlnet.gov/expapi/verbs/resumed	
Stopped	Verb	http://shindig2.epfl.ch/xapiextension.html#stopped	
Saved	Verb	http://adlnet.gov/expapi/verbs/saved	
Configured	Verb	http://adlnet.gov/expapi/verbs/configured	Configura la simulación
Experience	Verb	http://activitystrea.ms/schema/1.0/experience	
Accessed	Verb	http://activitystrea.ms/schema/1.0/access	Accede a la simulación
Exited	Verb	http://adlnet.gov/expapi/verbs/exited	Abandona la simulación

Passed	Verb	http://activitystrea.ms/schema/1.0/agree	Lanzamiento de proyectil (Acertó con la trayectoria o distancia)
Failed	Verb	http://activitystrea.ms/schema/1.0/disagree	Lanzamiento de proyectil (Fallo el lanzamiento)
Find	Verb	http://activitystrea.ms/schema/1.0/find	Busca un control clave
Focused	Verb	http://id.tincanapi.com/verb/focused	Centra el foco en algún control clave
Longitude	Extension	http://id.tincanapi.com/extension/longitude	Registra su ubicación
Datetime	Extension	http://id.tincanapi.com/extension/datetime	Fecha y hora
Duration	Extension	http://id.tincanapi.com/extension/duration	Duración de sesión

Tabla 14. Diccionario de xAPI para laboratorios virtuales.

5.1.4.2. Implementar un LRS

El LRS es el repositorio de los eventos generados por las plataformas e-learning. De acuerdo al análisis realizado, el software que implementaremos como LRS es el LearningLocker. El proceso de instalación lo podemos encontrar en la página oficial de la solución informática. http://docs.learninglocker.net/guides-custom-installation/

La plataforma sobre la que se instaló LearningLocker es Linux. Bajo una distribución de Ubuntu, debido que no se trata de un entorno de producción. Los principales componentes que se requiere instalar son:

- MongoDB. Base de datos noSql.
  Componente para el almacenamiento de los archivos Json, de acuerdo a la especificación xAPI.
- Aapache. Servidor de aplicaciones.
  Learning Locker es un servicio y se instala sobre el servidor de aplicaciones Apache.
- redis. Bróker de mensajería.
  redis gestiona el envío y la recepción de mensajes de acuerdo a la especificación de xAPI, hacia el LRS.

En la siguiente figura se visualiza el servicio instalado en una máquina virtual con el sistema operativo Linux.

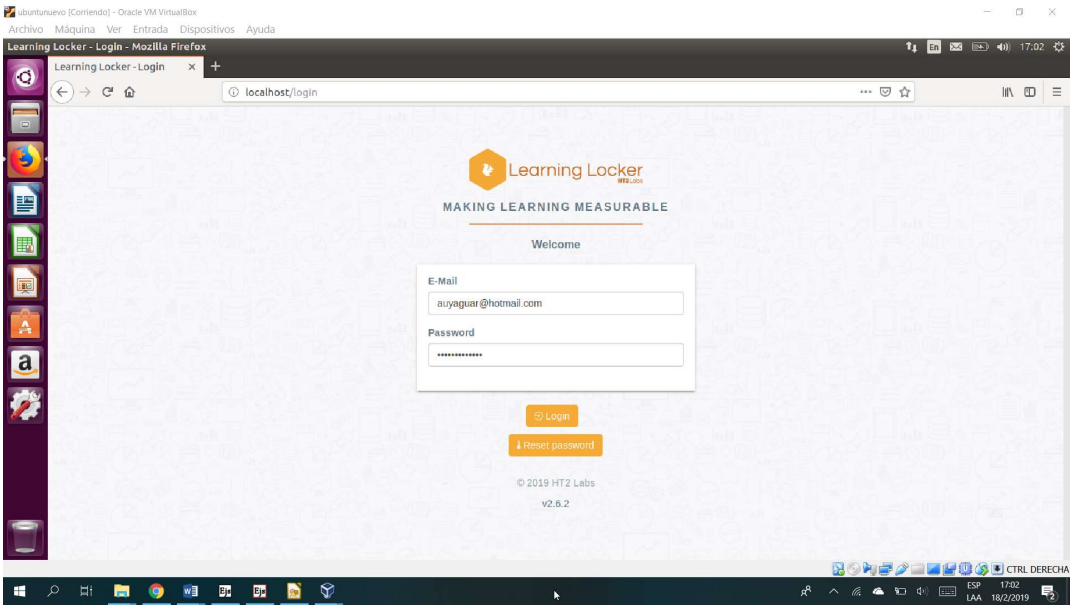

Figura 24. Servicio LearningLocker levantado

5.1.4.3. Integrar xAPI al laboratorio virtual.

xAPI tiene como uno de sus principales objetivos, el rastreo de cualquier evento de aprendizaje y la recolección de estas experiencias en casi cualquier lenguaje de programación que sea ampliamente difundido. Los clientes que xAPI tiene para la recolección de eventos son:

- JavaScript.
- Java.
- PHP.
- Python.
- .Net.
- Offline para JavaScript y Java.

EJS genera paquetes de código en JavaScript, por lo que el cliente que usaremos para la implementación es el cliente de JavaScript (TinCanJS). Y el proceso que seguiremos para la implementación es:

a. Agregar las librerías TinCanJS en el EJS.
b. Integrar en el EJS el código JavaScript necesario, para la conexión y el envío de mensajes al LRS. Adicional es necesario definir los eventos que serán notificados al LRS.

a. Agregar las librerías de TinCanJS en el EJS.

En opción HtmlView, opción “EJS options”, se agrega la librería TinCanJS. La que se puede descargar del siguiente link https://xapi.com/libraries/

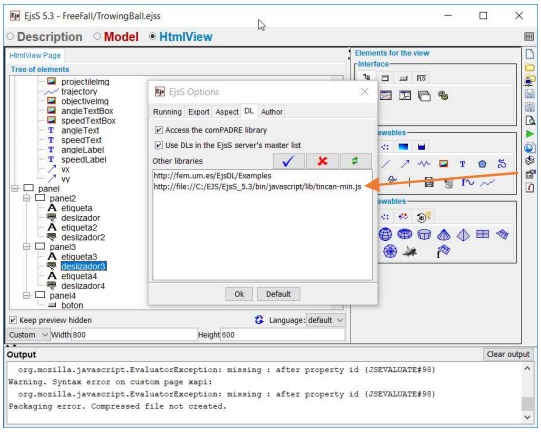

Figura 25. Agregar TinCanJS al EJS.

Este proceso permite que cuando se genere un modelo, de forma automática se agregue la librería a la página Web que contiene la simulación. Como se puede ver en la siguiente figura.

Figura 26. Página XHTML generada por el EJS e incluida la librería TinCanJS.

b. Integrar en el EJS el código JavaScript para el envío de eventos al LRS.

Los eventos son capturados en la Interfaz y tienen la siguiente estructura:

- Tipo de evento:
  - model: Indica que es un evento de inicio, pausa o stop de la simulación.
  - config: Cambio en la configuración del laboratorio.
  - event: Cualquier otro evento generado en el laboratorio. Por ejemplo. Se desliza en el panel.
- El tipo de control:
  - Button. Botón.
  - Slider. Slider.
  - Etc.
- Identificador del control. Ejemplo BtnStart.
- Tipo de acción que realiza. Estos son los verbos del diccionario de xAPI.

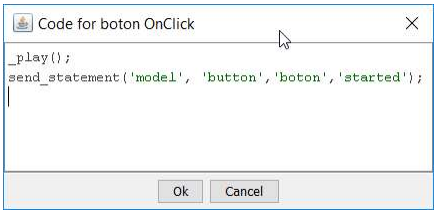

Figura 27. Notificación de evento desde un control Button

La figura 27 nos muestra un ejemplo de una notificación de envío de una sentencia al LRS. Esta sentencia será capturada y enviada al LRS. El código necesario para él envió de sentencias según la especificación de xAPI al LRS es incrustado en la opción modelo, pestaña custom. En este espacio del EJS se puede ingresar todo el código JavaScript personalizado, como se puede ver en la siguiente figura:

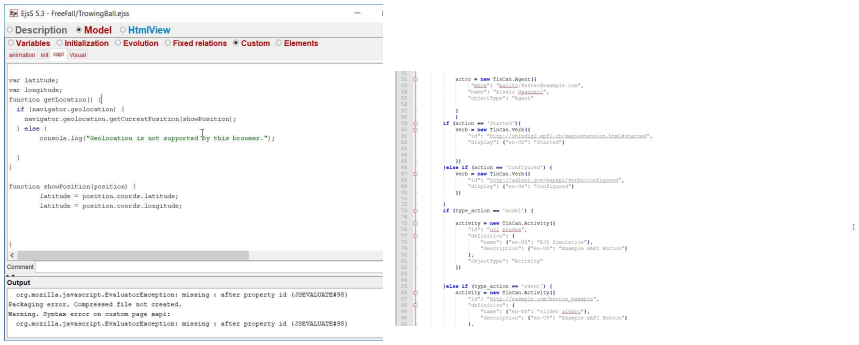

Figura 28. Código fuente para el registro de sentencias xAPI en el LRS.

5.1.4.4. Pruebas

Para el proceso de pruebas se procedió al envió de sentencias en formato Json desde el cliente de xAPI embebido en el EJS. Estas sentencias pueden ser visualizadas en el software LeraningLocker como se puede evidenciar en la siguiente figura.

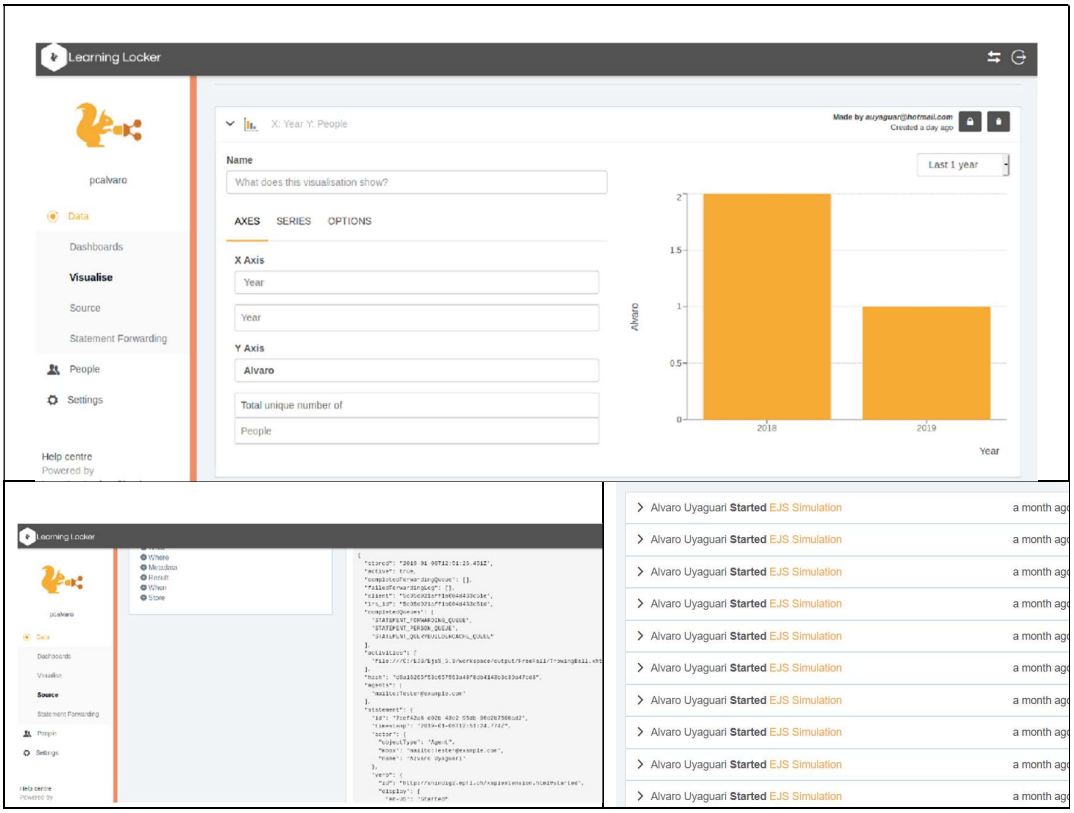

Figura 29. Reportes realizados en LearningLocker.

En la figura 26 se puede apreciar los diferentes reportes que se pueden construir con la herramienta LearningLocker. Estos reportes pueden ser resumidos, con gráficos y también se puede explorar una sentencia xAPI de forma individual.

5.1.5. Conclusiones

Como se explica al inicio de esta sección, la arquitectura involucra un amplio grupo de áreas de la ingeniería de software. Para la definición del esquema de la arquitectura y el establecimiento de los componentes, nos basamos en la literatura científica; pero también se realizaron implementaciones de tipo prototipo como la que detallamos en esta sección.

Esto nos ha permitido establecer componentes que pueden integrarse a la solución informática desde un enfoque teórico y práctico.

6. Conclusiones y trabajo futuro

6.1. Conclusiones

La aplicación de machine learning para crear servicios inteligentes y mejorar la experiencia de los usuarios se está aplicando en una gran diversidad de áreas. En el campo de la educación no es la excepción, debido a que existen múltiples investigaciones en este campo. Sin embargo, el aporte del presente trabajo, es el diseño del esquema de una solución informática para la búsqueda de patrones de comportamiento en los laboratorios virtuales del proyecto UNILabs. En estos laboratorios se generan datos finos y diversos. Lo que, vuelve más complejo el análisis y el almacenamiento de estos datos. A continuación, se describe los aspectos principales para el diseño de la arquitectura.

Una arquitectura es una representación abstracta de una solución informática. Como el plano de una casa para un arquitecto. El diseño analizado, determina la estructura general de una solución informática para predecir oportunamente el desempeño y el compromiso de un estudiante al realizar una práctica en un laboratorio virtual de la UNED. Para realizar esta predicción se utilizará algoritmos y técnicas de machine learning. Estos algoritmos pueden ser supervisados o no supervisados. Los algoritmos supervisados requieren variables independientes (predictoras) y dependientes (predecir) para la ejecución de un proceso de aprendizaje.

Las variables a predecir son el progreso y el compromiso del estudiante. Las variables predictoras son las determinadas en la sección 4 “Solución Informática” del presente trabajo. Las fuentes de información para extraer las variables son la base de datos de los logs de información del laboratorio virtual, el sistema académico y el sistema de gestión de aprendizaje (Moodle). La solución informática entrega al modelo de aprendizaje de máquina las variables sin ruidos, datos faltantes, ni datos incompletos y en una sola tabla. El diseño del proceso para entregar estas variables se detalla en los siguientes párrafos.

El diseño plantea el uso de la Arquitectura Orientada a Servicios (SOA) para que la solución pueda interactuar con otros sistemas o componentes mediante estándares internacionales. En el campo de la educación existen dos estándares para el registro de logs de información (xAPI y Caliper). xAPI norma un lenguaje y un vocabulario para cualquier tipo de log que genere el usuario en un ambiente e-learning Ejm. Inicio un video, termino un test, movió el slider, dio clic en un botón, etc. Mientras que Caliper limita los eventos a un contexto netamente educativo, debido a que solamente nos permite registrar eventos relacionadas al aprendizaje (Metric Profiles).

En conclusión, el estándar que mejor se adapta a el contexto de predicción que buscamos es xAPI. Debido a que las variables a recolectarse en el laboratorio virtual para

el proceso de predicción, son muy diversas y no se enmarcar solo en el proceso de aprendizaje Ejm. El usuario está configurando el laboratorio, está inactivo, movió el slider, etc.

El diseño arquitectónico parte de una arquitectura genérica para un sistema de inteligencia de negocios(BI). En un proceso de BI se contempla múltiples fuentes de información. Luego estas fuentes son procesadas por un integrador de datos que realiza un proceso de (ETL) extracción, transformación y carga de información en otra base de datos. Esta carga establece una nueva base de datos que sigue un modelo estrella o copo de nieve. El cual le permite al usuario realizar análisis en base a consultas dinámicas y multidimensionales. En el contexto de los laboratorios virtuales de la UNED. Se requiere este proceso de ETL y la reportaría dinámica. Por lo que, la arquitectura de BI fue una buena base para indicar el análisis.

Posterior a analizar la arquitectura para sistemas BI, se realizó el análisis de una arquitectura de Big Data. La cual incorpora herramientas que permiten realizar acciones similares a una arquitectura de BI. Pero, casi todas las herramientas de este entorno están diseñadas para manejar grandes volúmenes de información. Por lo que permiten un almacenamiento y procesamiento distribuido entre varios nodos. Adicional también incluyen una alta tolerancia a fallos.

Como parte del análisis se incluyó algunas herramientas de Big Data en el diseño de la arquitectura. Lo que le permite a la solución informática trabajar inicialmente en un solo servidor. Pero si luego se requiere más almacenamiento o procesamiento, el diseño arquitectónico le permite escalar horizontalmente en almacenamiento y procesamiento. Esto se logra incluyendo la herramienta TimescaleDB como base de datos de almacenamiento y el Apache Storm para realizar el proceso de extracción, transformación y carga.

En conclusión, en el presente trabajo se desarrolló un diseño de una arquitectura apegada a los requisitos y estándares internacionales. Es desacoplada y escalable debido a la naturaleza de la información que se va a almacenar y procesar (log files). Y el diseño esta validado en base a un prototipo y a investigaciones científicas relacionadas al contexto del presente trabajo.

6.2. Trabajo futuro

El presente trabajo es el punto de partida para llevar a cabo mi tesis doctoral. La cual consiste en el análisis del comportamiento de los estudiantes en entornos virtuales. Este análisis tiene como fuente de información los eventos de interacción entre el estudiante y la interfaz de la práctica desarrollada en el laboratorio virtual. Además, estos análisis serán realizados con modelos de aprendizaje de máquina y, también, sobre algoritmos y técnicas revisadas en el presente trabajo de fin de master.

Las aportaciones realizadas en el presente trabajo son fundamentales para futuros trabajos de estudiantes de postgrado que requiera implementar soluciones informáticas con fines analíticos y sobre flujos de datos generados en los sistemas informáticos, a

través de eventos o acciones ejecutados en la interfaz del usuario. Esto debido a que el estándar xAPI no se aplica únicamente a entornos educativos, sino también se está aplicando en otras áreas, como por ejemplo la salud, para el rastreo de acciones realizadas por los usuarios.

Referencias

[1] Aktan B., Bohus C., Crowl L., Shor M. (1996). Distance Learning Applied to Control Engineering Laboratories. IEEE Transactions on Education, Vol. 39, pages 320- 326. In press.

[2] Dormido, S. (2004). Control learning: Present and future. IFAC Annual Control Reviews, vol. 28, pages 115-136. In press.

[3] Guillet D., Nguyen A., Rekik Y. (2005). Collaborative Web-Based Experimentation in Flexible Enginnering Education. IEEE Transactions on Education, Vol. 48, Nº 4. In press.

[4] Sánchez J., Dormido S., Esquembre F. (2005). The learning of control concepts using interactive tools. Computer Applications in Engineering Education, Vol. 13, page 84. In press.

[5] Joseph Ingeno. (2018). Software Architect's Handbook. Packt Publishing, Birmingham, UK.

[6] TinCan/xAPI. (2019). Recipes: How it works. https://xapi.com/recipes-how-it-works/

[7] TinCan/xAPI. (2019). Statement design. https://xapi.com/statement-design/

[8] José Pedro Schardosim Simao, Lucas Mellos Carlos, Hamadou Saliah-Hassane, Juarez Bento da Silva, Joao Bosco da Mota Alves. (2018). Model for Recording Learning Experience Data from Remote Laboratories Using xAP. http://cleilaclo2018.mackenzie.br/docs/LACLO/FULL/184198.pdf

[9] Mert Onuralp Gökalp, Kerem Kayabay, Mohamed Zaki, Altan Koçyiğit, P. Erhan Eren, Andy Neel. (2017). Big-Data Analytics Architecture for Businesses: a comprehensive review on new open-source big-data tools. https://cambridgeservicealliance.eng.cam.ac.uk/resources/Downloads/Monthly%20Papers/2017OctPaperBigDataAnalytics.pdf

[10] Gunasekaran Manogaran, Daphne Lopez. (2017). A survey of Big Data architectures and machine learning algorithms in healthcare. Int. J. Biomedical Engineering and Technology, Vol. 25, pages 2-4. In press.

[11] Jacob Whitehill, Zewelanji Serpell, Yi-Ching Lin, Aysha Foster, and Javier R. Movellan. (2014). The Faces of Engagement: Automatic Recognition of Student Engagement from Facial Expressions. IEEE TRANSACTIONS ON AFFECTIVE COMPUTING, VOL. 5, NO. 1. In press.

[12] Chris Piech, Mehran Sahami, Daphne Koller, Stephen Cooper, Paulo Blikstein. (2010). Modeling How Students Learn to Program. In Proceedings of the 43rd ACM technical symposium on Computer Science Education, pages 153-160. SIGCSE '12.

[13] A. Serengul Guven Smith, Ann Blandford. (2002). MLTutor: An Application of Machine Learning Algorithms for an Adaptive Web-based Information System. International Journal of Artificial Intelligence in Education, vol. 13, pages 235-261, 2003.

[14] Mingjie Tan, Peiji Shao. (2015). Prediction of Student Dropout in e-Learning Program Through the Use of Machine Learning Method. iJET, Volume 10, Issue 1.In press.

[15] S. B. Kotsiantis, C. J. Pierrakeas, P. E. Pintelas. (2003). Preventing Student Dropout in Distance Learning Using Machine Learning Techniques. International Conference on Knowledge-Based and Intelligent Information and Engineering Systems, pages 267-274, Berlin. KES 2003.

[16] Hämäläinen W., Vinni M. (2006). Comparison of Machine Learning Methods for Intelligent Tutoring Systems. Intelligent Tutoring Systems. ITS 2006. Lecture Notes in Computer Science, vol 4053, Berlin. ITS 2006.

[17] Alireza Ahadi and Raymond Lister , Heikki Haapala, Arto Vihavaine. (2015). Exploring Machine Learning Methods to Automatically Identify Students in Need of Assistance. Proceedings of the eleventh annual International Conference on International Computing Education Research, pages 121-130, Omaha. ICER '15.

[18] Panagiotis Adamopoulos. (2013).What makes a great mooc? an interdisciplinary analysis of student retention in online courses. Thirty Fourth International Conference on Information Systems, Milan. ICIS.

[19] Mihaela Cocea, Stephan Weibelzahl. (2006). Can Log Files Analysis Estimate Learners' Level of Motivation?. https://researchportal.port.ac.uk/portal/files/223406/ABIS%202006.pdf

[20] S. B. Kotsiantis. (2012). Use of machine learning techniques for educational proposes: a decision support system for forecasting students' grades. Artificial Intelligence Review, Vol. 1, 1986 – Vol. 51. 2019.

[21] M.A. Chatti, A.L. Dyckhoff, U. Schroeder, H. Thüs. (2012). A Reference Model for Learning Analytics. International Journal of Technology Enhanced Learning (IJTEL), Vol. 4, No. 5/6. 2012.

[22] Nigel Bosch, Sidney D'Mello , Valerie Shute, Matthew, Jaclyn Ocumpaugh. (2015). Automatic Detection of Learning-Centered Affective States in the Wild. Proceedings of the 20th International Conference on Intelligent User Interfaces, Pages 379-388, Atlanta. IUI '15.

[23] UNED. (2019). Red Universitaria de laboratorios interactivos. https://unilabs.dia.uned.es/

[24] Henrik Brink Joseph W. Richards Mark Fetherolf. (2016). Real-World Machine Learning. Manning Publications, NY, United States.

[25] Bahaaldine Azarmi. (2016). Scalable Big Data Architecture: A Practitioner's Guide to Choosing Relevant Big Data Architecture. Apress, NY, United States.

[26] Jyotiswarup Raiturkar. (2018).Hands-On Software Architecture with Golang. Packt Publishing, Birmingham, UK.

[27] Francisco Esquembre. (2004). Easy Java/JavaScript Simulations: a software tool to create scientific simulations in Java. Computer Physics Communications, Pages 199-204. 2004.

[28] Ian H. Witten; Eibe Frank; Mark A. Hall. (2011). Data Mining: Practical Machine Learning Tools and Techniques, 3rd Edition. Morgan Kaufmann, Burlington, United States.

[29] Yinying Wang. (2016).Big Opportunities and Big Concerns of Big Data in Education. Y. TechTrends, Volume 60, pages 381–384. 2016.

[30] Timescale. (2019). Performant Time-Series Data Management and Analytics with PostgreSQL. https://www.timescale.com/

[31] MongoDB. (2019). Data Partitioning with Chunks. **https://www.mongodb.com/**

[32] LearningLocker. (2019). Install Learning Locker. **https://www.ht2labs.com/learning-locker-community/overview/**

[33] IMS. (2019). Caliper Analytics v1.1 Profiles. **https://www.imsglobal.org**

[34] Percona. (2017). High Performance JSON PostgreSQL vs. MongoDB. https://www.percona.com/live/e17/sites/default/files/slides/High%20Performance%20JSON%20-%20PostgreSQL%20vs.%20MongoDB%20-%20FileId%20-%20115573.pdf

[35] U.S. Government. (2019). ADL LRS. https://lrs.adlnet.gov/

Siglas, abreviaturas o acrónimos

Siglas, abreviaturas o acrónimos	Significado
Big Data	Grandes volúmenes de información.
xAPI	Experience API.
Caliper	Estándar de interoperabilidad para contenido de aprendizaje.
Json	Formato de texto para el intercambio de información.
XML	Lenguaje de marcado.
ETL	Extract, Transform and Load.
TimescaleDB	Open SourceTime-series database.
PostgreSQL	Gestor de base de datos Open Source.
BI	Business Intelligence.
Open Source	Código abierto.
Machine Learning	Aprendizaje automático de máquinas.
NoSQL	Sistemas de gestión de bases de datos que difieren del modelo clásico relacional.
MongoDB	Sistema de base de datos NoSQL orientado a documentos Open Source.
EJS	Easy Java/JavaScript Simulations.
DSL	Domain-specific language, Lenguaje de dominio específico.
Datamining	Minería de datos.
e-learning	Educación en línea.
UNILabs	University Network of Interactive Laboratories.
IEEE	Institute of Electrical and Electronics Engineers.
ACM	Association for Computing Machinery.
LMS	Learning Management System.
LRS	Learning Record Store.
AMS	Academic Management System.
Scopus	Base de datos bibliográfica de resúmenes y citas de artículos de revistas científicas.