

Universidad de la República

Montevideo, Uruguay

Extracción, transformación y almacenamiento de información de procesos educativos - Data Ánima

Autores:

Rodrigo Liñares
Antony Maciel
Joaquín Villar

Supervisor:

Dra. Libertad Tansini



*Proyecto de grado para la carrera Ingeniería en Computación
en el*

**Instituto de Computación
Facultad de Ingeniería**

22 de junio de 2021

Agradecimientos

Queremos agradecer a toda nuestra familia y amigos por el apoyo incondicional en estos años de formación. También a todas esas personas que a lo largo de la vida nos han motivado y acompañado en cada uno de nuestros proyectos. Además, queremos agradecer a la institución *Ánima* por haber confiado en nosotros para realizar este proyecto, en especial a los integrantes de la institución Soledad Mangino, Ignacio Gadola y Pablo Correa por haber estado siempre a disposición en el transcurso de este. En particular queremos agradecer a nuestra tutora Libertad Tansini por habernos guiado y acompañado en la última etapa de nuestra carrera.

Resumen

Ánima es una institución de educación formal de bachillerato tecnológico en Uruguay que se enfoca en alumnos con una situación socioeconómica vulnerable. En la institución se usan varios sistemas informáticos en donde los datos generados por estos son de gran valor dado que permite la toma de decisiones dentro de la misma. Esto es posible dado a que en Ánima se desarrollan reportes denominados cuadros de mando con dicha información para tener una visualización global del estado de la institución.

Este informe presenta el trabajo realizado en el marco del proyecto de grado de la Facultad de Ingeniería para abordar las problemáticas de extracción, transformación y almacenamiento de información de procesos educativos de Ánima, en principio para ayudar a la institución a demarcar objetivos, prioridades y encontrar soluciones para algunas de las necesidades más relevantes.

El proyecto tiene tres grandes objetivos. Como primer objetivo se planteó realizar un relevamiento de requerimientos para detectar las problemáticas en la institución. Algunas de las problemáticas detectadas en la institución fueron, por un lado, que no se obtenían datos de su plataforma Moodle dado que no se contaba con una forma de extracción amigable para el usuario. Y por otro lado se realizaba un proceso manual en la integración de sus datos para la generación de los reportes. A partir de estas dos problemáticas se obtuvieron el segundo y tercer objetivo, que fueron realizar un proceso de extracción de la plataforma Moodle y plantear una solución a la integración de todos los datos provenientes de los sistemas utilizados por la institución respectivamente.

Como resultado del segundo objetivo se desarrolló un prototipo de plataforma denominada Data Ánima que se encarga de extraer la información de Moodle y almacenarla en un archivo excel. Dicho archivo obtenido es utilizado para su posterior análisis por medio de un visualizador de datos. Con respecto al tercer objetivo se obtuvo como resultado un diseño de integración de los datos a extraerse de los sistemas informáticos que utiliza la institución. Esta integración involucra el diseño de un Data Warehouse que comprende la estructura de los datos a tratar.

Al concluir el proyecto se cumplieron con los tres objetivos planteados, los cuales fueron validados y verificados. A su vez se proponen trabajos a futuro para brindar una solución completa a las problemáticas de Ánima.

Palabras Claves: Moodle, ETL, Data Warehouse, API, Google Data Studio, Extracción de datos, Análisis de datos educativos

Glosario

ADM	Administración.
Amazon	Compañía de servicios de computación en la nube.
Ánima	Bachillerato Tecnológico en Administración y Tecnología de la Información y Comunicación.
API	Sigla de Application Programming Interface. Del español interfaz de programación de aplicaciones, en un sistema es un conjunto de servicios que se ofrecen para ser utilizado por otro software, lo que sirve como capa de abstracción.
Back end	Capa encargada de los procesos de negocios y accesos a datos, ejecutado del lado del servidor.
CMDM	Modelo Conceptual para la Especificación de Bases Multidimensionales.
COVID-19	Enfermedad por coronavirus de 2019, más conocida como COVID-19.
CSV	Tipo de formato de archivo que representa valores de una tabla separados por coma.
Cuadros de mando	Reportes creados en Google Data Studio y nombrados de esta manera por Ánima.
Data Ánima	Prototipo de extracción Moodle desarrollado por el grupo de proyecto para Ánima.
Data Marts	Es una versión específica de Data Warehouse centrado en un tema dentro de una organización.
Data Mining	Del español Minería de datos.
Data Warehouse	Del español Almacén de Datos. Colección de datos orientada a un determinado ámbito, integrado, no volátil y variable en el tiempo.
EC2	Tipo de instancia de servidor brindada por Amazon.
ETL	Sigla de Extract, Transform and Load. Del español Extracción, Transformación y Carga. Es un proceso de manipulación de datos para ser movidos desde una fuente de datos hacia otra.
Excel	Herramienta que ofrece hojas de cálculo.
Foreign Key	Del español clave foránea, es una referencia entre tablas en una base de datos.
Front end	Sitio web con la interfaz de Usuario.
Git	Software de control de versiones.
GitHub	Plataforma de desarrollo colaborativo que utiliza el sistema de control de

	versiones Git.
Google Data Studio	Visualizador de datos gratuito.
Google Form	Aplicación para realizar formularios (encuestas).
HOLAP	Siglas de Hybrid OLAP. Es una combinación de herramientas MOLAP y ROLAP.
HTTP	Protocolo de comunicación para transferencia de información en la web.
INOA	Sistema con información no académica de alumnos.
INSE	Sigla de Índice de Nivel Socio Económico.
IP	Es la dirección de un servidor en la red y se representa a través de un conjunto de números que lo identifica de manera lógica y jerárquica.
Javascript	Lenguaje de programación.
JOIN	Sentencia SQL que permite combinar registros de una o más tablas.
JSON	Formato de texto diseñado para el intercambio de datos.
Learning Analytics	Del español Analítica de Aprendizaje, es la medición, recopilación, análisis e informe de datos sobre los alumnos y sus contextos, con el fin de comprender y optimizar el aprendizaje y los entornos en los que se produce.
Logs	Registro de los acontecimientos, eventos o acciones que afectan a un proceso particular.
Microsoft	Compañía tecnológica multinacional.
MIDES	Sigla de Ministerio de Desarrollo Social. Es un organismo público del Uruguay.
MOLAP	Sigla de Multidimensional OLAP. Alternativa a la tecnología ROLAP, basada su almacenamiento en una matriz multidimensional.
Moodle	Plataforma web orientada a la educación.
NodeJs	Entorno de ejecución de Javascript para la capa del servidor.
OLAP	Sigla de Online Analytical Processing. Del español Procesamiento Analítico En Línea.
OLTP	Sigla de Online Transaction Processing. Del español Procesamiento de Transacciones En Línea.
Pandemia	Epidemia de una enfermedad infecciosa que se ha propagado en un área geográficamente extensa, afectando a un número considerable de personas.
PostgreSQL	Motor de base de datos relacional.
Postman	Plataforma colaborativa para el desarrollo de API's.
React	Librería Javascript para el desarrollo

	Front end.
REST	Sigla de Representational State Transfer, del español transferencia de estado representacional es un estilo de arquitectura de software para sistemas web.
ROLAP	Sigla de Relational OLAP. Herramienta OLAP implementada sobre sistemas relacionales.
Script	Secuencia de comandos informáticos.
SIGED	Sistema de Gestión Educativa.
Single Page Application	Del español aplicación de página única, se refiere a una aplicación que se basa en una página web que contiene una única página.
SQL	Lenguaje de consulta estructurada para una base de datos.
TIC	Sigla de Tecnología de la Información y Comunicación.
Token	Cadena de caracteres utilizada para facilitar la autenticación de usuarios en alguna plataforma.
Typescript	Lenguaje de Programación basado en Javascript.
Ubuntu	Sistema operativo de software libre y código abierto.
URL	Sigla de Uniform Resource Locator. Del español localizador de recursos uniforme es el identificador de una página web.
Visualizadores de datos	Plataformas que expresan los datos de forma visual.

Índice general

Agradecimientos	III
Resumen	v
Glosario	vii
1. Introducción	1
2. Contexto y marco teórico	3
2.1. Contexto	3
2.1.1. Ánima	3
Moodle en Ánima	4
SIGED	4
INOA	5
Evaluador	5
Google Form	5
Archivos excel internos	5
2.2. Trabajos relacionados	5
2.2.1. Learning Analytics	5
2.2.2. Conclusión de los trabajos relacionados	6
2.3. Data Warehouse	7
2.3.1. Modelo Multidimensional	7
Dimensiones	7
Medidas	8
Hipercubos	8
CMDM	8
2.3.2. OLAP	10
2.3.3. Diseño Lógico de Data Warehouse	12
ROLAP	12
Esquema Estrella	12
2.3.4. Arquitectura de Data Warehouse	13
2.4. Moodle	14
2.5. Visualizadores de datos	14
2.5.1. Google Data Studio	15
2.6. ETL	15
2.6.1. Extracción	16
2.6.2. Transformación	16
2.6.3. Carga	16

3. Metodología de trabajo	17
3.1. Planificación	17
3.2. Relevamiento de requerimientos	19
3.3. Herramientas utilizadas	20
3.3.1. Trello	20
3.3.2. Git	20
3.3.3. Google Drive	21
3.3.4. Google Meets	21
3.4. Reuniones de avance	21
3.4.1. Reuniones con Anima	22
3.5. Proceso de desarrollo	22
3.5.1. Obtención de información de Moodle	22
3.5.2. Integración de los datos	24
3.5.3. Verificación	24
3.5.4. Versionado	24
3.5.5. Gestión	25
4. Análisis y Diseño	27
4.1. Relevamiento de requerimientos	27
4.1.1. Problemática	27
4.1.2. Datos existentes	29
4.1.3. Relevamiento	31
Obtención de datos de Moodle	31
Integración	33
Resumen del relevamiento de requerimientos	33
4.2. Arquitectura	34
4.2.1. Módulo de extracción de Moodle	34
4.2.2. Data Warehouse	34
4.2.3. Solución general	35
4.2.4. Alternativas analizadas	36
5. Implementación del prototipo de extracción de Moodle	39
5.1. Arquitectura	39
5.2. Tecnologías	42
5.3. Funcionalidades	43
5.4. Reportes obtenidos	45
5.5. Verificación de los datos obtenidos	48
5.5.1. Cursos	48
5.5.2. Entregas	49
Calificaciones	49
5.5.3. Elementos	50
5.5.4. Usuarios	50
5.5.5. Roles	51
5.5.6. Foros	51
6. Diseño del Data Warehouse	53
6.1. Dimensiones y Medidas	53
6.2. Relaciones Dimensionales	57
6.3. Funciones de agregación	59

6.4. Diseño Lógico	60
6.5. Implementación de un prototipo para la verificación del Data Warehouse	63
7. Extracción, transformación y carga de datos	67
7.1. Moodle	67
7.1.1. API de Moodle	68
7.1.2. Formato de excel	71
7.2. Sistemas existentes	74
7.3. Carga de Data Warehouse	75
8. Verificación y validación de resultados	77
8.1. Verificación de la obtención de datos de Moodle	77
8.2. Verificación del diseño del Data Warehouse	77
8.3. Validación de los usuarios	77
9. Conclusiones y trabajo a futuro	79
9.1. Conclusiones generales	79
9.2. Trabajo a futuro	80
Bibliografía	81
A. Manual de usuario	85
A.1. Data Ánima	85
A.2. Consideraciones para tener en cuenta	88
B. Manual de instalación de Data Ánima	91
B.1. Configuración del servidor	91
B.1.1. Conexión al servidor e instalación de Data Ánima	91
B.1.2. Back end	93
B.1.3. Front end	93
B.2. Configuración de servicios en Moodle	93
B.2.1. Habilitar servicios web	94
B.2.2. Habilitar protocolos	94
B.2.3. Crear un servicio	95
B.2.4. Agregar funciones a un servicio	96
B.3. Configuración de permisos de usuarios en Moodle	96
C. Manuales del prototipo de Data Warehouse	99
C.1. Instalación y configuración de la base de datos	99
C.2. Ejecución del Script en la base de datos	100
C.3. Instalación del prototipo de Data Warehouse	100
C.3.1. Instalación del prototipo para pruebas y desarrollo	101
C.3.2. Instalación del prototipo en Heroku	102
C.4. Manual de usuario	102
D. Servicios web de Moodle	105
D.1. Obtención del Token	105
D.1.1. Envío	105
D.1.2. Respuesta	105
D.1.3. Información utilizada	105

D.2. Obtención de información de un usuario	105
D.2.1. Consulta	105
Envío	105
Respuesta	106
D.3. Obtención de información de los cursos	107
Consulta	107
D.3.1. Envío	107
D.3.2. Respuesta	107
D.3.3. Información utilizada	108
D.4. Obtención de los contenidos de un curso	109
D.4.1. Consulta	109
Envío	109
Respuesta	109
D.4.2. Información utilizada	110
Secciones	110
Elementos	110
D.5. Obtención de información de las entregas	110
D.5.1. Consulta	110
Envío	110
Respuesta	111
D.5.2. Información utilizada	112
D.6. Obtención de la calificación de las entregas	112
D.6.1. Consulta	112
Envío	112
Respuesta	112
D.6.2. Información utilizada	113
D.7. Obtención de información de los usuarios en un curso	113
D.7.1. Consulta	113
Envío	113
Respuesta	113
D.7.2. Información utilizada	114
Usuario	114
Roles	114
D.8. Obtención de información de los foros de un curso	115
D.8.1. Consulta	115
Envío	115
Respuesta	115
D.8.2. Información utilizada	115
D.9. Obtención de información de las discusiones de un foro	116
D.9.1. Consulta	116
Envío	116
Respuesta	116
D.9.2. Información utilizada	117
D.10. Obtención de información de los posts de una discusión	117
D.10.1. Consulta	117
Envío	117
Respuesta	117
D.10.2. Información utilizada	118

E. Diagramas del Diseño del Data Warehouse	119
E.1. Dimensiones y Medidas	119
E.2. Relaciones Dimensionales	122
E.3. ETL	127
E.4. Consultas SQL Prototipo	132
E.4.1. Consulta de cantidad de entregas de tareas en Moodle	133
E.4.2. Consulta de cantidad de mensajes en foros de Moodle	133
E.4.3. Consulta de cantidad de elementos en Moodle	134

Capítulo 1

Introducción

Ánima es una institución de educación formal de Bachillerato Tecnológico en la cual se utilizan varios sistemas informáticos, entre los cuales se encuentra la plataforma Moodle, para llevar registro de sus actividades diarias. Dado que en la institución se enfrentaron a varias problemáticas con respecto a estos sistemas, es que surge un proyecto de grado en la Facultad de Ingeniería que aborde las mismas. Para dicho proyecto se tenía como primer objetivo realizar un relevamiento de requerimientos en Ánima con respecto a los desafíos que tenían con los datos existentes en los distintos sistemas que utilizaban. Como parte del relevamiento se realizaron una serie de reuniones con los integrantes de Ánima donde se detectó que la información de dichos sistemas (a excepción de Moodle) es obtenida de forma manual en archivos excel y luego cargada en la plataforma Google Data Studio [14] donde se realizan reportes denominados cuadros de mando, desde los cuales se analiza la información en busca de mejorar ciertas áreas de la institución. A partir del comienzo de la pandemia la institución incrementó el uso de la plataforma Moodle [40], lo que la llevó a querer incluir cierta información en los reportes de cuadros de mando ya que hasta el momento no tenían ninguna forma de exportar la misma. Como consecuencia de este relevamiento de requerimientos se dedujeron los siguientes objetivos en el proyecto, por un lado, implementar un proceso para extraer y transformar la información proveniente de la plataforma Moodle como segundo objetivo. Y por otro lado se dedujo la falta de un diseño para centralizar la información de todos los sistemas informáticos, transformarla y unificarla para luego ser analizada, por lo que se fijó como tercer objetivo la integración de esta.

Como resultado al segundo objetivo de dicho proyecto se presenta el prototipo de extracción de Moodle implementado. Este es denominado Data Ánima y se compone de un sitio web y un servidor, los cuales se comunican entre sí y brindan una experiencia amigable a los usuarios. Solamente aquellos usuarios de Moodle que tengan los permisos designados por Ánima pueden ingresar al sitio web en donde podrán ejecutar el proceso de obtención de la información. Dicho procesamiento es ejecutado por el servidor y cuando este finaliza los usuarios pueden descargar la información procesada en forma de archivo excel desde el sitio web ofrecido. La implementación de Data Ánima está basada en un proceso ETL, ya que esta se basa en la extracción de información de Moodle, luego esta información es transformada para finalmente ser almacenada en un archivo excel.

Por otro lado, como resultado del tercer objetivo para la integración de los datos de la institución se presentan algunas alternativas analizadas y se detalla la seleccionada como solución. La solución elegida se trata de un Data Warehouse

que permite alojar los datos de todos los sistemas existentes que utiliza la institución y explotar los mismos para su posterior análisis a través de herramientas que permiten obtener reportes. Se brinda un detalle completo del diseño del Data Warehouse, comenzando desde el diseño conceptual que está basado en el modelo CMDM [3] para luego proceder al diseño lógico basado en bases de datos relacionales. A fin de diseñar el Data Warehouse fue necesario presentar una breve explicación de las fuentes de datos que se deseaba integrar y también describir los indicadores de interés para la toma de decisiones de la institución. Para luego sí, como parte del diseño conceptual mencionado anteriormente, presentar las estructuras de los modelos multidimensionales que permitirán alojar los datos (denominadas Medidas, Dimensiones y Relaciones Dimensionales) de manera que se obtengan los análisis buscados. A partir de estas estructuras se presenta el diseño lógico basado en tablas relacionales, de las cuales se puede inferir el Script de una base de datos relacional para dar soporte al Data Warehouse.

Con el fin de verificar las soluciones propuestas y desarrolladas se realizaron pruebas que determinaron la correctitud tanto del diseño del Data Warehouse a través de un prototipo implementado, como de la extracción de los datos obtenidos por Data *Ánima*. A su vez como resultado de dicho proyecto se validó mediante una encuesta la satisfacción de los referentes de *Ánima* con respecto al trabajo realizado.

Como se puede observar al final de dicho proyecto se obtienen resultados de gran valor para *Ánima* ya que de esta forma logran tener acceso a información que con anterioridad a este proyecto era de difícil acceso para la institución. Además, analizando a futuro, de lograr concretarse en su totalidad la solución general planteada en dicho proyecto se podría eliminar mucho trabajo manual y repetitivo que se realiza hoy en día en la institución para hacer un análisis de los datos provenientes de todos los sistemas informáticos que utiliza *Ánima*. Como se puede observar en el prototipo de integración implementado durante este proyecto, la solución le brindará grandes ventajas a la hora de analizar los datos de la institución ya que se encuentran todos centralizados en el Data Warehouse propuesto.

La estructura de este documento se compone en capítulos, donde en los primeros (2 y 3) se describe el contexto y marco teórico en base al que se desarrolló el proyecto, como también la metodología empleada para la realización de este. Luego en los siguientes capítulos (4, 5, 6 y 7) en base al análisis de los requerimientos realizado se plantea la solución y la arquitectura a desarrollarse, así como una descripción detallada de Data *Ánima*, el proceso de extracción de Moodle planteado y el diseño del Data Warehouse propuesto. Por último, en el Capítulo 8 se describen las verificaciones y validaciones realizadas sobre el proyecto desarrollado, y en el Capítulo 9 se obtienen las conclusiones y trabajo a futuro planteados para este proyecto. Por otra parte, se incluye en los anexos manuales de usuario, de instalación y configuración de los prototipos desarrollados, así como también una breve documentación de los servicios web de Moodle utilizados en el proyecto y en la última sección se incluyen diagramas sobre el diseño obtenido para el Data Warehouse.

Capítulo 2

Contexto y marco teórico

En este capítulo se describe el contexto del proyecto que se desarrolló. Este debía adaptarse a la realidad de una institución educativa que ya contaba con algunas soluciones. Además, se plantea el marco teórico sobre el cual se trabajó, desarrollando los conceptos de Data Warehouse, Moodle y Visualizadores de datos.

2.1. Contexto

En un principio se presentó en la Facultad de Ingeniería la propuesta de proyecto de grado para la institución *Ánima* para fijar objetivos, prioridades y diseñar soluciones para algunas de sus necesidades más relevantes. Este proyecto gira en torno a *Ánima* donde se pueden encontrar varios sistemas informáticos que intervienen en el día a día de la institución. Uno de los sistemas utilizados para el manejo de cursos, docentes y alumnos es Moodle. Con la información proveniente de algunos de dichos sistemas se generan reportes de cuadros de mando para la toma de decisiones y visualización del estado actual de la institución. A continuación, se detalla en mayor profundidad *Ánima* como institución y los diferentes sistemas utilizados.

2.1.1. *Ánima*

Es una institución de educación formal de Bachillerato Tecnológico en Administración y Tecnología de la Información y Comunicación. Se caracteriza por ser de gestión privada y acceso gratuito y se desarrolla en diferentes espacios de aprendizaje: el aula y las empresas. Cuenta con una propuesta de formación integral dirigida a los jóvenes de Uruguay situados en los departamentos de Montevideo y Canelones que viven en situación de vulnerabilidad social. Los alumnos deben haber aprobado ciclo básico y querer hacer bachillerato tecnológico. Su nueva propuesta educativa promueve la innovación, trabajando en proyectos y utilizando tecnologías en laboratorios digitales.

La misión de la institución, como su sitio web lo detalla, es: *“Potenciar el desarrollo humano y profesional de los jóvenes en situación de vulnerabilidad social, brindándoles las herramientas necesarias para su realización personal. Mediante el fortalecimiento de vínculos entre el mundo del trabajo, la institución educativa, los alumnos y sus familias, construimos aprendizajes significativos que permitan la inserción de los jóvenes en el campo laboral y su plena participación en la sociedad.”*. A su vez, la visión de *Ánima* tal cual como se la plantean es: *“Una sociedad comprometida con la educación del país, inclusiva y de calidad, que*

permita a los jóvenes desarrollar las habilidades, capacidades y conocimientos para desempeñarse como ciudadanos libres, responsables y colaboradores, para la construcción de un país con más y mejores oportunidades.”. [1]

La organización de los grupos de alumnos en *Ánima* se divide en dos por cada una de las tres generaciones de alumnos en curso. Uno de los grupos es de orientación TIC y otro de orientación ADM. Cada uno tiene un promedio de 14 materias. Los alumnos reciben las evaluaciones con las calificaciones e inasistencias obtenidas en las materias en las cinco entregas de boletines anuales.

Como toda institución educativa del Uruguay, *Ánima* tiene la obligación de utilizar ciertas herramientas digitales para el registro de la información generada. Esto lleva a que se divida en distintos sistemas toda la información de la institución. Por su parte, *Ánima* se encarga de unificar toda esta información en una única fuente de datos. Con ayuda de la herramienta Google Data Studio generan reportes denominados como cuadros de mando. Con los mismos logran realizar análisis sobre la institución, sus docentes y alumnos, para tener un mejor conocimiento de la realidad en la institución y tomar decisiones al respecto.

Durante el transcurso del proyecto la institución estaba viviendo una situación muy especial debido a la pandemia generada por el COVID-19. En consecuencia, la utilización de herramientas digitales y la necesidad de evaluar y analizar el uso que se le da a las mismas aumentó de forma exponencial. A continuación, se detallan los sistemas de información relevantes para este proyecto utilizados por la institución.

Moodle en *Ánima*

Ánima cuenta con un entorno virtual de aprendizaje llamado Moodle que se describe más adelante en la Sección 2.4 cuya versión es la 2.9.

Esta herramienta se comenzó a utilizar con mayor énfasis en el comienzo del año 2020 debido a la pandemia. Anteriormente, el uso que se le daba a la plataforma era mínimo. Se utilizaba principalmente para que los docentes subieran material relacionado al curso para sus alumnos, así también como para plantear propuestas de trabajos con entregas. Comenzada la pandemia, debido a que toda la actividad académica se llevó a cabo desde los hogares, se comenzó a utilizar Moodle, no solo para las finalidades mencionadas anteriormente, sino también como forma de comunicación a través de los foros. En consecuencia, aumentó el uso de la plataforma con respecto a los índices de uso previo.

Debido al gran aumento de uso de la plataforma, aumentó el trabajo administrativo manual. Los encargados de gestionar y planificar los cursos debían ingresar periódicamente en Moodle para verificar, por un lado, que los docentes subieran el material correspondiente para el desarrollo de los cursos, y por otro, que los alumnos participen de manera activa.

SIGED

Sistema de Gestión Educativa que automatiza totalmente el manejo de la información de los institutos educativos. Este sistema está certificado por el Consejo de Educación Secundaria. [39]

A través de un usuario que posee *Ánima* dentro del sistema SIGED, obtiene los datos almacenados en el mismo en archivos excel. En dicho sistema se encuentra la mayor parte de la información de interés de la institución actualmente.

INOA

Una aplicación para centros educativos que permite dar seguimiento a la información no académica de los alumnos [20]. *Ánima* solicita al proveedor de INOA un archivo excel que contiene toda la información de la base de datos del sistema, ya que no cuenta con reportes predeterminados.

Evaluador

Sitio de *Ánima* donde se aloja la evaluación de competencias transversales. Los datos en este sitio se exportan en archivos excel.

Google Form

Ánima utiliza formularios de Google Form para obtener datos que se registran de manera automática en archivos excel.

Archivos excel internos

Existen archivos excel con diversos datos de *Ánima* que son construidos por la institución y de uso interno.

2.2. Trabajos relacionados

Se investigaron ciertos aspectos generales que se sabía podrían llegar a ser tratados durante el proyecto. Sabiendo a priori que se debía interactuar con Moodle e implementar un módulo de Learning Analytics, se decidió indagar en estos temas. En la investigación sobre Learning Analytics se recabaron algunos proyectos existentes [26] [38] [29] para familiarizarse con el tema y analizar su utilidad ante la realidad del proyecto.

Para recabar información sobre cómo interactuar con Moodle para obtener datos de la plataforma se utilizó como referencia una tesis publicada en la Facultad de Ingeniería [11]. En ella se encuentra una similitud en cuanto al proceso de ETL utilizado para obtener la información de Moodle. El mismo detalla el uso de la API proporcionada por el sistema y los distintos servicios que brinda para obtener la información de interés en el proyecto. Si bien el uso de los datos que se plantea en la tesis es distinto al que se realiza en esta, se asemeja en gran manera la obtención de los mismos.

2.2.1. Learning Analytics

Learning Analytics se centra en la recopilación, medición, análisis y presentación de datos de aprendizaje. Su objetivo no es solo obtener conocimiento y visión en el progreso de alumnos, sino también evaluar la experiencia de aprendizaje

en general, justificar la efectividad de cualquier intervención de aprendizaje en tiempo real y mejorar el resultado del aprendizaje.

El propósito es informar y capacitar a los alumnos, instructores y organizaciones sobre el rendimiento y el logro de objetivos, evaluar el uso y la eficacia de los recursos educativos y facilitar la toma de decisiones. En consecuencia, se busca proporcionar recomendaciones para mejorarlos. [29]

El enfoque de Learning Analytics ha evolucionado más allá del análisis habitual de datos numéricos sobre los alumnos, para analizar también datos cualitativos como lo son el análisis de sentimientos, las motivaciones o la creatividad que muestran los alumnos durante su proceso de aprendizaje.

En la actualidad, la característica principal de los enfoques de Learning Analytics es que profundizan en el entorno multifacético, caracterizado por una gran cantidad de datos de múltiples contextos, dispositivos y procesos, y hacen amplio uso de técnicas de inteligencia artificial. Quienes toman decisiones en los centros educativos, incluidos los docentes, utilizan gradualmente la gran cantidad de datos producidos en entornos de aprendizaje. Estos dan origen al concepto de Learning Analytics.

Su aplicabilidad posee ventajas para todos aquellos involucrados en la educación. En el caso de los educadores, la disponibilidad del desempeño en tiempo real de los alumnos puede ser de gran ayuda en la planificación de actividades docentes. Para los alumnos, recibir información sobre su desempeño en relación con sus objetivos personales puede ser motivador y alentador. Finalmente, para los administradores y aquellas personas que toman decisiones dentro del centro educativo, los ayuda a saber cómo asignar recursos, desarrollar ventajas competitivas, y lo más importante, mejorar la calidad y el valor de la experiencia de aprendizaje. [38]

Las nuevas tecnologías han revolucionado el mundo del aprendizaje, ya que ahora es posible recopilar una gran cantidad de datos que se generan a partir de la interacción de los usuarios y las plataformas de aprendizaje a través de internet. Hoy en día se ha dado un paso más y ya no es suficiente conseguir la información, sino interpretarla y transformarla en elementos útiles para el área educativa. [22]

2.2.2. Conclusión de los trabajos relacionados

El estudio de trabajos relacionados con las áreas de conocimiento vinculados a este trabajo de tesis contribuyó a la preparación previa al desarrollo del proyecto. Gracias a esta investigación se tuvo más en claro cómo hacer uso de la API ofrecida por la plataforma Moodle [24]. Esto resultó de gran ayuda ya que se utilizaron varias funcionalidades de la API para realizar la extracción de la información que *Ánima* necesitaba obtener.

Por otro lado, si bien no se llegó a profundizar en Learning Analytics, es de valor tener una introducción sobre el mismo dado que *Ánima* busca extraer reportes con los datos de todos sus sistemas. De esta forma el centro puede adaptar su formación a los alumnos y sugerir nuevos métodos de aprendizaje. Es por esto que la investigación colaboró con detalles para la solución planteada en este proyecto.

2.3. Data Warehouse

Un Data Warehouse es una base de datos que se caracteriza por su capacidad de almacenar los datos a fin de lograr análisis que apoyen la toma de decisiones.

A fin de analizar los datos de forma más eficiente que las bases de datos convencionales y tomar decisiones a partir de estos, es que se parte de un paradigma conocido como OLAP. Este paradigma está enfocado en consultas de bases de datos orientadas a obtener diversos análisis y soportar grandes cantidades de datos.

Data Warehouse y OLAP se basan en un modelo multidimensional que representa los datos en Hipercubos. Sus Dimensiones o ejes corresponden a distintas perspectivas o criterios de análisis y las celdas a las Medidas o valores a analizar. Además de ser usados con el paradigma OLAP, los Data Warehouse son usados para soportar otras tareas de análisis, como obtener reportes, realizar análisis estadísticos, Data Mining, entre otros. Más adelante en la Sección 2.3.1 se brindan más detalles sobre las Dimensiones, las Medidas y el modelo multidimensional. Y en la Sección 2.3.2 se detalla sobre OLAP y sus operaciones. [43]

Usualmente los Data Warehouse son grandes repositorios que almacenan datos obtenidos de diferentes fuentes. Desde un punto de vista técnico un Data Warehouse está definido como una colección de datos orientados a las necesidades de análisis de la organización, integrados, no volátiles y variables en el tiempo. [43]

Los datos son integrados dado que provienen de varios sistemas, esto implica resolver problemas en base a diferencias en su definición y contenido: su formato, codificación, sinónimos (campos con diferentes nombres, pero son los mismos datos), homónimos (campos con mismos nombres, pero diferente significado), multiplicidad de ocurrencias de los datos, y muchos otros. Asimismo, los datos no son volátiles, lo que significa que su durabilidad se garantiza al no permitir su modificación y eliminación, expandiendo así su alcance a un período de tiempo típicamente de cinco a diez años. Esto a diferencia de las bases de datos operacionales que frecuentemente guardan sus datos por un período de tiempo corto (de dos a seis meses) requerido para operaciones diarias y son sobrescritos cuando es necesario. Por último, que los datos sean variables en el tiempo indica la posibilidad de registrar distintos valores para la misma información, así como también la fecha de cuando esos cambios ocurren. [43]

En el Capítulo 6 se presenta el diseño lógico de Data Warehouse que se sigue en este proyecto como también la arquitectura de este.

2.3.1. Modelo Multidimensional

El modelo multidimensional considera que los datos consisten en Medidas que vinculan varias Dimensiones [43]. A partir de este modelo se puede realizar el diseño conceptual de un Data Warehouse, en particular se menciona una especificación de modelos multidimensionales llamada CMDM.

Dimensiones

Dado que el modelo multidimensional permite ver los datos como Hipercubos y que también se debe extraer estratégicamente información de estos, es

necesario ver los datos en muchos niveles de detalle. Las Dimensiones de los Hipercubos son perspectivas usadas para analizar los datos y contienen Niveles que representan la “granularidad” o nivel de detalle en el que se representan las Medidas. Se le denomina Jerarquía al mapeo que relaciona conceptos detallados de Nivel inferior con conceptos más generales de Nivel superior dentro de una Dimensión. [43]

Medidas

Las Medidas son valores numéricos y se encuentran en la intersección de las Dimensiones del Hipercubo. Cada Medida en el Hipercubo está asociada con una función de agregación que combina muchos valores en uno solo. La agregación de las Medidas toma lugar cuando se cambia el nivel de detalle de los datos del Hipercubo que se están visualizando. Esto se realiza al atravesar las Jerarquías de las Dimensiones. [43]

Las Medidas pueden clasificarse según la forma en que son agregadas de la siguiente manera:

- **Medidas aditivas:** pueden ser agregadas a lo largo de todas las Dimensiones, utilizando la suma. Este es el tipo de Medidas más común.
- **Medidas semi aditivas:** pueden ser agregadas utilizando la suma en algunas Dimensiones, pero no en todas.
- **Medidas no aditivas:** no se pueden agregar mediante la suma en ninguna Dimensión. Algunos ejemplos típicos son el precio de artículos y el tipo de cambio.

Por tanto, para definir una Medida es necesario determinar las funciones de agregación que se utilizarán en las distintas Dimensiones. Esto es particularmente importante en el caso de Medidas semi aditivas y no aditivas. Por ejemplo, una Medida semi aditiva que representa las cantidades de inventario se puede agregar calculando el promedio a lo largo de las Dimensiones relacionadas al tiempo y calculando la suma a lo largo de otras Dimensiones. El promedio también se puede utilizar para agregar Medidas no aditivas como el precio de artículos o el tipo de cambio. Sin embargo, dependiendo de la semántica de la aplicación, se podrían utilizar en su lugar otras funciones (como mínimo, máximo o conteo). [43]

Hipercubos

Dentro de un modelo multidimensional la realidad se modela como un conjunto de Hipercubos. Cada Hipercubo está formado por un conjunto de Dimensiones y un conjunto de Medidas. [43]

CMDM

CMDM es un modelo que permite la especificación detallada de una base de datos multidimensional. Está basado en conceptos aceptados por los modelos multidimensionales y a su vez es independiente de las implementaciones. Dada una realidad, CMDM busca especificar la misma en términos multidimensionales.

La representa de manera directa, expresiva y precisa mediante dos lenguajes, uno gráfico que permite describir las estructuras de datos y algunas restricciones de integridad. Y otro con restricciones de integridad que permite dar una descripción precisa de las relaciones entre los datos.

El objetivo de CMDM es especificar bases multidimensionales y eso significa especificar datos para OLAP y Data Warehouse. A su vez, se propone nuevas estructuras llamadas Relaciones Dimensionales, más abstractas que los Hipercubos tradicionales. Estas estructuras permiten la especificación de conjuntos de Hipercubos con características comunes. De esta forma el modelo provee una estructura que permite manejar diferentes niveles de abstracción. En resumen, CMDM presenta tres estructuras básicas que serán descritas a continuación: Niveles, Dimensiones y Relaciones Dimensionales. [3]

Un Nivel representa un conjunto de objetos que son de un mismo tipo. Cada Nivel debe tener un nombre y un tipo. Para representar el esquema de un Nivel se utiliza un rectángulo que contiene el nombre y la estructura (o el nombre) del tipo de ese Nivel. En la Figura 2.1, se pueden ver tres Niveles. El primero representa un conjunto de vendedores de los que se conoce un determinado número de atributos. Análogamente, el segundo representa un conjunto de vehículos y el tercero un conjunto de clientes. [3]

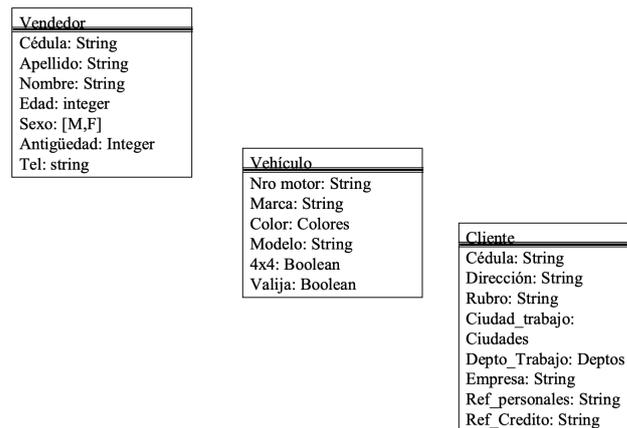


Figura 2.1: Tres Niveles [3].

Una Dimensión está determinada por una Jerarquía de Niveles. La instancia es una Jerarquía de elementos de esos Niveles y se tiene una relación 1-N (uno a varios) entre elementos de Nivel superior e inferior. De esta forma, el esquema de una Dimensión está representado por un rectángulo dentro del cual aparece un nombre para la Dimensión y un grafo dirigido en donde los nodos son los Niveles que participan de esa Dimensión como se puede ver en la Figura 2.2. [3]

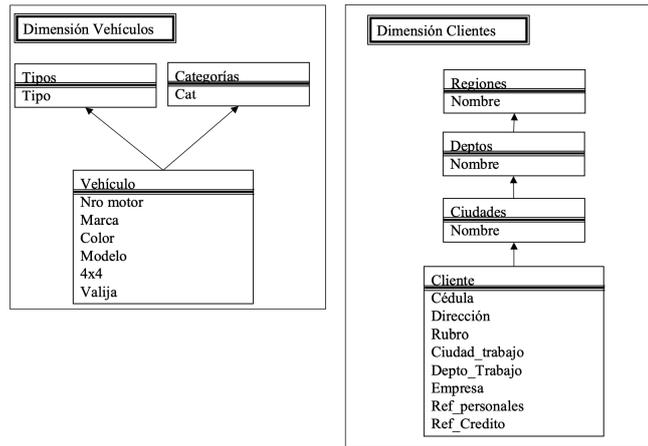


Figura 2.2: Dimensiones Clientes y Vehículos [3].

Una Relación Dimensional representa el conjunto de todos los Hipercubos que se pueden construir a partir de los Niveles de un conjunto dado de Dimensiones. Por lo tanto, el esquema de una Relación Dimensional está dado por un grafo en forma de estrella. El nodo central es de forma oval y tiene el nombre de la Relación Dimensional y los nodos “satélite” son rectangulares y tienen el nombre de cada una de las Dimensiones que participan de la relación. [3] En la Figura 2.3 se contempla un ejemplo de Relación Dimensional.

Por otro lado, un esquema multidimensional está formado por un conjunto de Relaciones Dimensionales que pueden compartir Dimensiones entre sí [3].

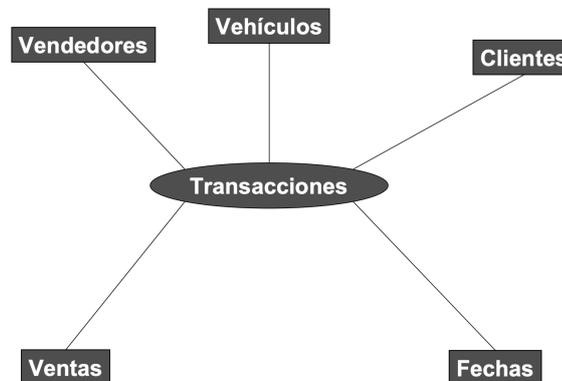


Figura 2.3: Relación Dimensional para Representar las Transacciones [3].

2.3.2. OLAP

Una gran variedad de sistemas y herramientas pueden usarse para acceder, analizar y explotar los datos de un Data Warehouse. Desde los comienzos de Data Warehouse el mecanismo típico para esas tareas ha sido OLAP. Los sistemas OLAP permiten a los usuarios consultar interactivamente los datos del Data Warehouse y aplicar funciones de agregación de forma automática sobre éstos. Estos sistemas facilitan la formulación de consultas complejas que pueden involucrar grandes cantidades de datos. Debido a estas características los usuarios

pueden acceder fácilmente a la información requerida y analizarla en varios niveles de detalle, así como encontrar patrones o tendencias de importancia para la organización. [43]

A pesar de que OLAP es utilizado en el contexto de Data Warehouse, este surgió por la necesidad de un nuevo paradigma específico para analizar la información en las bases de datos de las organizaciones a fin de apoyar la toma de decisiones. Se puede entonces concluir que el paradigma OLAP está enfocado en realizar consultas a las bases de datos a fin de analizar los datos, y típicamente estas consultas implican grandes cantidades de datos. [43]

Por otro lado, en los sistemas convencionales de paradigma OLTP las transacciones de base de datos, además de ser consultas, suelen ser de inserción, modificación y eliminación. Sin embargo, en este caso usualmente todas estas operaciones son en gran cantidad y manejan pequeñas cantidades de datos. Dadas las características de los sistemas OLTP su diseño debe evitar anomalías de actualización y, por lo tanto, sus bases de datos deben estar altamente normalizadas. Debido a esto, su rendimiento es deficiente al ejecutar consultas complejas que necesitan unir muchas tablas relacionales o agregar grandes volúmenes de datos. Además, las bases de datos operativas típicas contienen datos detallados y no incluyen datos históricos. [43]

Las bases de datos utilizadas en OLAP deben soportar consultas de grandes cantidades de datos. Esto se debe a que las consultas realizadas involucran funciones de agregación sobre los datos y, por lo tanto, la mayoría de las veces procesarlas requerirá atravesar todos los registros en una tabla. Las técnicas de indexación utilizadas en OLTP no son eficientes en este caso, por lo que nuevas técnicas de indexación y optimización de consultas son requeridas para OLAP. Resulta evidente que la normalización no es efectiva para estas consultas dado que divide la base de datos en muchas tablas y por lo tanto reconstruir los datos requiere un gran número de uniones de estas. [43]

La necesidad de un modelo de base de datos diferente para soportar OLAP es clara y conduce a la noción de Data Warehouse. Dado que los Data Warehouse son bases de datos dedicadas a análisis, estos se pueden diseñar y optimizar para soportar de manera eficiente consultas OLAP. Tanto Data Warehouse como OLAP se basan en el modelo multidimensional, capaz de visualizar los datos desde múltiples perspectivas y muchos niveles de detalle. A continuación, se presentan las principales operaciones OLAP que permiten que estas perspectivas y niveles de detalle sean materializadas por medio de la explosión de las Dimensiones y sus Jerarquías, proporcionando así un entorno de análisis de datos interactivo. [43]

- **Dice:** fija valores para algunas Dimensiones, atributos y Medidas.
- **Drill-across:** fusiona dos Hipercubos que tienen el mismo esquema e instancias mediante una condición de combinación.
- **Drill-down:** desagrega las Medidas a lo largo de una Jerarquía para obtener datos con una granularidad más fina. Es lo opuesto a la operación de Roll Up.
- **Drill-through:** muestra datos en los sistemas operativos de los que se derivó el Hipercubo. Esto es para acceder a datos descriptivos o seguir la traza de un dato.

- **Pivot:** gira los ejes de un Hipercubo para proporcionar una presentación alternativa de sus datos.
- **Slice:** elimina una Dimensión de un Hipercubo fijando un único valor en un Nivel de la Dimensión.
- **Roll Up:** agrega Medidas a lo largo de una Jerarquía para obtener datos con una granularidad más gruesa. Es lo opuesto a la operación de Drill-down.

2.3.3. Diseño Lógico de Data Warehouse

Para realizar una implementación de un modelo multidimensional pueden usarse distintos enfoques dependiendo de cómo se almacenen los Hipercubos. En esta sección nos basaremos en un enfoque conocido como ROLAP. Este almacena los datos en bases de datos relacionales, soporta extensiones a SQL y también métodos de acceso especiales para implementar de forma eficiente el modelo multidimensional y las operaciones OLAP. A su vez, existen otros enfoques como MOLAP y HOLAP. En MOLAP se almacena la información en estructuras de datos multidimensionales y se implementa las operaciones OLAP para estas estructuras, y HOLAP combina el enfoque MOLAP y ROLAP. [43]

Dentro del enfoque ROLAP existen varias representaciones relacionales del modelo multidimensional. Algunas de estas son el esquema Estrella (utilizado en este proyecto), el esquema Copo de Nieve y el esquema Copo de Estrella. [43]

ROLAP

En los sistemas ROLAP los datos multidimensionales son almacenados en tablas relacionales, y para aumentar el rendimiento, las agregaciones son calculadas previamente en esas tablas. Las agregaciones y las estructuras de indexación ocupan un gran espacio de la base de datos. Entonces las operaciones OLAP deben realizarse en tablas relacionales produciendo sentencias SQL normalmente complejas, ya que los datos multidimensionales residen en estas tablas. [43]

Algunas ventajas de los sistemas ROLAP es que gestionan todos sus datos en base al manejador de bases de datos relacional subyacente, los datos quedan estandarizados correctamente y se cuenta con una gran capacidad de almacenamiento. [43]

Esquema Estrella

En este esquema se puede encontrar una tabla de Hechos central, y un conjunto de tablas Dimensionales, una por cada Dimensión. La tabla de Hechos contiene una Foreign Key por cada tabla Dimensional relacionada y también contiene las Medidas correspondientes. Usualmente dentro del esquema Estrella las tablas de Dimensión no están normalizadas. Por ello es que pueden contener datos redundantes, especialmente en presencia de Jerarquías.

Las tablas de Hechos, sin embargo, generalmente están normalizadas. Su clave primaria es la unión de las Foreign Keys a las tablas de Dimensiones y su unión determina a las Medidas que contiene. En la Figura 2.4 se observa un ejemplo

particular del esquema Estrella donde la tabla de Hechos “Sales” está normalizada y su clave está compuesta por “ProductKey”, “StoreKey”, “PromotionKey” y “TimeKey”. [43]

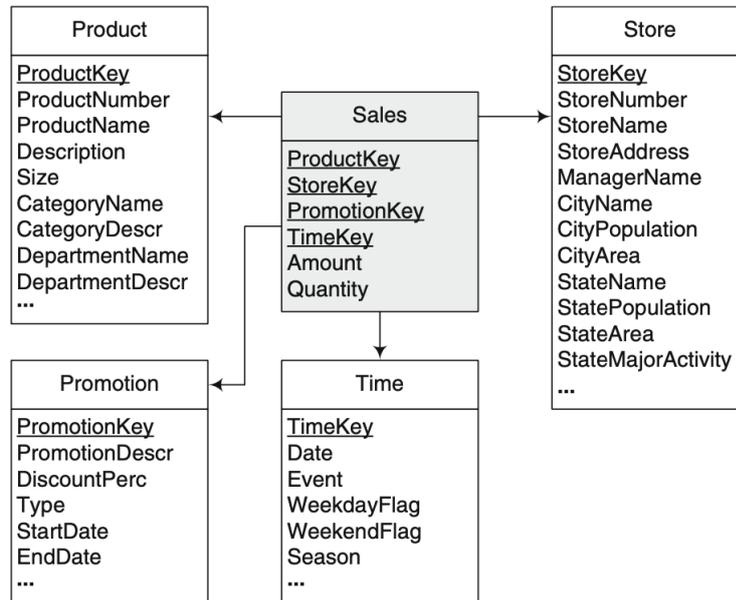


Figura 2.4: Un ejemplo de Esquema Estrella [43].

2.3.4. Arquitectura de Data Warehouse

La arquitectura de un Data Warehouse tiene varias capas. En primer lugar, la capa de Back end está compuesta por herramientas de ETL utilizadas para insertar los datos en el Data Warehouse desde diversas fuentes de datos. Por otro lado, se encuentra la capa de Data Warehouse compuesta por un Data Warehouse empresarial y/o Data Marts y una base de datos para metadatos que guarda información sobre el Data Warehouse y su contenido. También se cuenta con una capa denominada OLAP compuesta por un servidor OLAP que provee una vista multidimensional de los datos. Por último, existe una capa de Front end encargada del análisis y visualización de los datos. Esta última contiene herramientas de cliente como OLAP, de reporte, de estadísticas y de Data Mining. En la Figura 2.5 se puede apreciar una arquitectura típica de un Data Warehouse. En la zona izquierda de la imagen se encuentran las fuentes de datos a integrar y en la derecha las capas en la arquitectura. [43]

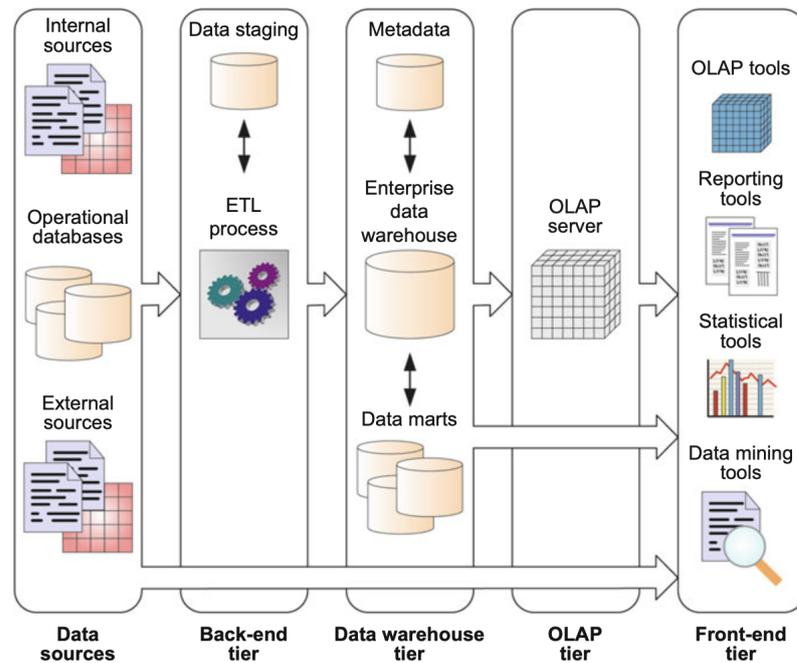


Figura 2.5: Típica arquitectura de Data Warehouse [43].

2.4. Moodle

Es una plataforma de aprendizaje diseñada para proveer a los docentes, administradores y alumnos un ambiente de aprendizaje personalizable. Se utiliza en todas partes del mundo por grandes instituciones y organizaciones académicas. Es gratuito y de código abierto por lo que se puede modificar y extender de forma libre, para su uso personal o comercial sin la necesidad de pagar licencias. [40] Su función es ofrecer servicios en la web, generando una interacción didáctica entre los alumnos y los docentes sin necesidad de estar físicamente en el mismo sitio. Dentro de las herramientas que ofrece se encuentra la posibilidad de dialogar a través de foros y mensajes privados, plantear tareas, entregar trabajos, evaluar, entre muchas otras. Para el desarrollo de este proyecto se utiliza la API de Moodle [24], en el Capítulo 7.1 se detalla la utilización de esta para la extracción de datos necesaria.

Esta plataforma cuenta con grandes ventajas. Una de ellas es la posibilidad de emular el comportamiento de un aula sin tener que estar físicamente en clase. También permite la comunicación sin restricción de días u horarios.

Por otro lado, se puede identificar la desventaja de la pérdida de comunicación interpersonal directa para la realización de trabajos grupales y dificulta el seguimiento particular de los alumnos, entre otras.

2.5. Visualizadores de datos

Luego de extraída, procesada y obtenida la información de los distintos sistemas, se busca poder visualizar estos datos de manera sencilla y que sea intuitiva para el usuario final. Para esto existen muchos sistemas que a partir de distintas fuentes (como ser bases de datos o archivos excel) pueden generar reportes con

tablas, gráficas, o listas, a las cuales se les pueden aplicar filtros y campos de búsqueda. Algunos ejemplos de herramientas que se utilizan para esta actividad son Pentaho [31], RapidMiner [35], Grafana [17] y Google Data Studio [14]. Esto facilita el análisis de la información para luego poder ser evaluada o controlada.

Cada organización puede tener su propósito específico; en el caso de *Ánima* se busca poder analizar a sus alumnos y docentes además de poder tener toda la información centralizada y con un fácil acceso a ella. A modo de ejemplo, se puede detallar el proceso que deben realizar los docentes de *Ánima* para saber si un alumno realizó una entrega en Moodle. Deben en ese caso acceder a la plataforma, buscar el curso, luego encontrar la tarea y dentro de la misma localizar si ese alumno la ha realizado. Contando con la información necesaria, con un visualizador se podrían aplicar filtros por usuario y curso obteniendo de forma fácil y rápida la información deseada.

A continuación, se detalla con más profundidad el visualizador de datos Google Data Studio, dado que *Ánima* ya tenía conocimiento adquirido sobre la utilización de este. En particular *Ánima* se ve favorecida de gran manera por esta herramienta por ser gratuita, fácil de usar e intuitiva. Esto genera un gran valor para la institución, ya que a partir de la misma pueden tomar decisiones para mejorar tanto la calidad de la educación de los alumnos como evaluar el rendimiento de los docentes y la dinámica de los cursos.

2.5.1. Google Data Studio

Google Data Studio es una plataforma que conforma el grupo de servicios que provee Google de forma gratuita, y se puede acceder a estos servicios con una cuenta personal de Google. Se puede cargar a Google Data Studio la información generada por distintas plataformas como ser YouTube, Google Analytics, Google Ads, Google Surveys, Asana, entre muchas otras. También se puede obtener la información que se encuentra en fuentes de datos como bases de datos, Google Sheets, archivos CSV, entre otros. A partir de las distintas fuentes de datos se pueden generar informes y paneles personalizables a los que se les puede agregar gráficos de líneas, de barras y circulares; mapas geográficos; gráficos de área y de burbujas; tablas de datos paginados y tablas dinámicas, por nombrar algunos. A estos se les puede agregar filtros interactivos para filtrar por fecha o por alguno de los campos de la información ingresada. También se puede agregar imágenes, videos, textos, diferentes estilos y colores. Todas estas características ayudan al usuario a poder expresar de forma visual sus datos para luego poder exponer, analizar, y tomar decisiones sobre estos.

2.6. ETL

Dadas las características y requerimientos del proyecto planteado, se identificó que se debía obtener información a partir de distintos sistemas de información. Por esta razón, se utilizó un proceso ETL (Extract, Transform and Load por sus siglas en inglés), como bien lo describe su nombre, se encarga de la extracción, transformación y carga de la información. Estos procesos permiten obtener y mover datos de varios sitios y normalizar la información obtenida de forma que sea beneficioso para la organización. Por último, estos datos obtenidos y ya

transformados son cargados en un sistema como puede ser una base de datos o un Data Warehouse.

2.6.1. Extracción

Esta es la primera de tres etapas del proceso, y se basa en obtener la información de los distintos sistemas. Para la extracción de la información puede ser requerido diferentes enfoques, dado que estos pueden estar basados en distintas plataformas (como las bases de datos), distintos sistemas operativos y distintos métodos de comunicación (REST, SQL, etc).

Se pueden identificar dos etapas en el proceso de extracción: la extracción inicial y la extracción realizada para actualizar la información. La primera etapa se realiza una única vez y resulta por lo general en una gran cantidad de información. La segunda etapa puede ser realizada cada cierto intervalo de tiempo o al dispararse cierto evento definido por la organización. En esta segunda etapa se actualiza la información ya registrada. Para esto se pueden utilizar distintas técnicas, como generar logs, agregar fechas, auditar los registros, entre otras. [37]

2.6.2. Transformación

Esta es la segunda etapa del proceso ETL y es la que se encarga de transformar la información obtenida en la etapa anterior para que ésta sea correcta, consistente y completa. También se puede incluir procesos de limpieza de la información. Dichos procesos se encargan de convertir tipos de datos, eliminar o completar datos a partir del resto de la información o datos obtenidos de forma externa, estandarizar tipos de datos como ser fechas, entre otros. [37]

2.6.3. Carga

Una vez ya obtenida y transformada la información se procede a la carga de la misma en algún sistema. En esta etapa la información es alojada en una base de datos, un archivo plano, un archivo excel u otros sistemas como por ejemplo un Data Warehouse. [37]

Capítulo 3

Metodología de trabajo

En este capítulo se detalla el proceso de trabajo que se mantuvo a lo largo del proyecto, indicando las herramientas y metodologías utilizadas para la construcción de la solución.

Cabe aclarar que el proyecto se desarrolló de manera simultánea a la aparición del COVID-19, por lo que se debió ajustar la metodología de trabajo habitual.

3.1. Planificación

Al comenzar con el proyecto se planificaron ciertas actividades que se distribuyeron de manera coherente, tomando como fecha de inicio el mes de Abril de 2020 y fecha de finalización el mes de Diciembre de 2020. Cabe aclarar que dicha planificación fue realizada previamente a la pandemia del COVID-19 que afectó a todo el planeta. Entre las actividades planificadas se identificaron:

- **Relevamiento:** en esta actividad se incluyó el relevamiento de las necesidades de Anima, la comprensión de los sistemas de información utilizados por la institución y el desarrollo de una búsqueda de herramientas de Learning Analytics.
- **Diseño del Prototipo:** diseño de prototipo para la integración de la información de los distintos sistemas utilizados.
- **Validación Diseño:** validación del diseño del prototipo para la integración de la información.
- **Implementación:** desarrollo de la solución planteada.
- **Integración:** integración con herramientas de Learning Analytics.
- **Validación:** validación con la institución y pruebas de verificación.
- **Informe:** redacción del informe final del proyecto.

A cada una de estas actividades se les asignó un tiempo considerado según la tarea a desarrollar. Para ello se construyó un diagrama de Gantt, detallado en la siguiente Figura 3.1.

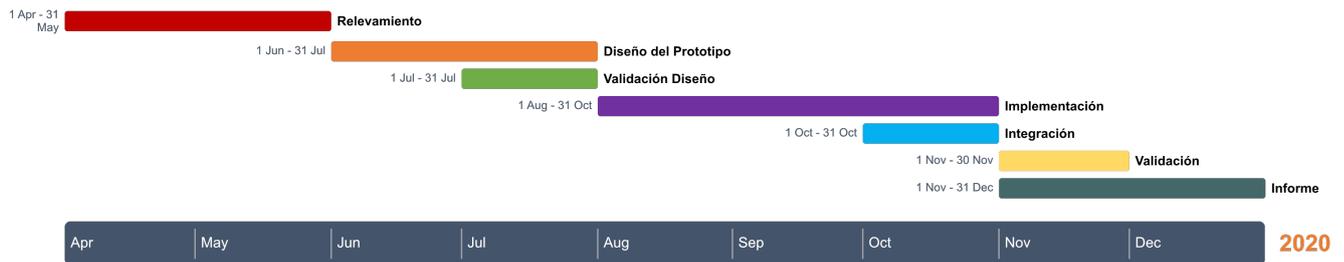


Figura 3.1: Planificación inicial

Como se mencionó anteriormente, la pandemia no permitió que la planificación inicial se realizara como estaba programada. Debido a que la educación estuvo detenida por un determinado período de tiempo, la actividad de relevamiento de requerimientos con *Ánima* se retrasó. A su vez, las reuniones resultaron más esporádicas de lo deseado al realizarse mediante video llamada y también se vio demorada la adquisición de los recursos requeridos para el desarrollo del proyecto. Otro factor influyente estuvo relacionado con la renuncia de la persona con la que se tenía comunicación por parte de la institución. Esto llevó a un cambio de referente, por lo que se tuvieron que valorar nuevos aspectos.

A partir de estas condiciones, se generó un nuevo diagrama de Gantt detallado en la siguiente Figura 3.2.



Figura 3.2: Planificación final

A continuación, se puede observar en la Tabla 3.1 una aproximación de la carga horaria llevada a cabo por el grupo de trabajo. La misma contiene desglosada las horas del trabajo realizado divididas por las tareas representadas en el diagrama de Gantt 3.2. A su vez muestra para cada tarea la carga horaria dedicada por todo el grupo semanalmente, como también la carga horaria total dedicada a la misma.

Tarea	Cantidad de semanas	Horas grupales aproximadas por semana	Total de horas grupales aproximadas
Relevamiento	27.6	3	82.8
Diseño de prototipo	4.2	30	126
Validación del diseño	14.6	3	43.8
Implementación	31.6	9	284.4
Integración	12.3	3	36.9
Validación	13.3	3	39.9
Informe	8.5	36	396
Total de horas aproximadas del proyecto			1009.8

Cuadro 3.1: Horas aproximadas del proyecto

Como se puede observar en la tabla, el grupo de trabajo dedicó aproximadamente **1009.8** horas durante el transcurso del proyecto. Dado que cada integrante del equipo tuvo una participación aproximadamente equivalente en cantidad de horas, se puede decir que cada miembro dedicó aproximadamente más de 300 horas al proyecto. Otra conclusión que se puede observar con un simple análisis es que se tuvo una gran cantidad de horas dedicadas a la implementación del proyecto y al diseño del prototipo como estaba planificado al inicio del proyecto **3.1**.

3.2. Relevamiento de requerimientos

Para el relevamiento de requerimientos se realizaron reuniones virtuales, de las que participaron los representantes de *Ánima*, la tutora del proyecto de grado y los representantes del grupo. A partir de estas se obtuvo información sobre los problemas que se presentaban en la institución, la forma en la que trabajan y sus preferencias buscando cumplir con el primer objetivo del proyecto. La institución brindó diferentes archivos excel, de los que se obtuvieron una gran parte de los requerimientos. Estos archivos contenían información recabada de los distintos sistemas informáticos utilizados en *Ánima*, lo que permitió cargar la información en la plataforma Google Data Studio y le facilitó a la institución la realización de sus reportes de cuadros de mando. Asimismo, fue una gran oportunidad para que el equipo se familiarizara con Google Data Studio, ya que no poseía conocimiento de esta herramienta.

Por otro lado, se relevaron los requerimientos que *Ánima* precisaba con respecto a la plataforma Moodle, que hasta el momento no brindaba ningún tipo de información dado que no contaban con una manera viable de obtenerla. Asimismo, dicha información no se consideraba tan relevante, ya que al tener clases presenciales el uso que los alumnos y docentes le daban a Moodle era mínimo. Sin embargo, a partir de la pandemia esto cambió ya que los cursos se comenzaron a dictar de forma virtual. Es así como Moodle se convirtió en la herramienta principal para el dictado de los cursos y la evaluación de los alumnos por parte de los docentes, lo que llevó a que el uso de la plataforma aumentara exponencialmente. Dadas todas estas condiciones, quienes llevan un control de los cursos debían realizarlo a través de Moodle. Esto implicaba que debían ingresar en cada

uno de ellos manualmente y verificar si el docente había publicado material para los alumnos y si éstos habían tenido participación.

3.3. Herramientas utilizadas

En todo el transcurso del proyecto, el equipo decidió utilizar diferentes herramientas para la administración, organización, comunicación y versionado del proyecto.

3.3.1. Trello

Dicha plataforma sirve para la asignación y gestión de las tareas de un proyecto. Cabe aclarar, que una tarea es una instancia de trabajo la cual consta de un título, una descripción, un estado y un usuario asignado a la misma. En Trello se pueden crear tableros, los cuales contienen columnas que corresponden a estados en los que puede transitar cada una de las tareas a realizar. Para el caso de este proyecto los estados utilizados fueron:

- **Recurrentes:** tareas que se realizaron periódicamente como ser, reuniones con *Ánima* o con la tutora del proyecto.
- **Por definirse:** tareas que no estaban totalmente definidas por falta de consentimiento de alguna de las partes (integrantes del equipo, *Ánima* o la tutora del proyecto).
- **Para realizar:** tareas que estaban prontas para ser asignadas.
- **En proceso:** tareas que estaban siendo realizadas por algún miembro del equipo, a las mismas se les asignaba un usuario específico.
- **Realizado:** conjunto de tareas que ya fueron finalizadas y validadas por algún miembro del equipo.

En esta plataforma se crearon tareas como ser: investigar ciertas tecnologías que fueran a ser utilizadas, desarrollar un módulo del proyecto, agendar o participar de una reunión, entre otras. La razón por la cual se decidió utilizar una plataforma de este estilo fue para ayudar en la organización de las tareas e ideas y mejorar la coordinación del equipo para poder tener registrado el trabajo realizado, ya que el transcurso del proyecto fue totalmente remoto. A su vez esta plataforma fue elegida debido a que es gratuita, flexible y conocida por los miembros del proyecto. [41]

3.3.2. Git

Es un sistema de control de versiones distribuido diseñado para trabajar en todo tipo de proyectos. Este sistema fue seleccionado para trabajar en el desarrollo del proyecto, de forma de poder tener un control de versionado del mismo, poder lograr una sinergia del equipo en el momento de desarrollo y beneficiarse de varias de las características que ofrece como ser la creación de ramas, entre otras. Dentro de las distintas plataformas que ofrece este sistema, se utilizó GitHub, dado que todo el equipo trabaja diariamente con esta plataforma. Otro

factor que se tomó en cuenta es que gracias a los beneficios que esta plataforma otorga a los alumnos se pudo trabajar con un proyecto privado proporcionando seguridad a los datos de la organización. [12]

3.3.3. Google Drive

Servicio de Google donde se puede almacenar todo tipo de documentos y archivos, además de poder crear documentos y trabajar en los mismos de forma simultánea entre varios usuarios. En el proyecto se utilizó para almacenar toda la información. En particular se crearon documentos para registrar requisitos, información importante de las reuniones, guías, documentación, investigaciones realizadas, entre otras. Así como se almacenaron documentos entregados por *Ánima*, como ser los excel de prueba, entre otros. [15]

3.3.4. Google Meets

Esta herramienta ofrecida por Google permite realizar video llamadas sin límite de tiempo, en las cuales pueden participar varios usuarios. Fue de suma importancia utilizar este tipo de plataforma, ya que el proyecto se desarrolló de forma remota. Es decir, que las reuniones de equipo, las reuniones con la tutora del proyecto y con los integrantes de *Ánima* se realizaron a través de videollamadas.

Esta plataforma fue elegida no solo por ser gratuita, sino también por no contar con límite de tiempo de llamada. Asimismo, se tomó en cuenta que es muy conocida, presenta una intuitiva conexión con el calendario de Google y se puede acceder simplemente con una cuenta de Google (con la que contaba cada uno de los interesados), lo cual facilitó su adopción. [16]

3.4. Reuniones de avance

A lo largo del proyecto se realizaron varias reuniones con distintas finalidades. Los diferentes tipos de reuniones se pueden categorizar en:

- **Reuniones internas del grupo de trabajo:** en estas reuniones participaron únicamente los integrantes del grupo. Son las reuniones que se utilizaron para actualizar los avances del proyecto de cada integrante.
- **Reuniones con la tutora del proyecto:** de dichas reuniones participaron los integrantes del grupo de trabajo y la tutora del proyecto. Eran utilizadas para mantener informada a la tutora de los avances obtenidos en los distintos aspectos del proyecto, comentar aspectos de las reuniones con los encargados del proyecto en *Ánima*, plantear problemáticas a resolver, entre otras cosas.
- **Reuniones con *Ánima*:** en las mismas participaron los integrantes del grupo de trabajo y los encargados del proyecto en *Ánima*. En ciertas ocasiones también participó la tutora del proyecto para poder realizar un seguimiento más detallado del proyecto. Dada la importancia de estas reuniones en la Subsección 3.4.1 se detallan más aspectos de las mismas.

La periodicidad de las reuniones fue variada ya que eran con distintos fines. Las reuniones internas del grupo solían ser los viernes o fines de semana de cada semana para hacer una puesta a punto de lo que venía trabajando cada integrante del grupo más allá de que muchas veces algunos de los integrantes (a veces todos) trabajaban en conjunto. Por otro lado, las reuniones con la tutora del proyecto solían ser antes de tener una reunión de validación con *Ánima*, para organizar bien los temas a tratar además de mostrar los avances conseguidos por el grupo. Por último, las reuniones de validación con *Ánima* solían ser una vez por mes o mes y medio para validar los avances del grupo.

3.4.1. Reuniones con *Ánima*

El método que se aplicó durante el proyecto estuvo basado en metodologías ágiles, lo que llevó a que el equipo mantuviera reuniones periódicas con representantes de la institución para validar los avances obtenidos. Estos avances se conformaban de una nueva funcionalidad en la que el equipo había trabajado durante un período de tiempo y luego de creerse finalizada o lo bastante avanzada, se organizaba una reunión de ese tipo para demostrar su funcionamiento. Con esta reunión se buscaba la aprobación por parte de la institución, y de ser necesario plantear alguna mejora o cambio a la funcionalidad en cuestión, ya que ellos serían el usuario final del prototipo de extracción de Moodle construido. De dichas reuniones se obtuvo muy buena retroalimentación, ya que posibilitaron detectar cambios en etapas tempranas del desarrollo y así afectar en menor medida el proyecto. Se trató de ajustar cada funcionalidad a lo que la institución consideró que le sería de mayor utilidad, ya que no cuentan con personal que pueda realizarle cambios al prototipo una vez finalizado.

3.5. Proceso de desarrollo

El proceso de desarrollo se vio solapado con el relevamiento de requerimientos, dado que el comienzo de este último se vio demorado y luego de iniciado transcurrió de forma lenta. Por lo tanto, en un principio se comenzó a investigar y probar distintas soluciones y tecnologías con la información inicial del proyecto. A partir de las pruebas realizadas se tomó la decisión de dividir el proyecto en dos grandes áreas de trabajo, y cada integrante del equipo tomó una participación específica en estas. Las áreas definidas fueron: la extracción de datos de Moodle (objetivo 2 del proyecto) y la integración de los datos de los sistemas utilizados en *Ánima* (objetivo 3 del proyecto). Estas tareas se fueron desarrollando con el correr del tiempo en simultáneo dado que no tenían dependencias entre sí.

3.5.1. Obtención de información de Moodle

Como resultado del segundo objetivo del proyecto se implementó un prototipo conformado por un sitio web y un servidor encargado de la obtención de información de Moodle. Para esto se decidió utilizar como tecnología base del servidor Typescript [42] con NodeJs [28], debido a que parte del equipo estaba familiarizado con las mismas. Asimismo, se contaba con un proyecto [11] de ejemplo

que estaba desarrollado con las mismas tecnologías. Luego de avanzar con el análisis de requerimientos e identificar que se iban a tener que generar archivos excel, se investigó si era posible con la tecnología base seleccionada. Durante la investigación se encontró una librería que cumplía con todos los requerimientos necesarios para poder realizar el manejo de archivos excel como ser, creación del documento, creación de hojas, agregar celdas, entre otras cosas. El desarrollo se ideó de forma de que sea lo más flexible y modularizable posible, lo cual genera la capacidad de poder agregar en un futuro módulos independientes que se encarguen de la extracción, transformación y carga del resto de los sistemas que utiliza *Ánima* y que hoy en día está realizando de forma manual.

Cabe destacar también que el proyecto está fuertemente relacionado con la plataforma Moodle, dado que uno de sus objetivos es la extracción y análisis de la información de dicha plataforma. Se buscó no depender de la instancia de Moodle utilizada, siendo esta una instalación de Moodle en el servidor de *Ánima*. De esta forma, si en un futuro *Ánima* decide cambiar su instancia de Moodle, cambiar la versión utilizada, cambiar el dominio del sitio utilizado actualmente o reutilizar este mismo prototipo en otra institución, lo va a poder realizar generando mínimos cambios de configuración en el proyecto y mantenimientos al prototipo en cuanto a los servicios utilizados de la API de Moodle.

Esta generalización con la que se llevó a cabo el proyecto también facilitó el proceso de desarrollo, ya que en un principio no se tenían los permisos requeridos para poder extraer la información desde el Moodle de *Ánima*, por lo tanto, se crearon instancias de prueba que ofrece Moodle de forma gratuita para comenzar con el desarrollo. Luego de obtener las credenciales, se continuó probando ciertas funcionalidades en estos sitios de prueba, ya que de esta forma se prevenía que alguna de estas pruebas afecte negativamente a la plataforma que está siendo utilizada actualmente por *Ánima*.

Luego se comenzó a implementar el sitio web, el cual se encarga de interactuar con el servidor antes descrito para poder descargar el archivo excel que contiene información obtenida de Moodle, de una forma amigable para los usuarios de *Ánima*. Para esta implementación se decidió utilizar Javascript [21] y React [36] como tecnologías base para su desarrollo.

Llegado su momento, *Ánima* proporcionó los recursos necesarios para el desarrollo del proyecto, estos son el usuario para ingresar en la plataforma de Moodle y el servidor donde se aloja el sistema implementado, el cual es una instancia EC2 de Amazon con sistema operativo Ubuntu. Esto permitió a corroborar que la implementación funcionaba correctamente en el entorno de producción y permitía que se hiciera entrega a *Ánima* del sistema desarrollado para ser probado y evaluado de forma directa.

Una vez asignados los permisos de usuario necesarios sobre la plataforma de Moodle de *Ánima*, se pudo probar el desarrollo realizado hasta el momento de la obtención de la información. Hecho esto, se realizaron cuadros de mando en la plataforma Google Data Studio, ya que es el método utilizado en *Ánima* para visualizar la información. A través de los cuadros de mando se procedió a validar la información de una manera accesible con representantes de la institución.

3.5.2. Integración de los datos

Al mismo tiempo que se realizaron los distintos desarrollos para la extracción de Moodle, se obtuvieron los requerimientos para la integración de los datos proveniente de los distintos sistemas usados por *Ánima*. Luego de algunos análisis que se presentan en el Capítulo 4 en la Subsección 4.2.4 se decidió diseñar un Data Warehouse para alojar dichos datos. Este diseño está basado en todas las fuentes de datos utilizadas por *Ánima*, los cuadros de mando existentes y los requerimientos obtenidos en las reuniones con la institución como resultado del primer objetivo del proyecto. De los cuadros de mando se analizaron los filtros de la información, así como el agrupamiento de la misma en los gráficos. Todo esto fue de gran utilidad ya que a partir de este análisis se obtuvieron los indicadores de interés para la institución y de qué conceptos o criterios dependían los mismos. Con la información recabada se pudieron crear los diagramas necesarios para el diseño del Data Warehouse, y a partir de este se realizó el Script de la base de datos.

3.5.3. Verificación

Para la verificación del sistema implementado se realizaron varios flujos de pruebas donde se verifican los datos obtenidos en los archivos excel con la información que se encuentra en la plataforma de Moodle de *Ánima*. Estos flujos se definieron buscando abarcar toda la información obtenida de la plataforma y se ejecutaron para una cantidad acotada de elementos seleccionados de forma aleatoria.

Por otro lado, como forma de verificación del diseño del Data Warehouse se decidió realizar una prueba de concepto o prototipo del mismo con los datos acotados únicamente a la plataforma Moodle de la institución. Para realizarlo se tuvo en cuenta desde un comienzo que la información de Moodle iba a estar disponible de la forma necesaria, ya que se podía obtener a conveniencia del equipo para ser manipulada. A partir de esta prueba se obtuvieron reportes que obtenían los datos directamente del prototipo del Data Warehouse implementado.

3.5.4. Versionado

Para el control de versionado del código implementado se utilizó la herramienta GitHub, la cual fue descrita anteriormente 3.3.2, para el manejo de esta herramienta se decidió utilizar el proceso Gitflow [9]. Este proceso es comúnmente utilizado para el control de versiones, ya que es independiente de la plataforma que se decida utilizar siempre y cuando esté basada en Git [12]. Gitflow propone un flujo de trabajo basado en la creación y mantenimiento de ramas, a las cuales se les asigna previamente el trabajo que deben cumplir en el flujo definido.

Las ramas utilizadas para este flujo son la rama "master" o "main", en esta se incluye el progreso ya finalizado que va a ser publicado. Luego está la rama llamada "develop", esta nace de la rama "master", y es la rama a la cual se le agrega la implementación de los requerimientos ya finalizados. Por último, se tienen las ramas de desarrollo, estas nacen de la rama "develop", se definen con un nombre característico del requerimiento que se esté implementando y se incluye

toda la implementación relacionada a este requerimiento. El flujo en sí consiste en la creación de ramas para cada una de las características a implementar, luego cuando una característica esté completada se crea un Pull Request apuntando hacia la rama “develop”. Un Pull Request es la forma que se utiliza para pedir a otro integrante del equipo que revise el código implementado para poder ser incluido en la rama destino. El integrante que realiza la revisión puede dejar comentarios en este y solicitar cambios, los cuales el que implementó el código va a tener que realizar antes de poder integrar su rama en la rama destino. Esta es una característica muy importante del flujo ya que permite al equipo realizar revisiones cruzadas del código, verificando de que se haya implementado lo que se requería y de forma correcta. Luego de que se incluyeron requisitos dentro de la rama “develop”, estos son probados de forma conjunta en un ambiente seguro para luego dar por finalizado el proceso de desarrollo e incluir estos cambios en la rama “master” generando que los mismos queden públicos.

3.5.5. Gestión

Durante todo el transcurso del proyecto se utilizaron distintas herramientas para la organización y gestión del mismo. Una de ellas fue Trello [41], como anteriormente se describió esta herramienta se encarga de gestionar las distintas tareas, a las cuales se les asigna un estado dependiendo de cuál es la acción que se está tomando en ese momento o cuál es la siguiente acción a tomar para una tarea específica. Esta herramienta fue de gran utilidad para la organización del trabajo y la división del mismo, ya que a pesar de la distancia física del equipo se podía saber en qué estaba trabajando cada integrante y se tenía una visión global del estado del proyecto para un momento dado.

Capítulo 4

Análisis y Diseño

En este capítulo se tratará el primer objetivo del proyecto y los requerimientos obtenidos como resultado detallando el análisis realizado paso a paso para entender la problemática que enfrentaba *Ánima*. A partir de los requerimientos resultantes, se presenta asimismo los detalles del diseño de la arquitectura lograda para dar solución al segundo y tercer objetivo de este proyecto.

4.1. Relevamiento de requerimientos

En busca de brindar solución al primer objetivo de este proyecto se realizó un análisis para comprender la situación de *Ánima*, el cual se dividió en tres etapas. Todas ellas se llevaron a cabo durante las primeras reuniones con los encargados del proyecto en *Ánima*. En un primer lugar se trató de comprender la problemática que estaban enfrentando en la institución. Luego se detallaron los distintos sistemas utilizados en *Ánima* y sus diferentes propósitos; a la vez que se presentaron los datos existentes que ya manejaban en la institución. Por último, se recabaron los requerimientos para solucionar la problemática planteada. Cada una de estas tres etapas serán explicadas en mayor profundidad en las siguientes subsecciones.

4.1.1. Problemática

Las primeras reuniones con los encargados del proyecto en *Ánima*, además de funcionar como presentación entre las partes, fueron un espacio donde el equipo se familiarizó con la problemática que atravesaba la institución hacia ya algún tiempo.

Ánima es una institución que hasta el 2020 llevaba cuatro años en funcionamiento. Hasta ese entonces habían utilizado una gran cantidad de sistemas informáticos a diario, los cuales les aportaban información útil para conocer el estado de situación de la institución en general. Entre los sistemas informáticos utilizados estaban SIGED, INOA, Evaluador, formularios de Google, archivos excel de la institución y Moodle.

Para lograr tener un análisis general de toda la institución, ya sea sobre la situación de los cursos, actividad de los docentes y alumnos, seguimientos de los egresados, u otros, *Ánima* realiza cuadros de mando en los que cuenta con toda la información relevante para tomar decisiones. Estos cuadros de mando se generan a partir de la información obtenida de todos los sistemas informáticos anteriormente mencionados (a excepción de Moodle), como se puede observar en el

diagrama de la Figura 4.1. La razón por la que no consideraban Moodle era porque la institución no contaba con una manera sencilla de obtener la información del mismo. Además, previo al 2020 Moodle no había sido usado habitualmente por alumnos y docentes como para obtener información que fuese relevante, ya que los cursos, como en cualquier institución educativa, se dictaban de manera presencial.

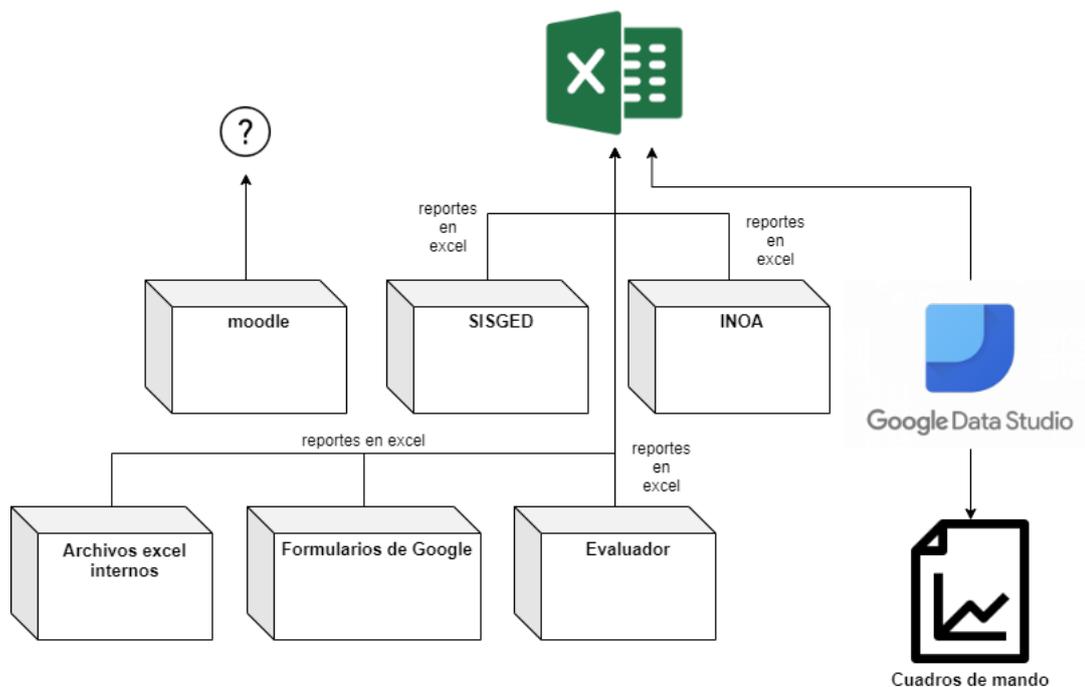


Figura 4.1: Solución utilizada por Ánima

Aunque esos cuadros de mando se generaban a partir de la información obtenida de varios sistemas informáticos, no se realizaba automáticamente. Era un proceso tedioso de realizar, ya que una persona encargada de manipular la información de los distintos sistemas debía unificarla en un solo archivo excel. Luego, con la herramienta Google Data Studio, se generaban los cuadros de mando a partir del archivo excel. Todo este proceso debía ser realizado cada vez que se obtenía nueva información al avanzar el año, ya que los cuadros de mando eran actualizados con información para cada entrega de boletines que se realizaba a los alumnos en Ánima. Esto se detectó como un primer problema a mejorar ya que podría existir alguna herramienta que facilitara la unificación de la información de todos los sistemas informáticos.

Llegado el 2020, las condiciones normales en que se dictaban los cursos en años anteriores cambiaron drásticamente. Dicha situación llevó a que todos los institutos de educación pasaran a dictar sus clases de forma virtual, incluido Ánima. Por esta razón es que en la institución empezó a acrecentarse el uso diario de Moodle: todos los materiales utilizados en clase, trabajos a realizar por los alumnos, correcciones de los docentes, entre otros, se comenzaron a realizar a través de esta plataforma. Las nuevas medidas evidenciaron que mucha información relevante para la nueva realidad no se estaba teniendo en cuenta por no tener ninguna herramienta para obtener dicha información. En consecuencia, se perdía el seguimiento de los cursos.

La persona encargada de llevar un control de los cursos en Moodle debía controlar que se desarrollaran normalmente al ingresar a la plataforma. Para esto, la persona encargada de dicha tarea debía fijarse si con el correr de los días el sitio de cada curso iba añadiendo nuevo contenido, si los foros eran utilizados tanto por alumnos como por docentes, si los trabajos planteados por los docentes eran realizados por los alumnos y si los trabajos realizados por los alumnos recibían una retroalimentación, entre otras cosas. Todo este trabajo debía ser realizado manualmente un curso a la vez, lo que demandaba demasiado tiempo. Es aquí donde se detectó el segundo problema a resolver, que resultaba de mayor necesidad en comparación al previo.

4.1.2. Datos existentes

Continuando el proceso de reuniones con *Ánima* se presentaron las fuentes de datos existentes. Los datos obtenidos de todos los sistemas informáticos (a excepción de Moodle) llegaban en forma de archivo excel, por lo que era necesario un gran trabajo para unificar la información en un único archivo. Además, era necesario una previa transformación de la información para obtenerla en un formato fácil de utilizar. Todo este proceso hecho a mano se asemejaba a un proceso de ETL muy rudimentario. De la Extracción se encargaban principalmente los proveedores de los sistemas informáticos, mientras que la información restante era información que *Ánima* tenía mediante encuestas de Google realizadas. El proceso de Transformación y Carga se llevaba a cabo por un miembro del personal de *Ánima* que realizaba una unificación de la información extraída en un solo archivo excel conteniendo la información de interés y en un formato adecuado. Por último, mediante Google Data Studio se creaban los cuadros de mando para visualizar de mejor manera la información, en las Figuras 4.3 y 4.2 se muestran algunos ejemplos.

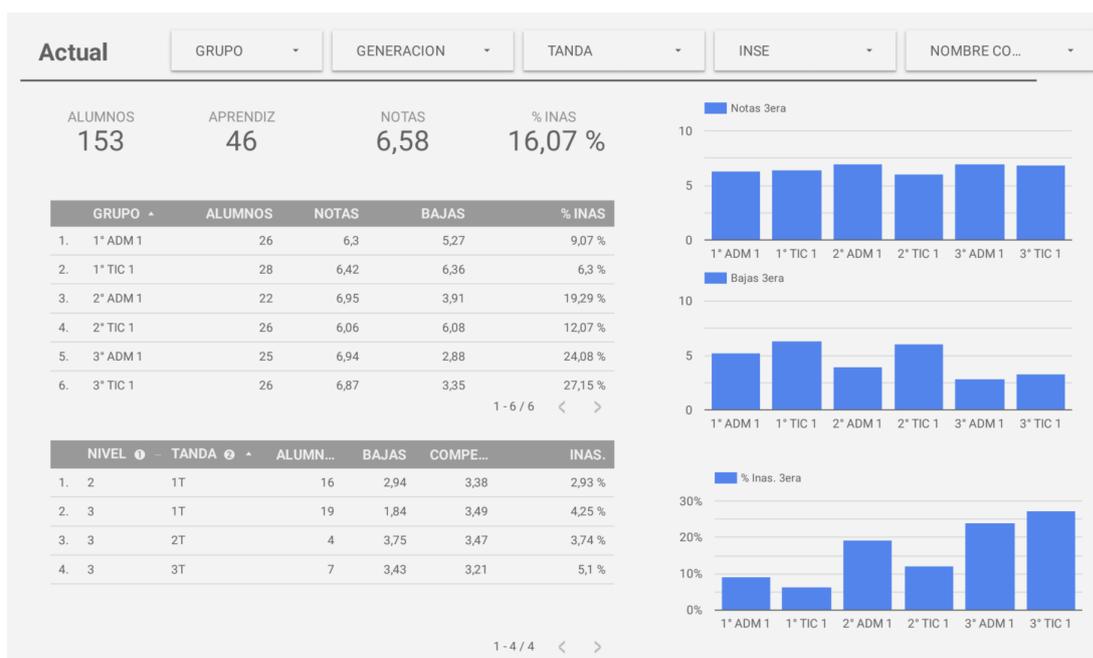


Figura 4.2: Cuadro de mando - Situación actual por grupo

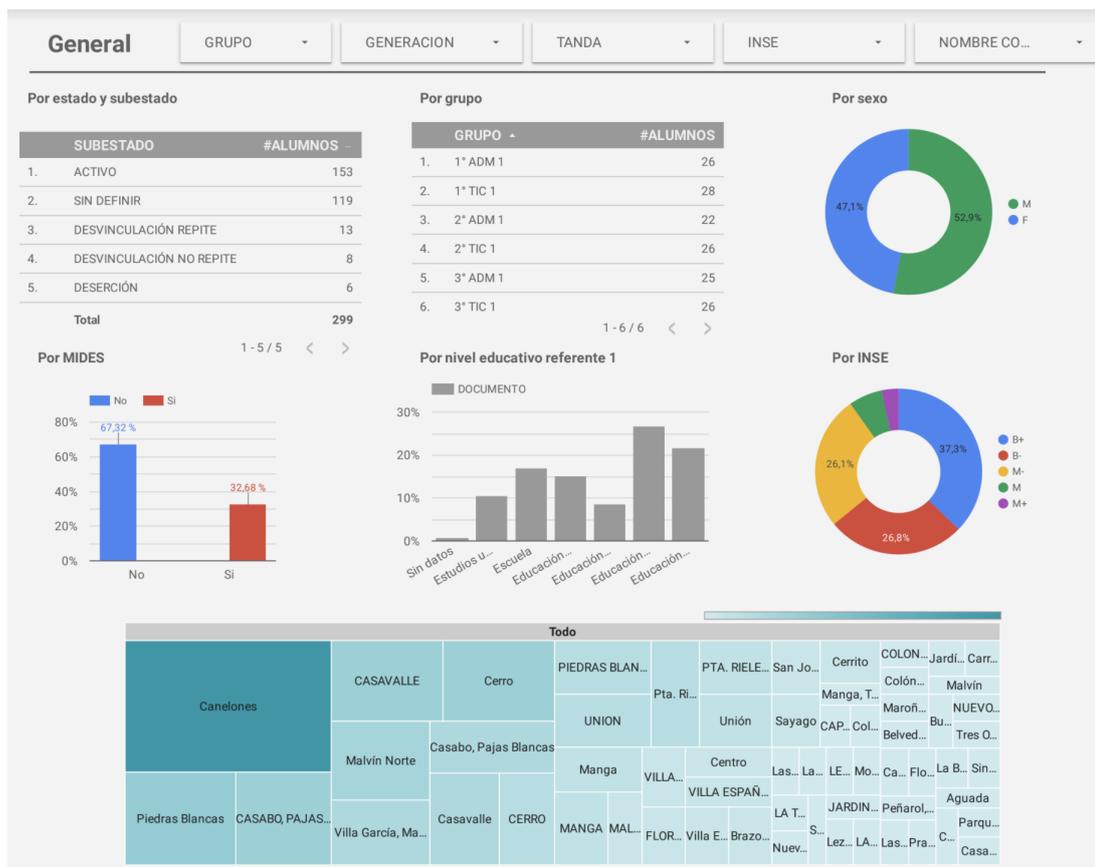


Figura 4.3: Cuadro de mando - Situación socioeconómica de los alumnos

Los cuadros de mando contienen información que se obtiene de cada sistema informático utilizado por Ánima, detallados a continuación:

- SIGED:** Ánima tiene la mayoría de sus datos en este sistema actualmente, incluidas las calificaciones, las asistencias y los cursos con sus alumnos y docentes. A través de estos datos se obtienen reportes con los porcentajes de asistencia, los promedios de calificaciones y promedio de materias con calificaciones bajas. Se pueden visualizar a nivel de alumno o grupo, con la posibilidad de ver el resultado en cada entrega de boletín y en el caso de las inasistencias, además se puede visualizar por semana.
- INOA:** es utilizado principalmente por los tutores de los alumnos en Ánima. Este sistema contiene los seguimientos psicológicos de los alumnos por parte de los tutores. Con estos datos se pueden obtener el número de intervenciones realizadas a cada alumno.
- Evaluador:** contiene el promedio de la evaluación por parte de los empleadores en las pasantías de los alumnos en las empresas colaboradoras. Con estos datos se obtiene el promedio de evaluación de competencias a nivel de alumno y grupo, como también por entrega de boletín.
- Formularios de Google:** entre los formularios se encuentran el Índice de Nivel Socio Económico (INSE), autoevaluaciones de competencias transversales e intervenciones de egresados.

En el formulario INSE se ingresan los datos de los alumnos al ingresar en *Ánima*. Este permite calcular el índice que caracteriza el poder adquisitivo o los ingresos de los hogares a partir de un conjunto de preguntas indirectas. A partir de esto se establecen siete niveles socio económicos: B-, B+, M-, M, M+, A-, A+. Contiene información sobre el barrio donde viven los alumnos y la descripción de su vivienda. Esto brinda la posibilidad de obtener reportes de la cantidad de alumnos según su ubicación geográfica y situación económica.

Las autoevaluaciones de competencias transversales contienen información de escala numérica en las distintas competencias que se evalúan en *Ánima*. Algunas competencias evaluadas son respeto, comunicación y participación, entre otras.

Las intervenciones de egresados contienen información de seguimiento a los mismos. Las intervenciones pueden ser entrevistas, recomendaciones, pruebas de orientación vocacional, por mencionar algunas. Estas se muestran en reportes visualizando la cantidad de cada una de ellas.

- **Archivos excel de la institución:** los archivos excel de uso interno de la institución contienen información de los docentes (que consiste en las asignaturas que dictan, las notas de los boletines y exámenes con que califican a sus alumnos, las asistencias a clases y coordinaciones) y un excel con las notas de los exámenes de los alumnos en cada período que se realizaron. Con estos datos se pueden obtener los promedios de las notas de los exámenes en cada año para los alumnos y cursos. En el caso de los datos de docentes, estos se visualizan en reportes obteniendo el total de inasistencias, como también las calificaciones que brindó en los boletines y exámenes por cada entrega.

4.1.3. Relevamiento

En esta sección se encuentran los resultados obtenidos para el primer objetivo del proyecto. A partir de las reuniones de análisis y de los datos existentes se obtuvieron los requerimientos para la implementación de la solución a diseñar. Dichos requerimientos se dividieron en dos categorías; uno se enfoca en la obtención de datos de Moodle como segundo objetivo del proyecto y el otro en la integración de los datos de los diversos sistemas como el tercer objetivo del proyecto. En las siguientes subsecciones se detallan los requerimientos anteriormente mencionados.

Obtención de datos de Moodle

Este tipo de requerimientos están relacionados a la plataforma de Moodle, de la cual no se obtenía información previamente a comenzar con este proyecto. Por esta razón es que *Ánima* procuraba obtener la siguiente información de forma rápida y sencilla:

- **Cantidad de elementos creados:** permite informar a la institución sobre cuántos elementos existen en un curso. Se denomina elementos a material

subido por el docente del curso, trabajos a entregar por parte de los alumnos, URL a sitios educativos como material extra, encuestas que se le realizan a los alumnos, entre otros. A su vez se deseaba que esta información se pudiera filtrar por curso, por tipo de elemento y por período de tiempo.

- **Entrega de tareas:** necesario para conocer qué alumnos realizan las entregas de los trabajos que plantean los docentes. Es imperativo que se pueda filtrar la información obtenida por alumno y por materia.
- **Retroalimentación de las entregas:** en este punto se deseaba saber la retroalimentación por parte de los docentes hacia los alumnos sobre las entregas de trabajos obtenidos en el requisito anteriormente descrito.
- **Entregas en fecha:** procuran conocer si los alumnos realizaron las entregas de las tareas dentro de las fechas estipuladas por los docentes.
- **Uso del foro:** se refiere a conocer la cantidad de los mensajes existentes en los foros de los cursos, requerida para saber si los foros de los cursos son utilizados y en qué medida, permitiendo saber si hay participación constante en los mismos.
- **Aprobación de tareas:** para este requisito es necesario a partir de las notas de las entregas obtenidas anteriormente saber cuáles son nota de aprobación y cuáles de reprobación. Permite a su vez realizar filtros por alumno.
- **Calificaciones:** se quiere obtener las calificaciones de los alumnos registradas en Moodle, como por ejemplo las evaluaciones de tareas.
- **Accesos de los usuarios a Moodle:** esta información es utilizada para tener conocimiento de cuán a menudo entran los usuarios a la plataforma, ya sean alumnos o docentes. A partir de esta información se puede prever, por ejemplo, si un alumno se encuentra siguiendo los cursos de forma continua.

En base a estos requerimientos se realizó un análisis de la API que brinda el sitio de Moodle para poder estimar el grado de dificultad de cada requerimiento y agruparlos en requerimientos de dificultad Baja, Media y Alta. Para realizar esta categorización se tomó en cuenta cuántos servicios de la API de Moodle se debían consultar por cada requerimiento. Finalizada dicha categorización se realizó una nueva reunión con los encargados del proyecto en *Ánima* y se los consultó para que según su experiencia le asignaran una prioridad Alta, Media o Baja a cada requerimiento. Se les solicitó que los calificaran dependiendo de qué tan necesario fuera la obtención de la información en cada requerimiento para la institución. Habiendo calificado cada requisito por su dificultad y su prioridad se los pasó a ordenar con el fin de tener una línea clara a seguir en el desarrollo. En el caso de que no se llegaran a poder realizar todos los requerimientos en el tiempo designado al proyecto, se podrían concretar los más importantes para *Ánima* en primer lugar. El orden resultante determinó que los requerimientos más prioritarios para *Ánima* eran los calificados por el grupo de trabajo con una dificultad Baja y Media, como se ve en el Cuadro 4.1, por lo que fueron los requerimientos por donde se comenzó a trabajar.

Requerimientos	Dificultad	Prioridad (ANIMA)
Cantidad de elementos creados	Baja	Alta
Entrega de tareas	Media	Alta
Retroalimentación de las entregas	Media	Media / Alta
Entregas en fecha	Media	Media
Uso del foro	Alta	Media
Aprobación de tareas	Media	Baja
Calificaciones	Media	Baja
Accesos de los usuarios a Moodle	Alta	Media / Alta

Cuadro 4.1: Requerimientos priorizados

Como se puede observar en el Cuadro 4.1 el último requerimiento, si bien tenía una prioridad “Media / Alta” para *Ánima*, se decidió dejarlo como último requerimiento a realizar. Esto se debe a que al momento de analizar cada uno de los requerimientos se detectó que era necesario utilizar funcionalidades de una versión de la API de Moodle más nueva que con la que contaba *Ánima*. Dicho requerimiento no fue descartado porque en un principio se discutió por parte de *Ánima* la intención de actualizar la versión de Moodle. Por lo tanto, se lo dejó como último requerimiento a desarrollar en caso de que esto ocurriese.

Integración

Por otro lado, este requerimiento hace referencia a la integración de los datos provenientes de los distintos sistemas informáticos utilizados por *Ánima*, incluido Moodle. Su objetivo final es tener toda la información en un mismo lugar para poder consultarla de una manera sencilla. Un ejemplo de esto es la creación de sus cuadros de mando, donde hacen uso de una gran cantidad de datos provenientes de distintos sistemas.

Una vez propuesta la solución de diseñar un Data Warehouse (la cual se explicará más en detalle en el Capítulo 6), por parte de *Ánima* se solicitó un Script correspondiente al diseño obtenido, por lo que se añadió como un requerimiento.

Resumen del relevamiento de requerimientos

Se pueden visualizar los requerimientos del proyecto en forma resumida en el siguiente Cuadro 4.2

Integración	Solución a la integración de los datos de los sistemas
	Script de la base de datos del Data Warehouse
Obtención de datos de Moodle	Cantidad de elementos creados
	Entrega de tareas
	Retroalimentación de las entregas
	Entregas en fecha
	Uso del foro
	Aprobación de tareas
	Calificaciones
	Accesos de los usuarios a Moodle

Cuadro 4.2: Requerimientos

4.2. Arquitectura

En esta sección se explicarán las arquitecturas propuestas como solución para el segundo 4.2.1 y tercer 4.2.2 objetivo del proyecto. Se comenzó planteando una arquitectura general que pudiera resolver toda la problemática de *Ánima*. Luego, como esta solución era muy extensa de desarrollar durante el tiempo que dura el proyecto, se buscó resolver los problemas más prioritarios para *Ánima*. En base a esto se desarrolló el Módulo de extracción de Moodle y un diseño para el Data Warehouse propuesto.

4.2.1. Módulo de extracción de Moodle

El módulo de extracción de Moodle, llamado “Data *Ánima*”, busca resolver la problemática que *Ánima* tenía en cuanto a Moodle y se desarrolla en solución a los requerimientos planteados en la Sección 4.1.3 en la Subsección de **Obtención de datos de Moodle**. Esta solución busca extraer la información necesaria desde Moodle utilizando la API que la plataforma ofrece y volcarla en un archivo excel, como se puede observar en el Figura 4.4. Luego para crear un cuadro de mando que contenga la información obtenida desde Moodle simplemente se consume el archivo generado por Data *Ánima* desde Google Data Studio.

Esta solución se basó en la dinámica de trabajo que la institución tenía hasta el momento para generar sus cuadros de mando intentando que no resulte algo ajeno a lo que ya utilizaban anteriormente.

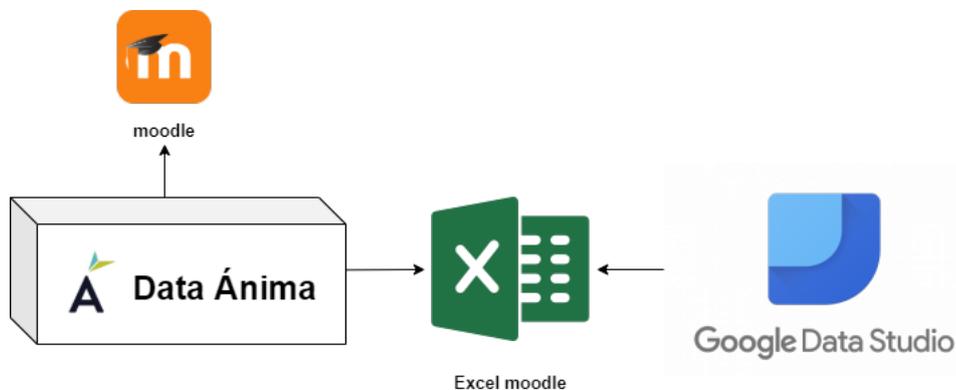


Figura 4.4: Solución diseñada para obtener datos de Moodle

El módulo denominado Data *Ánima* que se observa en la Figura 4.4 se detalla más en profundidad en el Capítulo 5.

4.2.2. Data Warehouse

En base al requerimiento planteado en la Sección 4.1.3 en la Subsección de **Integración** es que se planteó la utilización de un Data Warehouse en busca de una solución. La estructura del Data Warehouse diseñado, en gran parte se derivó de los archivos excel provenientes de los sistemas informáticos y por otra parte de los requerimientos que se querían resolver en cuanto a Moodle. Además, de estos archivos se obtuvieron detalles estructurales de los datos, el origen de cada

uno de éstos, cómo están relacionados y cuál es la información más relevante para la institución. Todos estos datos fueron esenciales para el diseño y la creación de la solución final.

4.2.3. Solución general

Esta solución fue diseñada buscando resolver la problemática de *Ánima* para no realizar el trabajo manual de unificar toda la información proveniente de los distintos sistemas informáticos utilizados por la institución. Dicha solución incorpora a su vez de manera muy sencilla el módulo de extracción de Moodle (*Data Ánima*), por lo cual es fácilmente reutilizable sin tener que realizar excesivas modificaciones para integrarlo.

Ya que toda la información proveniente de los distintos sistemas (*SIGED*, *INOA*, *Evaluador*, *Formularios de Google*, excel internos de la institución) llega en formato excel, y la nueva información brindada por *Data Ánima* también, se planteó realizar un proceso ETL. Este puede procesar toda la información de los archivos excel y cargarla en un *Data Warehouse* como se puede observar en la [Figura 4.5](#). Teniendo la información almacenada en un *Data Warehouse* es posible utilizar un visualizador de datos como *Google Data Studio* que ya es conocido por *Ánima*, para realizar sus cuadros de mando. Esto se debe a que se cuenta con toda la información proveniente de todos los sistemas informáticos que utiliza la institución en un mismo lugar.

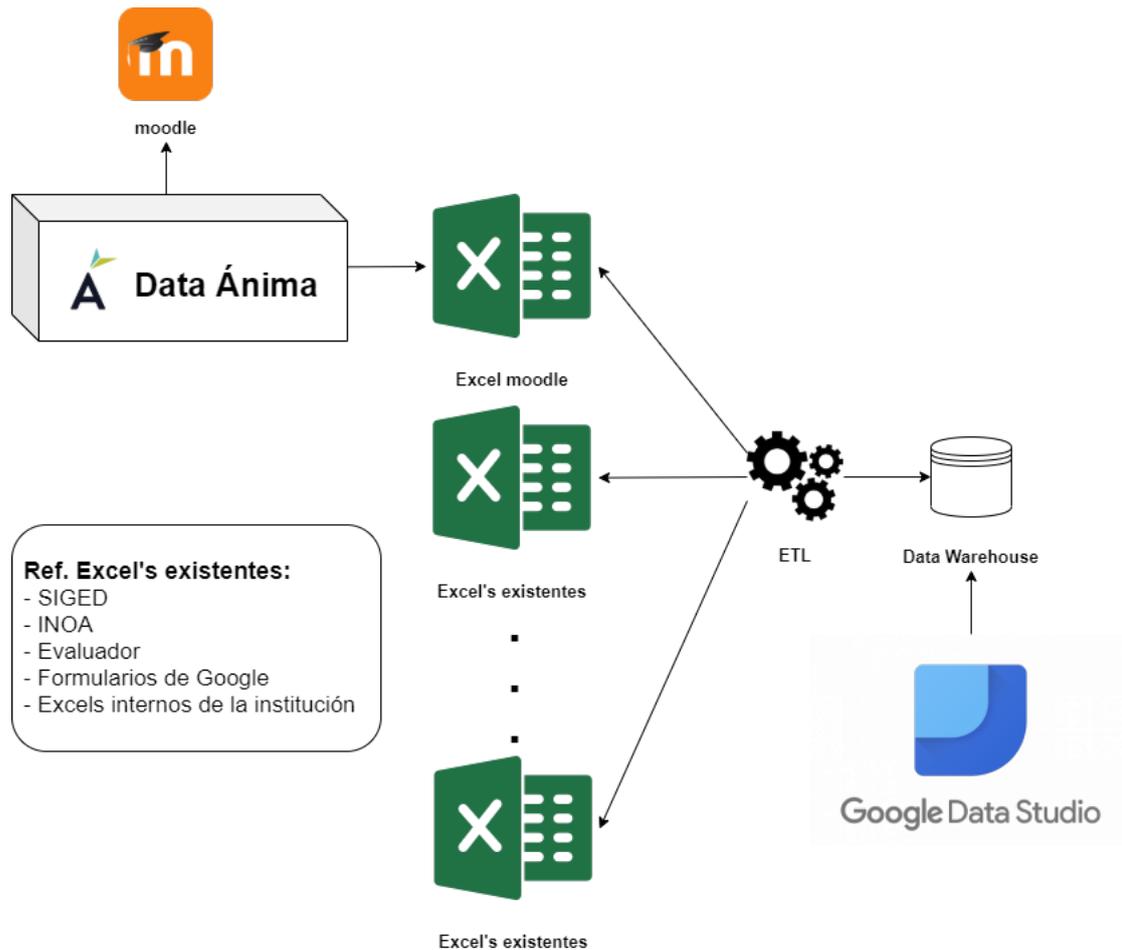


Figura 4.5: Solución general diseñada para Anima

La utilización de un Data Warehouse facilita el mantenimiento de toda la información histórica de la institución. De todos modos, si en un futuro se quisiera dejar de utilizar Google Data Studio para visualizar los datos, existen varias herramientas 2.5 para obtener reportes a partir de un Data Warehouse. Un análisis detallado del proceso de diseño del Data Warehouse puede observarse en el Capítulo 6.

4.2.4. Alternativas analizadas

La solución general planteada no fue la única analizada a la hora de tomar una decisión sobre qué arquitectura diseñar. Entre las alternativas que se tuvieron en cuenta estaba la posibilidad sustituir el Data Warehouse por un proceso ETL que genere un archivo excel con la información unificada y transformada desde todos los excel de origen que se pueden observar en la Figura 4.5. Otra alternativa analizada también fue utilizar una base de datos relacional en lugar del Data Warehouse y que el proceso ETL volcara la información procesada en ella. Dichas alternativas fueron descartadas por distintas razones que se explican a continuación.

Utilizar un único excel generado por el proceso ETL se asemejaba mucho a la tarea manual que realizaban en Anima, por lo que podía llegar a ser una buena

solución. Sin embargo, se encontró como limitante que esta solución no era eficiente a la hora de querer consultar el registro histórico de toda la información en *Ánima*. Para ese caso se debía guardar cada excel generado por el proceso ETL con distinta información. Aquí se generaba otro problema, ya que es muy complejo cuando desde Google Data Studio se quieren mostrar los datos desde distintos archivos excel. Estas razones dieron lugar a que se descartara esta opción.

Luego se pensó en utilizar una base de datos relacional para modelar toda la realidad de la institución y así poder conectar Google Data Studio a la base de datos y poder obtener los cuadros de mando con toda la información necesaria. Esta opción solucionaba el problema que se tenía con el archivo excel de la opción anterior. Pero se llegó a la conclusión de que una vez que la base de datos estuviera muy cargada de información con el correr del tiempo, el desempeño se iba a ir degradando, porque algunos datos a consultar por los cuadros de mando demandarían varios JOINS entre las tablas de la base.

El problema encontrado a la hora de querer utilizar una base de datos se lo solucionó utilizando un Data Warehouse diseñado específicamente para los indicadores que se quieren consultar en los cuadros de mando que *Ánima* genera. Esta solución resulta ser más eficiente que una base de datos relacional ya que está diseñado específicamente para lo que se quiere consultar y se evita consultar demasiadas tablas con operaciones de JOINS. Por todo esto se llegó a la solución general de la problemática enfrentada planteada en la Figura 4.5.

Capítulo 5

Implementación del prototipo de extracción de Moodle

En este capítulo se presenta el prototipo creado como posible solución de los requerimientos planteados en el Capítulo 4 que cumple el segundo objetivo del proyecto. Se detalla la arquitectura del prototipo implementado y las diferentes tecnologías utilizadas para su desarrollo. A su vez, se presentan nuevos cuadros de mando realizados con los datos extraídos a través del prototipo desarrollado. Por último, se presenta la verificación efectuada a la información obtenida de Moodle.

5.1. Arquitectura

En esta sección se presenta la arquitectura del prototipo desarrollado para la extracción, transformación y carga de la información obtenida de la aplicación Moodle utilizada por la institución. Este prototipo fue diseñado de forma modular, para que pueda ser escalable agregando más módulos que trabajen con otros sistemas. A continuación, se puede observar la arquitectura diseñada en la Figura 5.1.

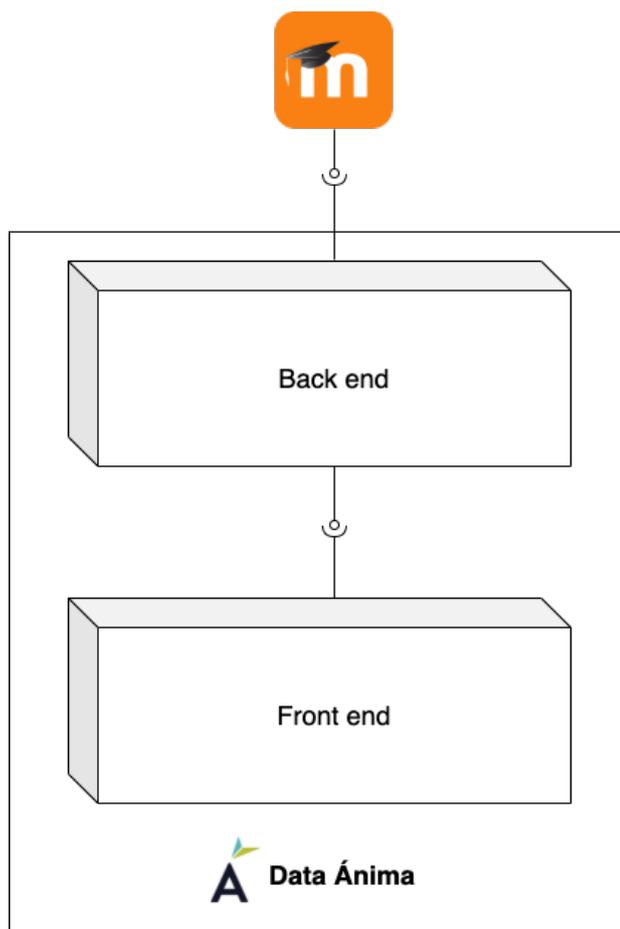


Figura 5.1: Arquitectura Data Anima

Esta arquitectura se define mediante dos módulos separados, donde uno se encarga de la interfaz de usuario (módulo Front end) y otro se encarga de la extracción, transformación y envío de los datos a la interfaz de usuario (módulo Back end).

El Front end fue desarrollado utilizando la tecnología React. Dada la baja complejidad de este módulo, se utilizó la arquitectura Single Page Application, esto se refiere a que la aplicación se basa en un sitio web que contiene una única página en la cual se encuentran todas las funcionalidades disponibles.

Por otro lado, el módulo Back end está implementado con las tecnologías TypeScript y NodeJs. En este se extrae la información de Moodle la cual es transformada y manipulada para que pueda ser fácilmente interpretada. A partir de esto se puede obtener un mejor rendimiento al momento de manipular la información como también almacenar únicamente la información que va a ser necesaria. Con el resultado de este proceso se genera un archivo excel que cumple con los requerimientos definidos por la institución.

Para dicho módulo se puede identificar una arquitectura en capas, las cuales son: Interfaces, Lógica de Negocio y Servicios Web. Cada una de estas capas cumple un rol específico en el módulo y no todas se comunican entre sí. A continuación, en la Figura 5.2 se puede observar el diseño de dichas capas y sus interacciones.

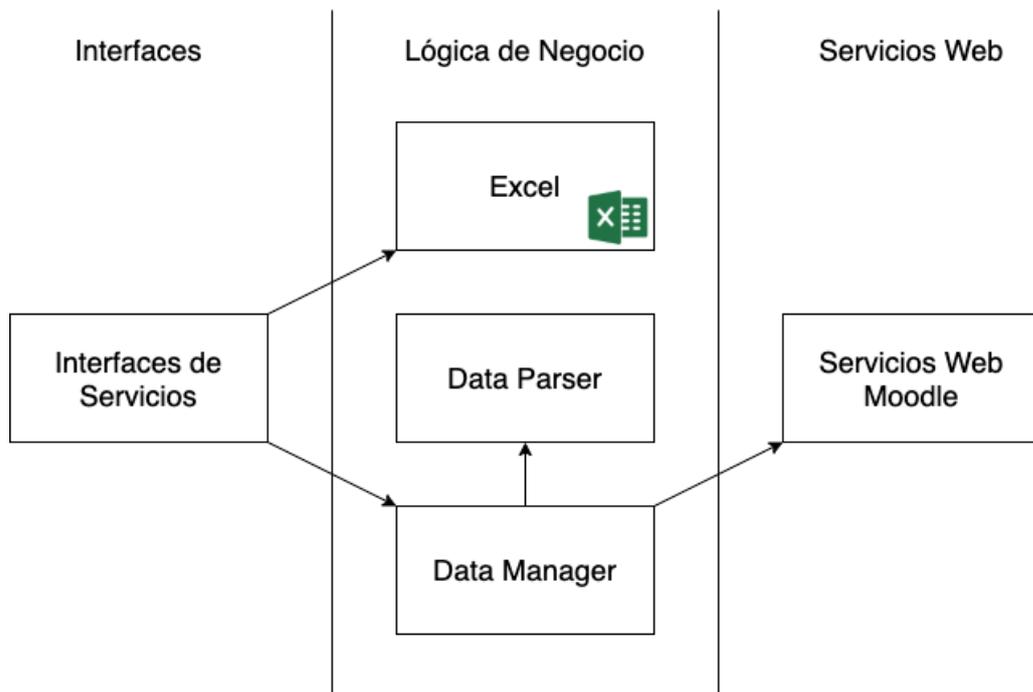


Figura 5.2: Arquitectura Backend

Dentro de la capa de interfaces se encuentra el módulo denominado Interfaces de Servicios, este se encarga de definir las distintas URL a partir de las cuales se ejecutan cada una de las funcionalidades del Back end desde el exterior de este. La misma se encuentra pública para ser utilizada desde el exterior del servidor utilizando una API REST y se comunica con la capa denominada Lógica de Negocio.

Dentro de la capa de Lógica de Negocio se encuentran los módulos Excel, Data Manager y Data Parser. Estos son los responsables del manejo y transformación de la información obtenida a partir de la capa de Servicios Web. El módulo Excel se encarga de generar un archivo excel a partir de la información que se le es proporcionada. En este se puede encontrar todas las librerías necesarias para el manejo de archivos y la lógica requerida para organizar de forma adecuada la información proporcionada en los archivos generados.

Por otro lado, el módulo Data Manager se encarga de administrar todas las estructuras necesarias para la creación del archivo, las cuales son creadas a partir de la información obtenida de Moodle. En este módulo también se mantiene un control de cada una de las ejecuciones realizadas de forma que no se puedan realizar ejecuciones simultáneas, ya que la información generada por una de las ejecuciones al finalizar queda disponible para todos los usuarios que tengan los permisos necesarios para acceder al prototipo de extracción implementado. Este módulo es el que se comunica con la capa de Servicios Web y con el módulo Data Parser para la obtención y transformación de la información respectivamente.

Por último, se encuentra el módulo Data Parser, a este se le proporciona la información que le fue entregada al Data Manager a través de las consultas realizadas por la capa de Servicios Web. Esta información es transformada a estructuras de datos y a su vez es filtrada ya que las respuestas a las consultas realizadas a

Moodle son obtenidas en formato JSON y contienen información que no es utilizada. Y por último estas estructuras de datos son utilizadas por el Data Manager para realizar las tareas de dicho módulo.

La capa Servicios Web se encarga de realizar todas las consultas a los distintos servicios. Dentro de esta se encuentra el módulo Servicios Web Moodle, este es ejecutado por el Data Manager y se encarga de realizar todas las consultas necesarias a la API REST de Moodle para la obtención de la información requerida para la creación del archivo. Está implementado de manera de reutilizar la mayor parte del código de forma que el mantenimiento del mismo sea lo más simple posible y disminuya su complejidad. Luego de obtenida la información, ésta es entregada al Data Manager sin efectuar ningún cambio previamente. Esta capa utiliza un archivo de configuración del cual obtiene la dirección URL de la API a la que se quiere realizar las consultas, brindando la posibilidad de cambiar de forma sencilla en un único lugar del prototipo la dirección de la cual se quiere obtener la información, y de esta forma se pueda generar los mismos archivos, pero para otra instancia del sistema Moodle.

Para poder trabajar con la API de Moodle de *Ánima* fue necesario realizar diferentes configuraciones, las cuales son descritas en el Apéndice B.2. Por un lado, se realizaron configuraciones que habilitan la funcionalidad y los permisos necesarios para su utilización. Por otro lado, se debió definir un servicio personalizado en el cual se agregaron todas las funciones requeridas de la API de Moodle para la implementación de *Data Ánima*, ya que algunas de estas no están incluidas en el servicio que provee Moodle por defecto. De esta forma al incluir todas las funciones utilizadas en un único servicio se evita el manejo de distintos servicios para las distintas funciones.

Tanto la capa Lógica de Negocio al igual que la capa Servicios Web utilizan un archivo de configuración en el cual está especificado el formato de algunos valores como ser fechas. Esto genera el beneficio de que si este formato se quiere modificar solo es necesario cambiarlo en un lugar en la aplicación.

Como fue detallado anteriormente en el Capítulo 3.5.1, se ideó una solución en la cual no se dependa de la instancia de Moodle utilizada, para esto se evitó utilizar lógicas específicas de negocio utilizadas actualmente por *Ánima*, como ser configuraciones de roles de usuario en la institución, categoría de los cursos, entre otros. Esto favorece en varios puntos a la institución ya que disminuye la probabilidad de futuros errores en el prototipo y disminuye el mantenimiento del mismo dado que como se mencionó anteriormente en el Capítulo 3.4.1 la institución no cuenta con un equipo de desarrollo para realizar futuros mantenimientos y mejoras al mismo. Por otro lado, esta implementación permite trasladar *Data Ánima* a otra instancia de Moodle de ser necesario generando el beneficio de que el proyecto pueda ser de valor tanto para esta organización como para otra.

5.2. Tecnologías

Para la creación de los módulos fue necesario utilizar distintas tecnologías, el módulo Back end se implementó con Typescript [42], este es un lenguaje de programación de código abierto creado por Microsoft. Este lenguaje trabaja sobre Javascript [21] y le agrega definiciones de tipos estáticos entre otras características que facilitan el uso y aprendizaje del mismo. Luego este código se interpreta

en el entorno de ejecución de NodeJs, el cual es un entorno de ejecución de Javascript orientado a eventos asíncronos, NodeJs está diseñado para la creación de aplicaciones escalables [28].

Dentro de este entorno se utilizaron diferentes librerías, módulos y frameworks:

- **File System:** permite acceder al directorio de archivos del sistema [8], este fue utilizado para obtener acceso al archivo excel generado.
- **Express:** es un framework que se utiliza en NodeJs que se encarga de construir un servidor el cual provee servicios HTTP. Para el prototipo se utilizó en la creación de una API REST, la cual responde a los pedidos realizados por el Front end. [7]
- **Path:** genera una ruta a archivos dentro del directorio de archivos. Este es utilizado en conjunto con el File System para obtener el archivo excel generado. [30]
- **Query-string:** provee herramientas para la creación de URL y darle formato a las mismas. Este fue utilizado para la creación de las consultas hacia la plataforma de Moodle. [34]
- **Node-fetch:** se encarga de ejecutar consultas HTTP. Este fue utilizado para realizar las consultas hacia la plataforma de Moodle. [27]
- **Dateformat:** encargado de la creación de un objeto de tipo Date y provee las herramientas para darle el formato de fecha deseado. Este fue utilizado para la transformación de las fechas obtenidas de Moodle. [5]
- **Excel4node:** brinda herramientas para la creación y edición de archivos excel. Esta fue utilizada para la creación del archivo excel resultante del proceso de obtención de la información de Moodle. [6]

Para la creación del módulo Front end se utilizó el lenguaje de programación Javascript utilizado para el desarrollo web [21]. Dentro de este módulo se utilizaron diferentes librerías:

- **React:** es una librería utilizada para crear interfaces de usuario, está basada en componentes y en el manejo de estados. Esta fue utilizada para crear la interfaz de usuario de la aplicación. [36]
- **Axios:** es utilizada para realizar consultas HTTP. Esta fue utilizada para realizar las consultas HTTP hacia el módulo Back end. [2]
- **Material-ui:** ofrece un conjunto de componentes React para la creación de páginas web. Esta fue utilizada para la creación de la interfaz de usuario de la página web. [23]

5.3. Funcionalidades

Como se explicó anteriormente este prototipo de extracción de Moodle cuenta con un sitio web en el cual se pueden encontrar todas las funcionalidades

del mismo. Estas funcionalidades fueron implementadas a partir de los requerimientos obtenidos y algunas consideraciones tomadas por el equipo para brindar ciertas características de seguridad a la aplicación. Las funcionalidades son las siguientes:

- **Login:** para acceder al resto de las funcionalidades del prototipo es requerido autenticarse en este. El mismo está integrado a los servicios web de Moodle, por lo tanto, es posible autenticarse utilizando las cuentas de los usuarios de Moodle de *Ánima*. Para mayor seguridad también se chequea que el usuario autenticado tenga los permisos suficientes para realizar las consultas. Es decir, que solamente los usuarios con suficientes permisos de la institución puedan obtener esta información, ya que la misma es confidencial. Para lograr una mejor experiencia de usuario se crea una sesión del mismo dentro del navegador y se mantiene guardada para que este no tenga que ingresar sus credenciales nuevamente al volver a ingresar al sitio.
- **Procesar información:** esta funcionalidad se puede ejecutar pulsando el botón “PROCESAR INFORMACIÓN” que se ve en la Figura 5.3, la cual genera que comience el proceso de la obtención y procesamiento de la información de Moodle que es almacenada en un archivo excel en el servidor. En caso de que ya haya un proceso en curso se mostrará un mensaje indicando que el proceso ya está en curso.
- **Descargar información:** luego de finalizado el procesamiento y pulsando el botón “DESCARGAR INFORMACIÓN” que se observa en la Figura 5.3, se procede a descargar del servidor el archivo excel generado por el último procesamiento. En caso de que haya un proceso en curso se mostrará un mensaje indicando que todavía no se puede descargar el archivo.

En la siguiente Figura 5.3 se muestra el sitio web del prototipo de extracción implementado.

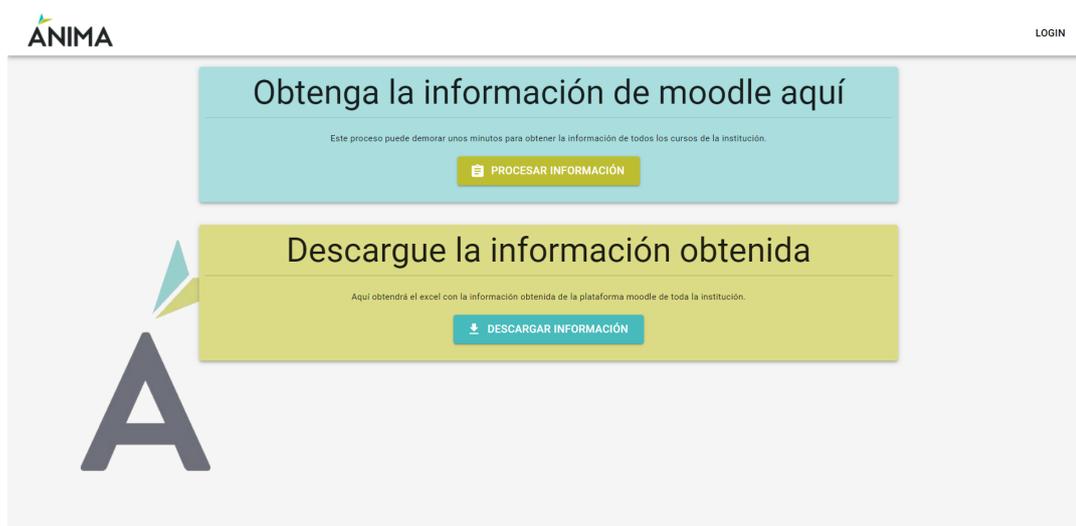


Figura 5.3: Sitio web Data *Ánima*.

A continuación, en la Figura 5.4 se puede observar un diagrama que contiene el flujo de las distintas funcionalidades del prototipo de extracción implementado.

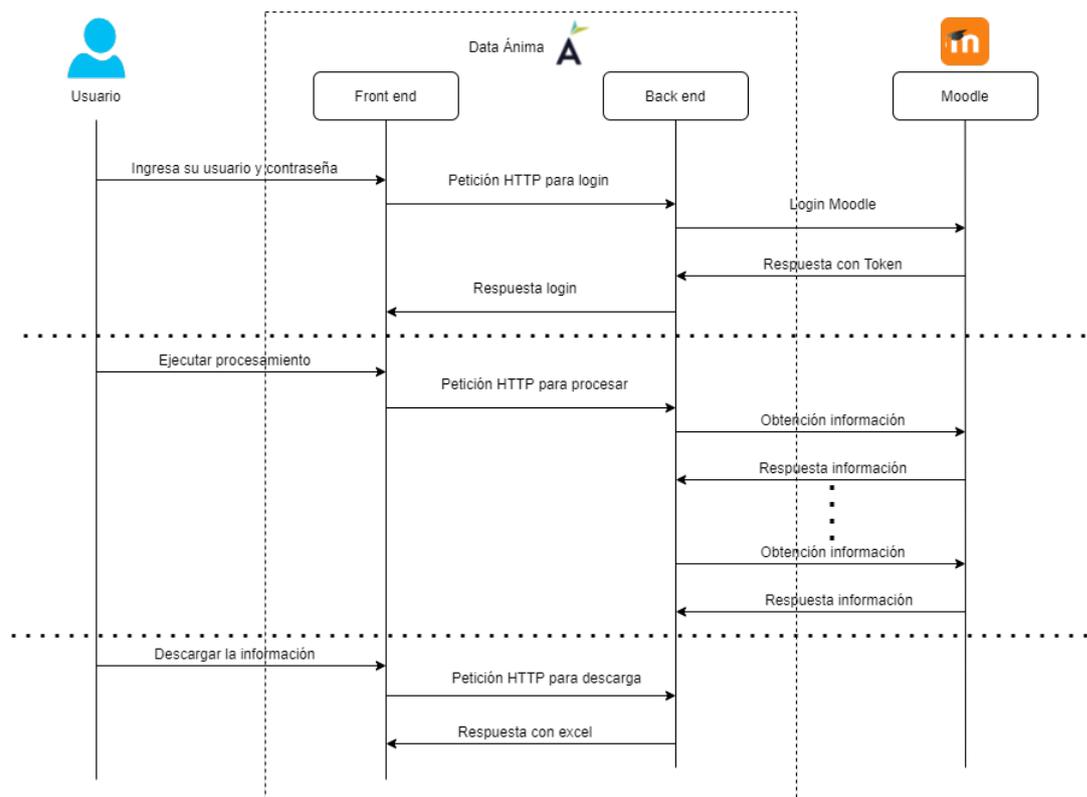


Figura 5.4: Flujo de ejecución de Data Ánima.

Para el manejo del prototipo implementado se crearon manuales de usuario que se pueden encontrar en el Apéndice A.

5.4. Reportes obtenidos

Una vez obtenido el excel con los datos de Moodle se procedió a crear algunos reportes en Google Data Studio de acuerdo con los requerimientos, con el fin de que la institución tenga de ejemplo la manera de cómo tratar los datos. A continuación, se describen los reportes realizados.

Por un lado, se obtuvieron reportes para obtener la cantidad de elementos existentes en Moodle. Estas cantidades se pueden ver por cursos a los que pertenecen los elementos y por su tipo. A su vez, de acuerdo con Ánima, estas cantidades también son filtradas por visibilidad en Moodle, dado que muchos de los elementos están ocultos y se hacen visibles a medida que avanza la actividad académica durante el año. En la siguiente Figura 5.5 se muestra el reporte obtenido en Google Data Studio.

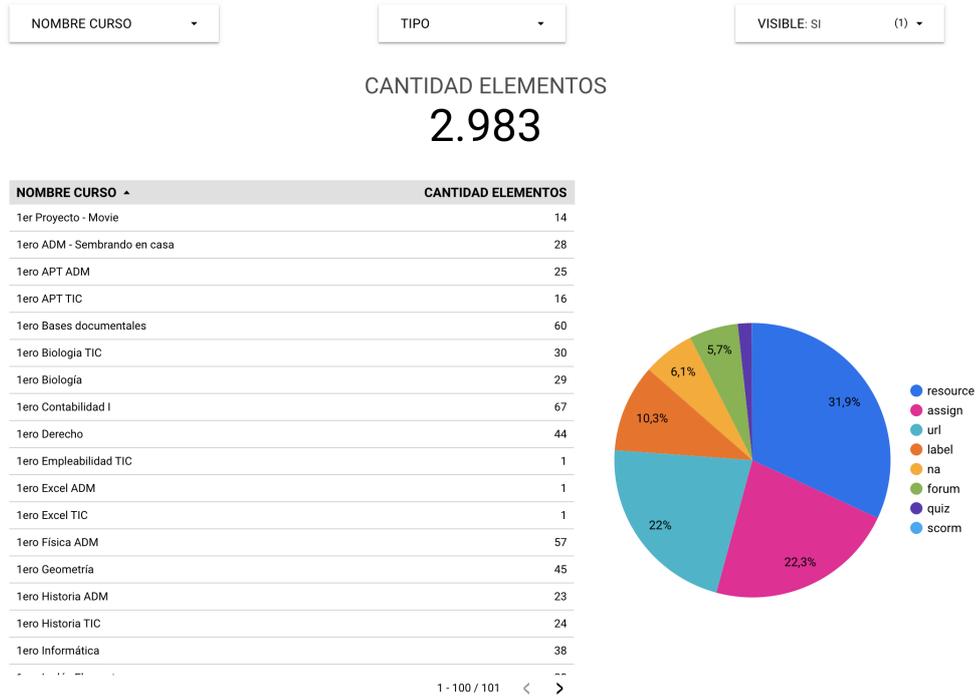


Figura 5.5: Cuadro de mando - Elementos de Moodle

También se realizaron reportes con la cantidad de entregas de tareas realizadas en Moodle. En este caso se puede filtrar por el curso donde pertenecen las tareas correspondientes a las entregas, por el nombre del alumno que realiza la entrega de la tarea, el estado de la entrega, y la fecha de entrega. A su vez se realizó un reporte que indica la cantidad de entregas que los docentes tuvieron en los cursos que dictan, pudiéndose filtrar por docente y por sus cursos dictados, además del estado de las entregas. En la Figura 5.6 se muestra el reporte creado en Google Data Studio.

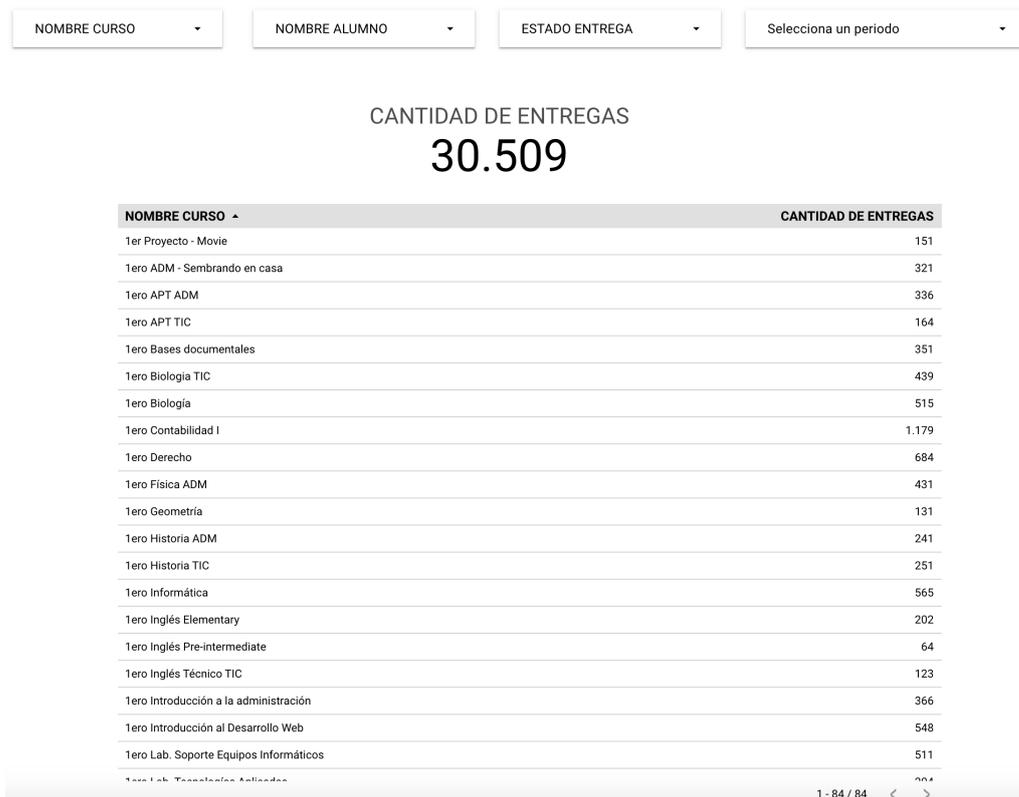


Figura 5.6: Cuadro de mando - Entregas de tareas

Por último, se obtuvieron reportes de la cantidad de mensajes en los foros en Moodle. Los mismos pueden filtrarse por fecha de envío, el nombre y rol del usuario que envía el mensaje, el nombre del curso donde se encuentra el foro correspondiente y el nombre del foro. También se creó un reporte a nivel de docente para obtener la cantidad de mensajes en los foros que se encuentran en los cursos que dictan. Estos reportes se pueden filtrar por el nombre del docente de interés, el nombre de los cursos que dicta, y el rol del usuario que envía los mensajes. Un ejemplo de este reporte se puede observar en la Figura 5.7.

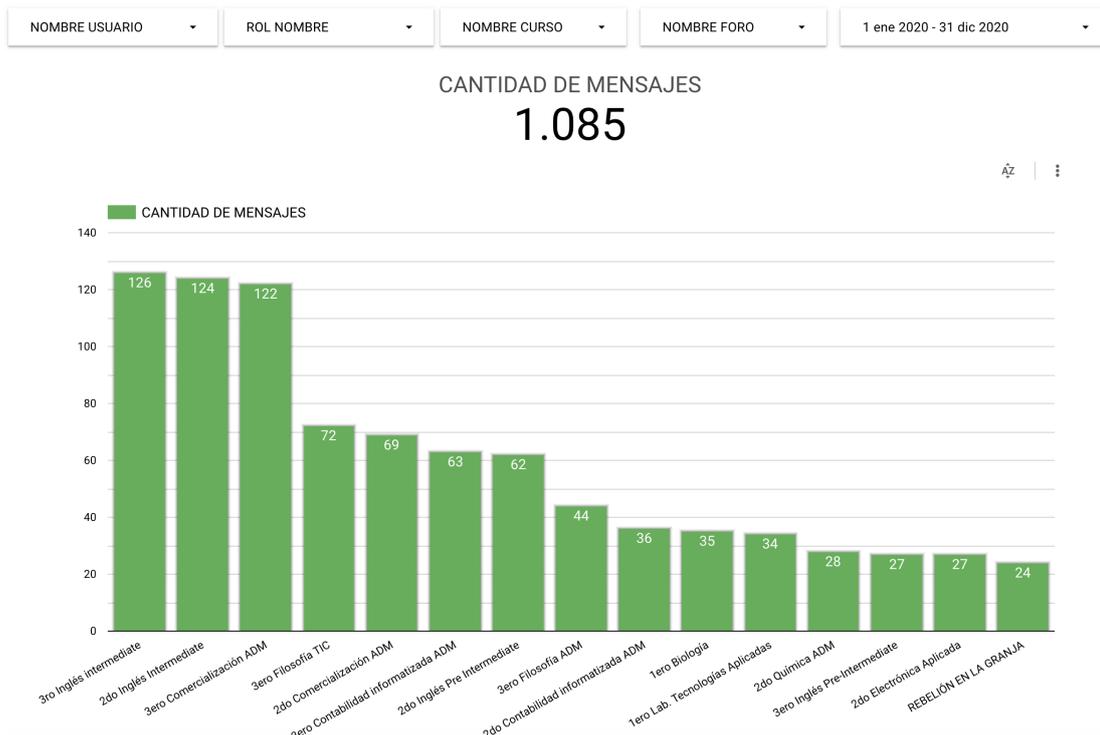


Figura 5.7: Cuadro de mando - Mensajes en foros

5.5. Verificación de los datos obtenidos

Para la verificación de que los datos extraídos por Data Ánima coinciden con la información generada por la plataforma Moodle se definieron varios flujos de prueba. Estos flujos se aplicaron a una cantidad acotada de datos, ya que corroborar toda la información obtenida de forma manual llevaría demasiado tiempo y no se obtendría un grado mucho mayor de verificación al efectuado. Es decir que se efectuaron flujos de prueba para cada uno de los requerimientos y para cada uno de éstos se analizó de forma arbitraria un conjunto de datos. Es relevante indicar que todas estas pruebas están ubicadas temporalmente el día 27 de febrero del año 2021, a excepción de las pruebas con respecto a las calificaciones de las entregas de tareas que se realizaron el día 31 de mayo del año 2021.

A continuación, se presentan los flujos planteados y los resultados obtenidos para cada uno de estos. Dentro de los resultados no se detalla cierta información ya que la misma es sensible para la organización, por lo tanto, se utilizarán solamente los identificadores (id) de cada elemento a analizar en la plataforma Moodle.

5.5.1. Cursos

Los cursos representan en el sistema Moodle a los cursos curriculares de la institución. En este caso de uso, se busca por un lado validar que la cantidad de cursos obtenidos sea igual a la cantidad de cursos en Moodle. Por otro lado, se seleccionó un conjunto de cursos y se verificó que los mismos constaban con las características encontradas en Moodle. La cantidad de cursos en su momento fue

verificada y tanto en el prototipo como en Moodle se tenían 105 cursos, luego se verificaron los cursos seleccionados y sus características coincidían en ambos.

5.5.2. Entregas

Las entregas son aquellas que se realizan en respuesta de un elemento de Moodle de tipo “assign”, el cual se refiere a una tarea que requiere de una entrega de un trabajo por parte de los alumnos. Para verificar la extracción de entregas de tareas se tomaron dos cursos como prueba para ver si la cantidad de entregas extraídas por el prototipo implementado coincide con la cantidad de entregas que existen en el sitio de Moodle. Los cursos seleccionados fueron, el curso de id 55 y el curso de id 13.

Para el curso de id 55 se seleccionó la tarea de id 1098, de la cual se extrajeron 13 entregas realizadas en el año 2017, y la información correspondiente a los usuarios que realizaron cada una de las mismas. Esta información se chequeó en Moodle para cada una de las entregas y tanto la cantidad de entregas como la información de las mismas y de los usuarios coincidían. Luego se validó la tarea de id 2024, la cual no tenía ninguna entrega realizada, lo que se verificó en el prototipo y en Moodle.

Las mismas validaciones se realizaron para el curso de id 13, donde la tarea de id 102 obtuvo seis entregas realizadas en el año 2016. Al igual que con el curso anterior se verificó que la información de las tareas y de los usuarios que realizaron estas entregas coincidía en el prototipo y en Moodle. Luego se verificó las tareas de identificadores 70 y 173, las cuales no tenían ninguna entrega realizada en ninguna de las dos fuentes de datos.

Realizando la verificación se encontró que algunas entregas tienen el id del usuario que realizó la entrega, pero no su nombre, eso se investigó para ver cuál era la razón de esto y se llegó a la conclusión de que estos usuarios no están más en Moodle, es decir que fueron eliminados. A continuación, en la Figura 5.8 se puede observar un ejemplo de la respuesta de Moodle cuando se quiere consultar la información del usuario de id 63, el cual tampoco aparece en la lista de usuarios generada por el prototipo.

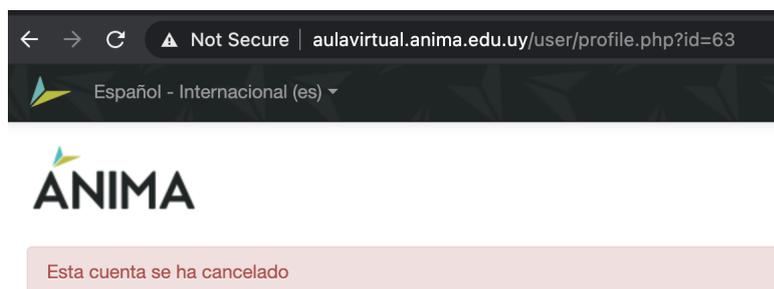


Figura 5.8: Usuario de id 63 eliminado

Calificaciones

Para la verificación de las calificaciones de las entregas de tareas se identificó y utilizó un curso que contuviera entregas calificadas. Dado que el curso de

identificador 3 tenía dos tareas con entregas calificadas dentro de la información obtenida con el prototipo de extracción realizado se procedió a verificar si estaban presentes en la plataforma Moodle y que no faltara ninguna entrega o calificación. La primera tarea cuyo id es 27 contenía 26 entregas calificadas en el año 2017 y se hallaban tanto en el excel extraído como en la plataforma Moodle, para cada una de ellas se pudo comprobar que coincidía el usuario que hizo la entrega y la calificación de la misma. Para la segunda tarea de id 244 se encontraron 59 entregas realizadas en 2016 y 2019 (esto sucede cuando reutilizan las tareas para distintos años) las cuales estaban en el excel y en la plataforma de Moodle, coincidiendo el usuario y la calificación en estos casos también.

5.5.3. Elementos

Los elementos son cada una de las actividades que se pueden agregar a un curso dentro de Moodle. Estos elementos pueden ser de distintos tipos como tareas (antes descritas), URL, encuestas, foros, entre otros. Para realizar esta verificación se seleccionaron dos cursos de forma aleatoria y se contaron cada uno de los elementos que estaban presentes en los cursos con sus características, algunas de ellas son, visible, tipo, fecha de creación, entre otras.

Para el curso de id 61 se identificaron siete elementos, los cuales coincidían en cantidad y características con la información obtenida por el prototipo de extracción. Y esto ocurrió de igual forma con el curso de id 13, el cual tenía 10 elementos.

5.5.4. Usuarios

Los usuarios representan en Moodle a un docente o alumno de la institución. Para realizar esta verificación se controló que la cantidad de usuarios en Moodle sea la misma que la obtenida por el prototipo implementado. Esto se verificó ya que en el sistema de Moodle en la sección de usuarios (Figura 5.9) se puede observar la cantidad total de estos, la cual coincide con la cantidad de usuarios que se obtuvo previamente.



Figura 5.9: Cantidad de usuarios en Moodle

5.5.5. Roles

Los roles dentro de Moodle son utilizados para asignar permisos a los usuarios dentro de la plataforma. Para el caso específico de *Ánima* estos también representan el tipo de usuario para cada uno de los cursos. Es decir, identifica si un usuario para un curso dado es alumno o docente. Para verificar si todos los roles utilizados en Moodle son obtenidos por el prototipo, se ingresó en la sección de roles de Moodle y se verificó que sean los mismos que los extraídos.

5.5.6. Foros

Los foros son un tipo de elemento de Moodle, que tiene la capacidad de crear hilos de conversación entre los usuarios del sistema. En estos se crean publicaciones, las cuales tienen un asunto y un cuerpo, y a partir de ellos los usuarios pueden responder un mensaje. Estos mensajes pueden contener texto, imágenes, entre otras cosas y se pueden crear varios hilos de mensajes anidados. Para la verificación de los foros se seleccionaron dos cursos de identificadores 3 y 28.

Para el curso de id 3 se chequeó la cantidad de mensajes dentro de dos foros de identificadores 172 y 267. Estos tenían siete y ocho mensajes respectivamente, los cuales se verificaron que coincidían en cantidad y características con los extraídos por el prototipo.

Para el curso de id 28 de igual forma que el curso anterior se chequeó la cantidad de mensajes dentro del foro de id 317. Este contenía cuatro mensajes, de los cuales se verificó que coincidían en cantidad y características con los extraídos por *Data Ánima*.

Capítulo 6

Diseño del Data Warehouse

En este capítulo se presenta un posible diseño del Data Warehouse que soporta todos los requerimientos solicitados por *Ánima* para resolver el tercer objetivo de este proyecto. A fin de realizar el diseño propuesto del Data Warehouse se comenzó por agrupar los indicadores de interés para la institución y a partir de estos definir las Medidas a ser utilizadas dentro de los esquemas. A su vez se obtuvieron las dependencias o criterios que determinaban esos indicadores y con estos se definieron las Dimensiones. A través de las Medidas y las Dimensiones se realizaron los esquemas restantes agrupando las mismas. Los esquemas mencionados anteriormente siguen el modelo CMDM. Por otro lado, se realizó el diseño lógico a través de un esquema de bases de datos relacional.

También se presenta un prototipo que aloja los datos extraídos de Moodle en el Data Warehouse.

6.1. Dimensiones y Medidas

A continuación, se presentan las Dimensiones y Medidas del diseño conceptual del Data Warehouse. Por un lado, se definieron las siguientes Medidas:

- **M_CantidadAlumnos:** indica la cantidad de alumnos.
- **M_BoletinAlumno:** hace referencia a los boletines de los alumnos y posee el promedio de notas, promedio de bajas y promedio de exámenes.
- **M_Competiciones:** contiene el promedio de notas dadas a las competencias transversales de los alumnos.
- **M_InasistenciasAlumno:** abarca las inasistencias de los alumnos a clase y las horas totales de las clases a las que están inscritos los mismos.
- **M_InasistenciasClaseDocente:** incluye las inasistencias y la carga horaria total de las clases que dictan los docentes.
- **M_InasistenciasCoordinacionesDocente:** es la suma de las inasistencias y la carga horaria total a las reuniones de coordinación docente.
- **M_AsignaturasDocente:** cantidad de asignaturas por docente.
- **M_ResultadoEntregasTareas:** lleva el registro de las cantidades y los resultados de las entregas realizadas por los alumnos.
- **M_CantidadEntrevistas:** posee la cantidad de entrevistas realizadas por los coordinadores de *Ánima* a los alumnos egresados de la institución.

- **M_CantidadRecomendaciones:** son el total de las recomendaciones hechas por los coordinadores de *Ánima* a los alumnos egresados de la institución.
- **M_CantidadAccesosMoodle:** tiene los accesos a la plataforma Moodle por parte de los usuarios, ya sean alumnos o docentes.
- **M_CantidadAcciones:** contabiliza las acciones que se emplean para los egresados de *Ánima*.
- **M_IntervencionLaboral:** registra todas las intervenciones realizadas por *Ánima* a la actividad laboral de los alumnos egresados.
- **M_CantidadElementosMoodle:** monto total de los elementos creados por los docentes para los cursos en la plataforma Moodle.
- **M_CantidadMensajesForoMoodle:** comprende el número de mensajes en los foros de los cursos de Moodle.

Cada una de las Medidas tiene uno o más indicadores de interés para la institución. Por ejemplo, la Medida “M_ResultadoEntregasTareas” como se ve en la Figura 6.1 contiene los indicadores “CantidadEntregaTareas” con la cantidad de entregas de tareas en la plataforma Moodle y “PromedioCalificacion” que es el promedio de las calificaciones de las entregas de tareas en la plataforma Moodle. En la Figura E.6 en el Apéndice E.1 se puede observar el diagrama del resto de las Medidas.

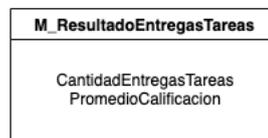


Figura 6.1: Medida “M_ResultadoEntregasTareas”

Por otro lado, se definieron las siguientes Dimensiones:

- **Tiempo:** es el tiempo transcurrido en la institución.
- **Entrega Tareas:** son las entregas de trabajos planteados por los docentes que deben realizar los alumnos a través de la plataforma Moodle.
- **Académico:** contiene los cursos de *Ánima* dentro de la plataforma Moodle.
- **Elementos Moodle:** posee los elementos creados dentro de cada curso en la plataforma Moodle.
- **Usuario:** representa los integrantes de la institución, en los que se pueden encontrar los alumnos cursando, alumnos egresados y los docentes.

Los diagramas correspondientes a las Dimensiones se encuentran en el Apéndice E.1. Para cada una de las Dimensiones se cuenta con un Nivel superior denominado “All” que agrupa todos los datos, y es útil para obtener el total buscado en cada función de agregación.

La Dimensión “Entrega Tareas” se contempla en la Figura 6.2. Se puede observar que en su Nivel inferior se encuentran las entregas de tareas que existen dentro de la plataforma Moodle, que son agrupadas por distintas características al subir de Nivel. Estas características son:

- **Estado:** pudiendo ser entregado o no entregado.
- **Calificada:** con un valor binario indicando si la entrega fue calificada.
- **Resultado:** que tendrá un valor aprobado o reprobado en caso de tener una calificación.
- **Retroalimentación:** valor binario indicando si tiene retroalimentación por parte del docente.
- **En Fecha:** valor binario dependiendo si la entrega se realizó en fecha.

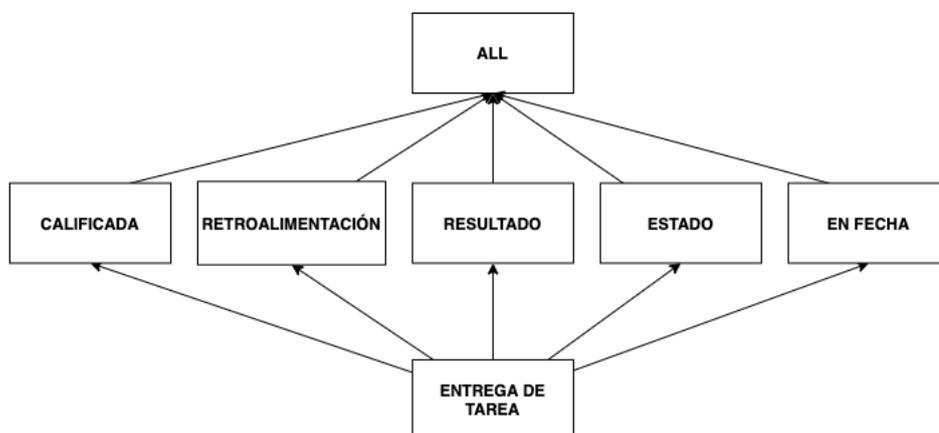


Figura 6.2: Dimensión “Entrega Tareas”.

En la Dimensión “Tiempo” (Figura E.2) se observa que la misma está conformada por la fecha en su Nivel inferior y los Niveles superiores son los agrupamientos de estas fechas por semana, mes, entrega y año. Para el caso del Nivel de “Entrega” se toman los números de las cinco entregas de boletín anuales y agrupa las fechas que se encuentran dentro del período de evaluación para las mismas.

Dentro de la Dimensión “Académico” (Figura E.3) se observa que se parte de las materias en el Nivel inferior y al subir de Nivel estas se agrupan por:

- **Grupo:** indicando el grupo al que pertenece la materia. En *Ánima* se tienen grupos por cada orientación en todos los niveles del bachillerato.
- **Nivel:** agrupa las materias por el nivel de bachillerato pudiendo ser uno, dos o tres.

Por otro lado, se observa en la Dimensión “Elementos Moodle” (Figura E.5) que los elementos en la plataforma Moodle son agrupados únicamente por el tipo de elemento pudiendo ser foro, tarea, cuestionario, recurso, entre otros.

En el caso de la Dimensión “Usuario” (Figura E.4) se tiene en el Nivel inferior los usuarios que pueden ser alumnos incluso los alumnos egresados, y también los docentes de la institución. Dentro del mismo Nivel se cuenta con algunos atributos con información de interés sobre los usuarios, estos son:

- **id Usuario:** contiene la cédula del usuario.
- **Nom Usuario:** es el nombre del usuario.
- **Centro Carrera Egresado:** contiene el nombre del centro de educación para las carreras terciarias que inicia el usuario en caso de ser egresado.
- **Empresa Egresado:** contiene el nombre de la empresa donde trabaja el usuario en caso de ser egresado.
- **Año Alta Docente:** contiene el año en que el usuario ingresó a la institución en caso de ser docente.
- **Edad:** tiene la edad del usuario.

Al subir de Nivel por la Jerarquía los usuarios son agrupados por diversas características, en algunos casos estas características aplican solo a alumnos, egresados o docentes. Las características específicas para los alumnos son:

- **Orientación:** pudiendo ser TIC o ADM.
- **Generación:** que contiene el año de generación en que ingresó el alumno en *Ánima*.
- **Sexo:** que contiene el sexo del alumno.
- **INSE:** que contiene el índice de INSE para el alumno.
- **MIDES:** binario que indica si el alumno viene de MIDES.
- **Aprendiz:** binario que indica si el alumno es aprendiz realizando prácticas en empresas formadoras.
- **Tanda:** contiene el número de tanda en que el alumno comienza como aprendiz dentro de su generación pudiendo ser la primera, segunda o tercera.
- **Nivel Educación Referente:** es el máximo nivel de educación alcanzado por el referente o tutor del alumno.

En el caso que los alumnos sean egresados se tienen además las características:

- **Seguimiento Egresado:** es un binario que indica si se le realizó seguimiento al alumno luego de egresar.
- **Bachiller Egresado:** indica si es bachiller.
- **Situación Ocupacional Egresado:** indica si está estudiando y/o trabajando.

- **Categoría Estudio Egresado:** contiene la situación actual de sus estudios.
- **Categoría Trabajo Egresado:** contiene la situación laboral actual del egresado.
- **Prioridad:** puede ser Alta, Media o Baja de acuerdo con la prioridad del egresado para *Ánima* de acuerdo con sus necesidades.

Por último, para distinguir si el usuario es alumno, egresado o docente se utiliza los Niveles “Rol” y “Estado”. El Nivel “Rol” indica si el usuario es docente o alumno. En el caso de ser alumno, “Estado” indica si es activo dado que está cursando actualmente, inactivo por abandono antes de terminar el bachillerato o egresado en caso de haber egresado de la institución. También se cuenta con el Nivel “Subestado” que puede tener los siguientes valores:

- **activo:** si es alumno actualmente.
- **deserción:** si abandona la institución y no continua los estudios.
- **desvinculación no repite:** si abandona la institución por decisión propia, pero continúa sus estudios.
- **desvinculación repite:** si abandona la institución por repetición, pero continúa sus estudios.
- **bachiller *ánima*** si termina sus estudios en *Ánima*.
- **bachiller:** si cursó los tres años en *Ánima*, pero termina en otra institución.
- **recursa:** si cursó los tres años en *Ánima* y actualmente se encuentra terminando sus estudios en otra institución.
- **no estudia:** si cursó los tres años en *Ánima* y decidió abandonar los estudios.

6.2. Relaciones Dimensionales

En esta sección se detallarán las Relaciones Dimensionales que se utilizaron para el diseño del Data Warehouse. Los diagramas correspondientes a las Relaciones Dimensionales se encuentran en el Apéndice E.1. Se tienen las siguientes Relaciones Dimensionales:

- **Resultado Entrega Tareas:** permite filtrar los resultados de entregas de las tareas en Moodle por las características de las mismas, por cada alumno, por los cursos y por el docente de los mismos. La Figura E.7 contiene el diagrama correspondiente.
- **Cantidad Alumnos:** permite filtrar la cantidad de alumnos por curso, fechas en que se contabilizan y por las características de los alumnos. El diagrama se puede ver en la Figura E.8.
- **Boletín:** cada indicador en los boletines puede ser filtrado por el tiempo en el que se entregaron, los alumnos, los cursos y los docentes de los mismos. En la Figura E.9 se encuentra el diagrama correspondiente.

- **Competencias:** las notas de las competencias de los alumnos pueden verse a nivel de alumno, cursos, y el tiempo que se entregan. El diagrama se encuentra en la Figura E.10.
- **Inasistencias Alumnos:** la Medida de inasistencias de alumnos puede ser vista por cada alumno, por cursos y la fecha que se registran las mismas. La Figura E.11 contiene el diagrama de la misma.
- **Inasistencias Clase Docente:** las inasistencias a clase de los docentes son filtradas por docente y el tiempo en que se registran estas. En la Figura E.12 se encuentra el diagrama correspondiente.
- **Inasistencias Coordinaciones Docente:** en este caso se filtran las inasistencias a coordinaciones para cada docente y el tiempo en que se registran estas. El diagrama se halla en la Figura E.13.
- **Asignaturas Docente:** la cantidad de asignaturas de los docentes se puede hallar para cada docente y en la fecha de interés. El diagrama asociado a la misma se puede ver en la Figura E.14.
- **Cantidad Elementos Moodle:** esta cantidad se permite filtrar por la fecha en que extrae la información, el tipo de elemento, los cursos y los docentes de los mismos al que pertenecen los elementos. El diagrama se encuentra en la Figura E.15.
- **Cantidad Entrevistas:** las cantidades de entrevistas realizadas a los egresados pueden ser filtradas por cada alumno egresado y el tiempo de su realización. La Figura E.16 posee el diagrama de la misma.
- **Cantidad Recomendaciones:** en este caso se tienen las cantidades de recomendaciones pudiéndose filtrar por los alumnos egresados y el tiempo en que se efectúan. La Figura E.17 contiene el diagrama correspondiente a esta Relación Dimensional.
- **Cantidad Accesos Moodle:** las cantidades de accesos a Moodle se pueden filtrar por el usuario de la plataforma Moodle, por fecha, por cursos y docentes de los mismos. Dentro de la Figura E.18 se encuentra el diagrama para este caso.
- **Cantidad Acciones:** en el caso de las acciones tomadas se pueden ver esas cantidades a nivel de cada alumno egresado, y la fecha en que se hacen. La Figura E.19 contiene el diagrama asociado a la misma.
- **Cantidad Intervención Laboral:** las intervenciones laborales que Anima realiza a los egresados se pueden ver a nivel de alumno egresado y el tiempo en que se realiza la misma. En este caso la Figura E.20 contiene el diagrama correspondiente.
- **Cantidad Mensajes Foro Moodle:** la cantidad de mensajes de los foros de la plataforma de Moodle se puede filtrar por el usuario autor del mensaje, la fecha en que se enviaron, por cursos y docentes que los dictan. La Figura E.21 contiene el diagrama correspondiente.

Cada Relación Dimensional está compuesta por un conjunto de Dimensiones ubicadas hacia la izquierda de los diagramas y por un conjunto de Medidas ubicadas hacia a la derecha. En estos diagramas se puede observar que la Medida va a tomar distintos valores al fijar los Niveles de las Dimensiones a la izquierda de acuerdo con las Jerarquías de las mismas.

En todas las Relaciones Dimensionales aparece la Dimensión “Tiempo” y esto refleja la capacidad de los Data Warehouse de almacenar datos históricos.

Por otro lado, las Relaciones Dimensionales que tienen la Dimensión “Usuario” se especifica qué rol cumple el usuario. Algunos casos tienen el rol de alumno, otros de docente y los restantes pueden tomar más de un rol. En especial, para las Relaciones Dimensionales “Resultado Entrega Tareas”, “Cantidad Accesos Moodle”, “Cantidad Mensajes Foro Moodle” aparece dos veces la Dimensión “Usuario”, cada una con su rol correspondiente y esto es porque interesa saber el Docente de la materia de donde se extraen esos datos y además el alumno o usuario al que pertenecen los mismos.

Por ejemplo, la Relación Dimensional “Resultado Entrega Tareas” que se observa en la Figura 6.3 está compuesta por las Dimensiones “Usuario”, por un lado, con el rol de docente y por otro lado con el rol de alumno, “Académico”, “Tiempo”, y “Entrega Tarea”. A su vez contiene la Medida “M_ResultadoEntregasTareas”. A través de la misma se puede observar que la cantidad de entregas de tarea en Moodle y el promedio de calificaciones de las mismas puede variar a lo largo del tiempo dentro de la institución dado que se realizan entregas para las nuevas tareas que se requieren año a año. A su vez, al navegar por las Jerarquías de las Dimensiones se puede obtener distinta granularidad para la cantidad de entregas de tareas y el promedio de calificación.

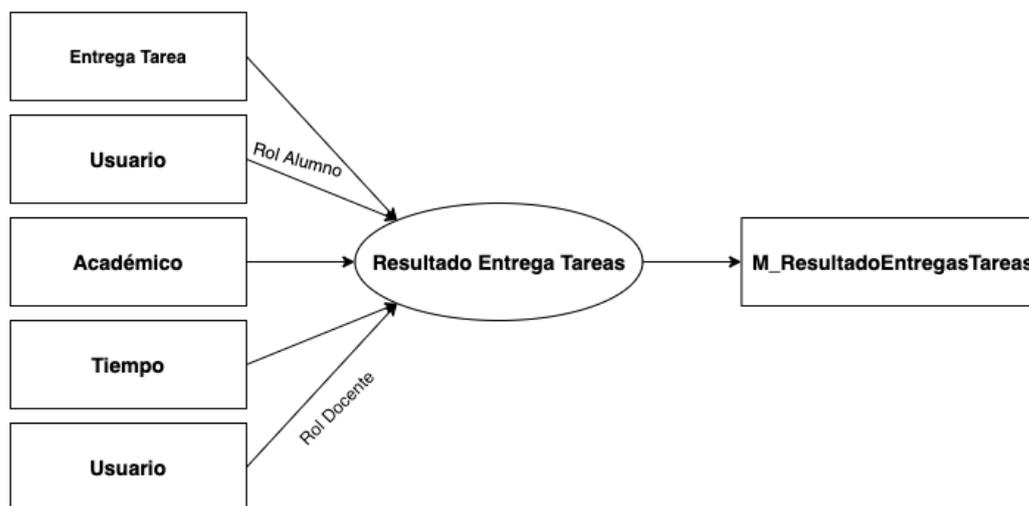


Figura 6.3: Relación Dimensional “Resultado Entrega Tareas”.

6.3. Funciones de agregación

En esta sección se presenta un análisis de las funciones de agregación para aplicar a cada Medida al atravesar las Jerarquías de sus correspondientes Dimensiones especificadas anteriormente en las Relaciones Dimensionales dentro de la Sección 6.2.

A continuación, se presentan algunas aclaraciones en cuanto a las funciones de agregación para las Medidas consideradas como semi aditivas.

La Medida “M_ResultadoEntregasTareas” en el indicador “CantidadEntregasTareas” es aditiva para todas las Dimensiones y con el indicador “PromedioCalificacion” se utiliza el promedio para ser agregada en todas las Dimensiones.

En el caso de la Medida “M_CantidadAlumnos” es aditiva en la Dimensión “Usuario”. Pero con respecto a las Dimensiones “Tiempo” y “Academico” esta Medida no se puede agregar con la operación suma dado que pueden aparecer duplicados al subir Niveles por la Jerarquía. De todas formas, se debe solucionar, dado que es de interés para obtener reportes. Una alternativa es tomar los alumnos con distintos identificadores previniendo duplicados, es decir realizar un conteo de alumnos distintos.

Las Medidas “M_BoletinAlumno” y “M_Competencias” son agregadas utilizando el promedio para todas las Dimensiones, exceptuando la Dimensión “Tiempo”. Para esta última no es posible agregarlas dado que sólo tiene sentido fijar el Nivel de “Entrega” en este contexto, debido a que las calificaciones de los alumnos tienen períodos específicos donde se entregan a través de los boletines a los alumnos. De esta manera la fecha no es relevante agruparla a Nivel de “Semana” o “Mes” y por otro lado las entregas son independientes entre sí impidiendo agruparlas también a Nivel de “Año”.

Para el caso de la Medida “M_InasistenciasAlumno” es aditiva para todas las Dimensiones exceptuando la Dimensión “Tiempo” para el caso del indicador “CantidadInasistencias”. Esta excepción se da al agrupar las fechas en los Niveles “Semana” o “Mes” donde se debe tener en consideración llegadas tarde y otros tipos de inasistencias en la institución que no se pueden resolver con funciones de agregación básicas, sino que se resuelve con una nueva función particular para este caso.

Por último, la Medida “M_AsignaturasDocente” es aditiva, pero hay algunas consideraciones para tener en cuenta. Dentro de la Dimensión “Usuario”, la única agregación de interés se encuentra en el Nivel “Rol” para agrupar a los usuarios que son docentes. Por otro lado, para la Dimensión de “Tiempo” la única agregación de interés se encuentra en el Nivel “Año” dado que las asignaturas para un docente se obtienen de forma anual.

El resto de las Medidas que no fueron mencionadas se identificaron como aditivas en todas sus Dimensiones.

6.4. Diseño Lógico

Aquí se presenta el diagrama del diseño lógico obtenido luego de realizar todo el proceso detallado en las secciones anteriores. Como se puede observar en el diseño resultante en la Figura 6.4, cada una de las distintas Dimensiones que fueron analizadas en la Sección 6.1 fueron mapeadas a tablas en la base de datos. Por ejemplo, la Dimensión “Usuario” es representada en la tabla “usuario”, la cual es una de las más relacionadas con casi el total de las restantes existentes en el diseño. Estas tablas son las que se encuentran a la derecha del diagrama del diseño lógico. Por otro lado, las Relaciones Dimensionales también fueron mapeadas a una tabla de Hechos como es el caso de la Relación Dimensional “Cantidad Alumnos” que es representada por la tabla “cantidad_alumnos”. Éstas

son las tablas que se encuentran a la izquierda en el diseño que se visualiza en la Figura 6.4. Cada tabla de Hechos contiene la Medida que ésta representa y además contiene las relaciones a las Dimensiones correspondientes.

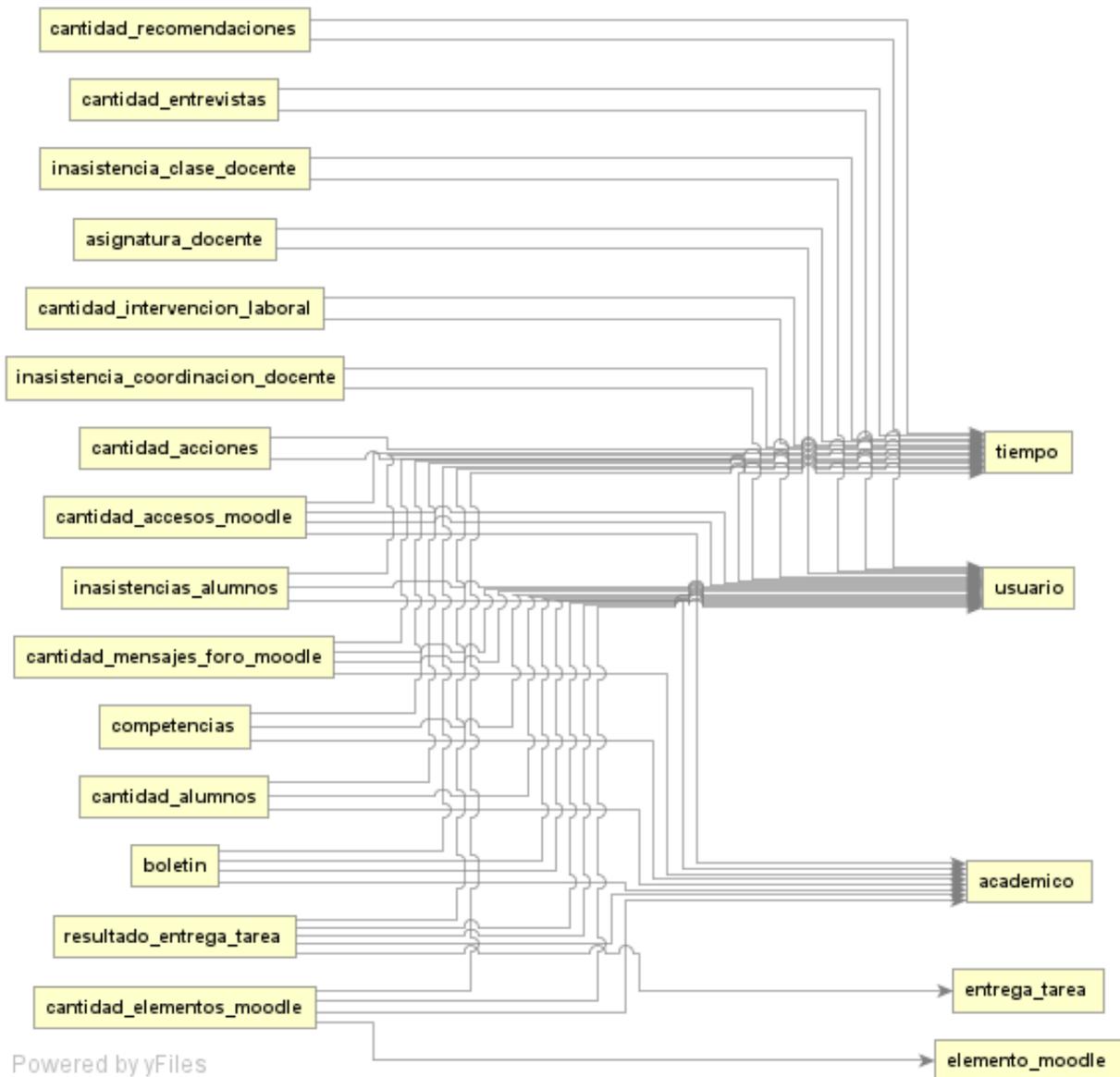


Figura 6.4: Diseño Lógico de Data Ánima

Dentro del diseño se sigue el esquema Estrella al contemplar cada una de las tablas de Hechos individualmente. Por ejemplo, como se observa en la Figura 6.5 la tabla de Hecho “resultado_entrega_tarea” sigue el esquema Estrella dado que posee Foreign Keys a las tablas de Dimensiones “usuario”, “tiempo”, “academico” y “entrega_tarea”.

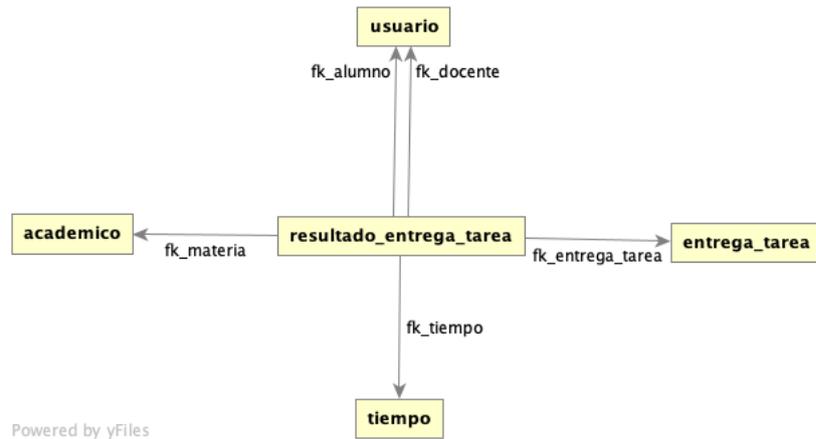


Figura 6.5: Esquema Estrella de Hecho “resultado_entrega_tarea”.

Cada una de las tablas correspondientes a las Dimensiones se encuentran en la Figura 6.6, las mismas contienen cada uno de sus atributos y el tipo de los mismos.

usuario	
ci	CHARACTER VARYING
email	CHARACTER VARYING
nombre	CHARACTER VARYING
centro_carrera_egresado	CHARACTER VARYING
empresa_egresado	CHARACTER VARYING
orientacion	CHARACTER VARYING
generacion	INTEGER
subestado	CHARACTER VARYING
estado	CHARACTER VARYING
inse	CHARACTER VARYING
tanda	INTEGER
aprendiz	BOOLEAN
sexo	CHARACTER VARYING
mides	BOOLEAN
nivel_educacion_referente	CHARACTER VARYING
barrio	CHARACTER VARYING
seguimiento_egresado	BOOLEAN
situacion_ocupacional_egresado	CHARACTER VARYING
categoria_estudio_egresado	CHARACTER VARYING
categoria_trabajo_egresado	CHARACTER VARYING
prioridad_egresado	CHARACTER VARYING
bachiller_egresado	BOOLEAN
edad	INTEGER
anio_alta_docente	INTEGER
rol	CHARACTER VARYING

tiempo	
fecha	TIMESTAMP(6) WITHOUT TIME ZONE
semana	INTEGER
mes	INTEGER
entrega	INTEGER
anio	INTEGER
id	BIGINT

entrega_tarea	
id	INTEGER
calificacion	INTEGER
retroalimentacion	BOOLEAN
estado	CHARACTER VARYING
resultado	CHARACTER VARYING
entregado_en_fecha	BOOLEAN

academico	
nombre_materia	CHARACTER VARYING
id_materia	INTEGER
nombre_grupo	CHARACTER VARYING
nivel	INTEGER
id_grupo	CHARACTER VARYING

elemento_moodle	
id	INTEGER
nombre	CHARACTER VARYING
tipo	CHARACTER VARYING

Figura 6.6: Tablas de Dimensiones.

Por último, las tablas de Hechos se pueden ver en la Figura 6.7, que contienen las Medidas y las referencias a las Dimensiones correspondientes. En este caso también se indica cuál es el tipo de datos de cada atributo. La clave de cada una

de estas tablas está compuesta por el conjunto de sus Foreign Keys a las tablas de Dimensiones.

cantidad_mensajes_foro_moodle	cantidad_accesos_moodle	boletin
cantidad_mensajes_foro_moodle INTEGER	cantidad_accesos_moodle INTEGER	promedio_notas DOUBLE PRECISION
ci_usuario CHARACTER VARYING	ci_usuario CHARACTER VARYING	promedio_bajas DOUBLE PRECISION
ci_docente CHARACTER VARYING	ci_docente CHARACTER VARYING	promedio_examenes DOUBLE PRECISION
id_materia INTEGER	id_materia INTEGER	ci_alumno CHARACTER VARYING
id_tiempo BIGINT	id_tiempo BIGINT	ci_docente CHARACTER VARYING
		id_materia INTEGER
		id_tiempo BIGINT
inasistencias_alumnos	cantidad_elementos_moodle	cantidad_acciones
cantidad_inasistencias DOUBLE PRECISION	cantidad_elementos INTEGER	cantidad_acciones INTEGER
cantidad_horas_totales DOUBLE PRECISION	id_elemento_moodle INTEGER	ci_alumno CHARACTER VARYING
ci_alumno CHARACTER VARYING	ci_docente CHARACTER VARYING	id_tiempo BIGINT
id_materia INTEGER	id_tiempo BIGINT	
id_tiempo BIGINT	id_materia INTEGER	
competencias	inasistencia_clase_docente	resultado_entrega_tarea
promedio_competencias DOUBLE PRECISION	cantidad_inasistencias DOUBLE PRECISION	cantidad_entregas INTEGER
ci_alumno CHARACTER VARYING	carga_horaria_total DOUBLE PRECISION	promedio_calificacion DOUBLE PRECISION
id_materia INTEGER	ci_docente CHARACTER VARYING	id_materia INTEGER
id_tiempo BIGINT	id_tiempo BIGINT	ci_alumno CHARACTER VARYING
		id_tiempo BIGINT
		ci_docente CHARACTER VARYING
		id_entrega_tarea INTEGER
inasistencia_coordinacion_docente	cantidad_alumnos	asignatura_docente
cantidad_inasistencias DOUBLE PRECISION	cantidad_alumnos INTEGER	cantidad_asignaturas INTEGER
carga_horaria_total DOUBLE PRECISION	ci CHARACTER VARYING	ci_docente CHARACTER VARYING
ci_docente CHARACTER VARYING	id_materia INTEGER	id_tiempo BIGINT
id_tiempo BIGINT	id_tiempo BIGINT	
cantidad_intervencion_laboral	cantidad_recomendaciones	cantidad_entrevistas
cantidad_intervencion_laboral INTEGER	cantidad_recomendaciones INTEGER	cantidad_entrevistas INTEGER
ci_alumno CHARACTER VARYING	ci_alumno CHARACTER VARYING	ci_alumno CHARACTER VARYING
id_tiempo BIGINT	id_tiempo BIGINT	id_tiempo BIGINT

Figura 6.7: Tablas de Hechos.

6.5. Implementación de un prototipo para la verificación del Data Warehouse

Una vez realizado el diseño y alcanzado la definición del Script de la base de datos del Data Warehouse como se describió en las secciones anteriores de este mismo Capítulo, se validó el mismo con los referentes de *Ánima*. A partir de esto se decidió por parte del grupo de trabajo realizar un pequeño prototipo que verificara el resultado obtenido para el tercer objetivo planteado en este proyecto. El mismo está enfocado en almacenar datos y realizar la integración del Data Warehouse con Google Data Studio. Dicho prototipo resultó una prueba de concepto ya que no comprendía todas las fuentes de datos que se debían tener en cuenta. Dentro del prototipo implementado se validó con Google Data Studio el acceso al Data Warehouse como fuente de datos debido a que esta plataforma brinda la posibilidad de obtener la información con la cual trabajar

desde distintos tipos de fuentes de datos. Dado que la base de datos se logró implementar en el servidor de Amazon brindado por *Ánima*, es que se puede tener acceso desde Google Data Studio para consultar los datos que allí se almacenen.

El problema aquí era que se tenía una base de datos vacía, ya que no se contaba con un proceso ETL que obtuviera los datos de los archivos excel, ya mencionados anteriormente, y los almacenara en la base de datos. Para poder realizar una prueba rápida de esto, se tomó la decisión de cargar únicamente los datos obtenidos desde Moodle, ya que son los que maneja el prototipo Data *Ánima* que se construyó y que estos son transformados para armar el excel en conveniencia para que coincidan con el formato que se pide en la base de datos. Dado esto, es que se implementó un prototipo con una arquitectura y tecnologías similares al prototipo de extracción de datos de Moodle pero que, en lugar de interactuar con la API de Moodle, solo recibe el excel con la información e inserta la misma en la base de datos del Data Warehouse.

Una vez con los datos pertenecientes a Moodle en la base de datos se pudo corroborar que efectivamente se podían consultar de igual manera que con el excel anteriormente manejado. El único cambio que se debe tener en cuenta es que la consulta a los datos para realizar los cuadros de mando es a través de consultas SQL. Las consultas SQL realizadas fueron hacia las tablas de Hechos realizando JOINS con las tablas de Dimensiones correspondientes. En el siguiente fragmento se puede observar un ejemplo de las consultas SQL realizadas, en este caso se puede ver la que fue utilizada para obtener los resultados de entregas de tareas en la plataforma Moodle.

```
SELECT
    alumno.email alumno_email,
    alumno.nombre alumno_nombre,
    academico.nombre_materia materia,
    tiempo.fecha fecha,
    tiempo.anio anio,
    tiempo.mes mes,
    docente.nombre docente_nombre,
    entrega_tarea.estado estado_entrega
FROM resultado_entrega_tarea
INNER JOIN entrega_tarea
    ON entrega_tarea.id = resultado_entrega_tarea.id_entrega_tarea
INNER JOIN tiempo
    ON tiempo.id = resultado_entrega_tarea.id_tiempo
INNER JOIN academico
    ON academico.id_materia = resultado_entrega_tarea.id_materia
INNER JOIN usuario alumno
    ON alumno.ci = resultado_entrega_tarea.ci_alumno
INNER JOIN usuario docente
    ON docente.ci = resultado_entrega_tarea.ci_docente;
```

El resto de las consultas SQL realizadas para el prototipo se encuentran en el Apéndice [E.4](#).

Luego de probar armar cuadros de mando con la fuente de datos correspondiente a la base de datos construida, se encontraron algunas ventajas en comparación a armar los cuadros de mando a partir de los archivos excel. Por ejemplo,

la principal ventaja encontrada, es que una vez que se obtienen nuevos datos y estos se insertan en la base de datos simplemente con ingresar al cuadro de mando que se tenía anteriormente y recargarlo, ya se pueden consultar los nuevos datos obtenidos. Esto evita tener que volver a crear un nuevo cuadro de mando o reutilizar el cuadro de mando ya creado, pero cambiándole la fuente de datos. En el caso de los archivos excel, si se obtienen nuevos datos, es decir un nuevo archivo excel, para consultar estos datos se debe crear un nuevo cuadro de mando en Google Data Studio o cambiar la fuente de datos del cuadro de mando que ya se tenía. Otra ventaja, que es consecuencia de lo anteriormente mencionado, es que teniendo la información en la base de datos se evita tener varias versiones de cuadros de mando ya que con uno solo se logra consultar toda la información.

Algunas consideraciones que se deben tener en cuenta de este prototipo construido es que se debieron fijar algunos datos ficticios en la base ya que no se contaba con cierta información, por ejemplo, no se contaba con las cédulas de los alumnos y docentes, y estas eran requeridas por el diseño de la base de datos. Por esto es que se decidió fijar los id de Moodle de los usuarios en lugar de sus cédulas ya que estas no eran estrictamente necesarias para realizar la prueba. Otra gran consideración para tener en cuenta es que el servidor que se tenía dónde se encuentra el sistema construido y la base de datos, no contaba con la suficiente memoria para ejecutar el prototipo con éxito. Por esta razón es que se tuvieron que hacer las pruebas desde fuera del servidor en un ambiente de desarrollo y posteriormente en un servidor diferente al utilizado para la base de datos a fin de hacerlo accesible por la institución.

En el Apéndice **C** se encuentra un manual de instalación y de usuario para el prototipo de Data Warehouse utilizado.

Capítulo 7

Extracción, transformación y carga de datos

En este capítulo se detalla el proceso de extracción, transformación y carga (ETL) diseñado como parte de la solución planteada para el prototipo de extracción de Moodle que resuelve el segundo objetivo del proyecto 7.1. A su vez se presentan diagramas del ETL para el resto de los sistemas informáticos utilizados por *Ánima* a fin de cargar el Data Warehouse propuesto, el cual resuelve el tercer objetivo del proyecto 7.3.

7.1. Moodle

Durante el período de investigación del proyecto se estudiaron las distintas formas de extracción de información de la plataforma Moodle. Se analizó si existía dentro de Moodle una forma de extraer la información en formato excel o CSV que cumpliera con los requerimientos de *Ánima*, para evitar implementar un nuevo método de extracción. En este punto se encontró una forma de realizar un respaldo de cada uno de los elementos de la plataforma, el cual se obtenía en un formato no amigable. Esta alternativa se descartó ya que la misma no estaba diseñada para extraer información. Además, el formato del archivo obtenido en este método de extracción puede sufrir cambios con el correr del tiempo, lo cual requeriría mantenimiento en el sistema que lo procesa para considerar los mismos.

Luego se investigó la capacidad de extracción de la información de Moodle por medio de la API provista por la plataforma. En este punto se analizó la API de Moodle identificando si esta podía proveer la información requerida por la institución. Dentro del sitio de Moodle [4] se encontró documentación sobre cómo realizar la configuración y los servicios que provee con sus formatos y contenidos, estos pasos se detallan en el Apéndice B.2. Al mismo tiempo se estudiaron cuáles son los servicios que estaban disponibles para la versión de Moodle utilizada por *Ánima*, ya que dependiendo de la versión utilizada puede variar el conjunto de servicios que se encuentran disponibles [24]. Luego de analizar la API provista por Moodle, se decidió avanzar con su utilización ya que la versión utilizada por *Ánima* permitía realizar gran parte de los requerimientos. Esta solución tiene como beneficio un bajo mantenimiento del sistema desarrollado ya que no habrá grandes cambios en el tiempo en la API de Moodle. Por otro lado, si en un futuro se decide migrar a una versión más nueva de Moodle se pueden agregar nuevos requerimientos que surjan con el tiempo en la institución gracias a las nuevas

funcionalidades que proveen las últimas versiones. De hecho, por parte de la institución se mostró interés de migrar a una versión más nueva en un futuro, lo que implicaría un mínimo mantenimiento de las funcionalidades utilizadas.

7.1.1. API de Moodle

El sistema Moodle provee una API, a partir de la cual se obtuvo toda la información requerida por *Ánima*. Debido a que la documentación provista por Moodle no es del todo clara, se tuvo que realizar una búsqueda más profunda del código fuente de esta plataforma, dado que la misma es de código abierto. También se investigó en foros [10] y en un proyecto de tesis que se tenía a disposición [11], que consume servicios de esta API. Además, se realizaron pruebas en la plataforma Postman [33] donde se analizaba la forma en la que se deben realizar las consultas, la manera de pasar los parámetros requeridos y el formato en el cual se retorna la información. Estas pruebas se realizaron hacia el Moodle de *Ánima* y hacia una instancia de Moodle web creada por el equipo de trabajo en un sitio que permite desplegar instancias de Moodle de prueba [25]. Dichas instancias fueron creadas para realizar pruebas de configuración de forma de no cometer errores en el Moodle de *Ánima*, y asimismo realizar pruebas de funcionamiento del prototipo implementado.

Dado que el usuario provisto por *Ánima* mencionado en la Sección 3.5.1 contaba con permisos para realizar cambios en la configuración dentro de Moodle, fue posible configurar la aplicación para habilitar el uso de la API, definir qué roles de usuarios pueden acceder a esta y qué configuración va a ser utilizada. Todas estas configuraciones están definidas en el Apéndice B en las Secciones B.2 y B.3.

A continuación, se detallan las funciones de la API de Moodle utilizadas para la obtención de la información:

- **Obtención del Token:** para la ejecución de cada una de las funciones de la API es requerido un token. Este identifica al usuario y valida los permisos que tiene dentro de Moodle. Por lo tanto, antes de realizar cada una de las consultas es necesario obtener un token que identifica al usuario. Para esto se debe ejecutar una consulta con el formato descrito en el Apéndice D.1.3. De la respuesta obtenida se almacena el campo del “token” para luego poder ser utilizado en el resto de las consultas. Si el token utilizado no tiene los permisos suficientes para su ejecución, las consultas van a responder con un mensaje de error indicando que el usuario no puede acceder a esa funcionalidad. De esta manera solo los usuarios con los roles habilitados a usar los servicios pueden utilizar Data *Ánima*.
- **Obtención de información de un usuario:** retorna la información de los usuarios aplicando uno o más filtros. Para aplicar estos filtros se le especifica los atributos de usuario y los valores de los mismos. Para la ejecución de esta función se puede observar un ejemplo en el Apéndice D.2. En el sistema esta función es utilizada únicamente para verificar si un usuario tiene los permisos necesarios para acceder al sistema.
- **Obtención de información de los cursos:** devuelve todos los cursos creados en el sistema. El formato de la consulta y un ejemplo de respuesta puede

ser encontrado en el Apéndice [D.3](#). De todos los campos que se obtienen en la respuesta solo son utilizados los detallados en el Apéndice [D.3.3](#).

- **Obtención de los contenidos de un curso:** este servicio, a partir del identificador de un curso, devuelve la información completa de todos los contenidos del mismo. Los contenidos de los cursos están organizados por secciones; cada una de ellas pueden contener distintos elementos. Los elementos tienen un tipo y este puede tomar el valor de foro (“forum”), tarea (“assign”), dirección web (“url”), entre otros. Las características que se obtienen de los mismos dependen del tipo de elemento. La obtención de estos elementos fue importante para la realización del excel ya que a partir de esto se puede saber qué tareas son planteadas por los docentes, cuáles de estas están disponibles para los alumnos, y a su vez se obtienen las tareas y el tipo de estas (“modname”) que más adelante es utilizado para obtener las entregas de los alumnos.

El formato para la ejecución de la consulta y un ejemplo de respuesta puede ser encontrado en el Apéndice [D.4.1](#). En el Apéndice [D.4.2](#) se encuentra la información utilizada de las secciones de los cursos obtenidos y de los elementos de dichas secciones.

- **Obtención de información de las entregas:** esta función devuelve información sobre las entregas realizadas por los usuarios para cada una de las tareas a partir de una lista de identificadores de elementos de tipo “assign”, los cuales se obtuvieron en base a la función previamente descrita. A partir de esta información se obtienen métricas sobre la cantidad de tareas realizadas por un alumno, por lo tanto, acompañado de otra información se puede sacar conclusiones de la participación de este en cada uno de los cursos.

El formato para realizar esta consulta y un ejemplo de respuesta puede ser encontrado en el Apéndice [D.5.1](#). De cada una de las respuestas se utiliza la información descrita en el Apéndice [D.5.2](#).

- **Obtención de la calificación de las entregas:** en esta función se retornan las calificaciones que realizan los docentes a las entregas de los usuarios para cada una de las tareas. Esto permite determinar qué docentes hacen calificaciones y cuáles no, como también qué uso se le da al sistema de calificación de tareas en Moodle.

El Apéndice [D.6.1](#) contiene el formato para realizar esta consulta y un ejemplo de respuesta. De cada una de las respuestas se utiliza la información descrita en el Apéndice [D.6.2](#).

- **Obtención de información de los usuarios en un curso:** dicha funcionalidad toma un identificador de un curso y devuelve todos los usuarios que están matriculados al mismo actualmente, así como también la información detallada de todos estos usuarios. En Data Ánima es utilizada para obtener todos los usuarios que están activos en el sistema y su información. Para esto se recorre todos los cursos y para cada uno de los cursos se obtienen los usuarios que están matriculados a este. Luego la información de los usuarios es almacenada controlando las repeticiones, ya que un usuario puede

estar matriculado a más de un curso al mismo tiempo. De esta forma se puede conseguir la información de todos los usuarios y los cursos a los cuales están matriculados cada uno de estos. Esta información es relacionada con las entregas, foros y cursos logrando que se pueda obtener la actividad de los alumnos dentro de la plataforma y el trabajo que realizan los docentes.

Al mismo tiempo con las respuestas de esta consulta se guarda el rol que tienen los usuarios para cada uno de los cursos, porque no se cuenta con un rol global de la plataforma, sino que con un rol por curso. Esta es una característica para destacar debido a que esta forma de obtener los roles en cada curso no la provee Moodle directamente en un servicio de la API. Por otro lado, se tiene la limitante de que únicamente se puede obtener los roles que tienen los usuarios para cada uno de los cursos que están matriculados al momento de hacer la consulta, lo cual impide obtener un histórico de los mismos.

En el Apéndice [D.7.1](#) se detalla el formato de la consulta y un ejemplo de respuesta. La información utilizada de los usuarios en cada consulta y la información de los roles almacenada se detalla en el Apéndice [D.7.2](#).

- **Obtención de información de los foros de un curso:** dada una lista de identificadores de cursos, devuelve los foros que pertenecen a estos cursos. La información obtenida en esta consulta complementada con las dos consultas que se explican a continuación es utilizada para entender la interacción entre los usuarios y qué participación tienen los alumnos y docentes en la plataforma. La información de los foros es relacionada con los cursos, los usuarios y roles para poder obtener la realidad de cada uno de estos componentes en la plataforma.

En el Apéndice [D.8.1](#) se puede encontrar el formato que debe tener la consulta y un ejemplo de respuesta, de estas respuestas es utilizada la información detallada en el Apéndice [D.8.2](#).

- **Obtención de información de las discusiones de un foro:** a través del identificador de un foro, devuelve toda la información de las discusiones creadas en el foro. Las discusiones son preguntas o planteos que se realizan dentro de un foro. Estas tienen un título, una descripción y la capacidad de agregar archivos adjuntos.

El formato necesario para realizar esta consulta y un ejemplo de respuesta puede ser encontrado en el Apéndice [D.9.1](#). La información utilizada de estas discusiones se detalla en el Apéndice [D.9.2](#).

- **Obtención de información de los posts de una discusión:** a partir del identificador de una discusión, devuelve toda la información de los posts creados en la misma. Los usuarios pueden responder a cada una de las discusiones en forma de post, creando una conversación. De esta forma se puede identificar la participación de los usuarios dentro de los cursos, es decir, la actividad que tienen los docentes como los alumnos en los cursos. El formato requerido para realizar esta consulta y un ejemplo de respuesta puede ser encontrado en el Apéndice [D.10.1](#). La información almacenada de los posts es la detallada en el Apéndice [D.10.2](#).

7.1.2. Formato de excel

A continuación, se define el formato del archivo excel utilizado para almacenar la información obtenida de Moodle. Este fue definido de forma que pueda ser entendido y analizado por los usuarios y al mismo tiempo que cumpla con ciertas características que permitan realizar las consultas deseadas en Google Data Studio. Para esto, se dividió el documento en varias hojas. Esto permite que la información quede más ordenada facilitando la búsqueda de forma intuitiva y se pueda comprender con mayor facilidad. Al momento de manipular la información dentro de Google Data Studio se identificaron restricciones que impedían la interconexión de la información entre más de dos hojas. Esto llevó a que se tenga que duplicar información en varias hojas, y de esta forma poder representar la información deseada con sus respectivos filtros.

Dentro de cada una de las hojas del archivo excel se encuentran filas que representan una instancia de la información obtenida y cada columna una característica de la misma.

El archivo excel resultante se divide en las siguientes ocho hojas:

- **Cursos:** almacena la información de los cursos de Moodle. Las características registradas de estos son:
 - **ID:** número identificador del curso auto generado por Moodle.
 - **NOMBRE:** nombre del curso asignado por *Ánima*.
 - **ABREVIACIÓN:** abreviación del nombre del curso.
 - **CATEGORÍA:** número identificador de la categoría a la que pertenece el curso. En el caso de *Ánima* las categorías son utilizadas para agrupar los cursos correspondientes a un año-orientación, por ejemplo: 1ro TIC.
 - **FECHA DE CREACIÓN:** fecha en la cual se creó el curso en la plataforma.
- **Elementos de Moodle:** contiene la información de cada uno de los elementos que se encuentran en un curso, esta es:
 - **ID:** número identificador asignado por Moodle para cada uno de los elementos.
 - **NOMBRE:** nombre del elemento asignado por el encargado del curso.
 - **TIPO:** nombre definido por Moodle para cada tipo de elemento explicado anteriormente en la Subsección [7.1.1](#).
 - **VISIBLE:** determinante de visibilidad. Establece si el elemento se encuentra visible para todo público en una instancia de tiempo.
 - **ID CURSO:** número identificador del curso al cual pertenece este elemento.
 - **NOMBRE CURSO:** nombre del curso al cual pertenece el elemento.
 - **FECHA OBTENCIÓN:** fecha de obtención de esta información por *Data Ánima*. Esta se define ya que los elementos no tienen fecha de modificación y a *Ánima* le interesa poder comparar diferentes extracciones entre sí en distintos periodos de tiempo a fin de analizar si los elementos pasan a estar visibles o nunca están visibles, entre otros posibles análisis.

- **Entregas:** las entregas representan los trabajos entregados por los alumnos a una tarea de Moodle. La información registrada de las entregas es la siguiente:
 - **ID:** número identificador de una entrega.
 - **TAREA ID:** número identificador de la tarea a la cual pertenece la entrega.
 - **CURSO ID:** número identificador del curso al cual pertenece la tarea.
 - **USUARIO ID:** número identificador del usuario que realizó la entrega.
 - **NOMBRE COMPLETO:** nombre completo del usuario que realizó la entrega.
 - **MAIL:** correo electrónico del usuario que realizó la entrega.
 - **NUMERO INTENTO:** cantidad de intentos de entrega realizados por el usuario para una tarea.
 - **FECHA CREACIÓN:** fecha de creación del primer intento de entrega del usuario.
 - **FECHA MODIFICACIÓN:** fecha de modificación del último intento de entrega del usuario.
 - **ESTADO:** estado de la entrega, las entregas pueden estar en tres diferentes estados, “new” (la entrega todavía no se realizó), “draft” (se agregó una entrega a la tarea, pero no se envió), “submitted” (se agregó una entrega a la tarea y se envió).
 - **GRUPO ID:** número identificador del grupo al cual corresponde la entrega en caso de que esta pueda ser realizada en grupo.
 - **CALIFICACIÓN:** calificación realizada por el docente a la entrega.
- **Usuarios:** los usuarios son las cuentas de Moodle de los alumnos, docentes y administradores de *Ánima*. De estos se guarda la siguiente información:
 - **ID:** número identificador del usuario.
 - **NOMBRE DE USUARIO:** nombre de usuario en la plataforma.
 - **NOMBRE:** nombre de la persona a la cual corresponde el usuario.
 - **APELLIDO:** apellido de la persona a la cual corresponde el usuario.
 - **NOMBRE COMPLETO:** nombre completo de la persona a la cual corresponde el usuario.
 - **MAIL:** correo electrónico de la persona a la cual corresponde el usuario.
- **Roles:** los roles habilitan distintas funcionalidades dentro de Moodle y son asignados a los usuarios por un administrador. Los usuarios pueden tener los siguientes roles para un curso específico: gestor (“manager”), alumno (“student”), docente (“teacher”) o docente editor (“editingteacher”). Esto es importante diferenciar ya que un usuario puede tener un rol distinto dependiendo del curso. De los roles se almacena las siguientes características:
 - **ID:** número identificador del rol.

- **NOMBRE:** nombre del rol.
- **NOMBRE CORTO:** abreviación al nombre del rol.
- **Usuario Rol Curso:** esta hoja relaciona un usuario con un rol y un curso. La razón por la cual se creó esta hoja fue para poder identificar el rol del usuario para un curso en el resto de las hojas, dado que en la hoja de “Roles” se mencionó que un usuario puede tener distintos roles para los diferentes cursos. Los elementos que contiene esta hoja son los siguientes:
 - **USUARIO ID:** número identificador del usuario.
 - **NOMBRE COMPLETO:** nombre completo del usuario.
 - **MAIL:** correo electrónico del usuario.
 - **ROL ID:** número identificador del rol que tiene el usuario para el curso.
 - **CURSO ID:** número identificador del curso en el cual está matriculado el usuario.
- **Foros:** los foros son un tipo de elemento de Moodle entre los que fueron detallados anteriormente [7.1.1](#). Estos permiten una comunicación entre los usuarios dentro de un curso. Dentro de cada foro los usuarios pueden crear posts que pueden contener tanto texto como imágenes y otros usuarios pueden responder de la misma forma. Para los foros se guardan los siguientes atributos:
 - **FORO ID:** número identificador del foro.
 - **NOMBRE FORO:** nombre del foro.
 - **FECHA MODIFICACION:** fecha de modificación del foro.
 - **CURSO ID:** número identificador del curso en el cual se encuentra el foro.
 - **NOMBRE CURSO:** nombre del curso en el cual se encuentra el foro.
 - **USUARIO ID:** número identificador del usuario que creó el post dentro del foro.
 - **NOMBRE COMPLETO:** nombre completo del usuario que creó el post dentro del foro.
 - **MAIL:** correo electrónico del usuario que creó el post dentro del foro.
 - **ROL ID:** rol del usuario que creó el post dentro del foro.
 - **ROL NOMBRE:** nombre del rol del usuario que creó el post dentro del foro.
 - **POST ID:** número identificador del post.
 - **POST FECHA CREACIÓN:** fecha de creación del post.
 - **POST FECHA MODIFICACIÓN:** fecha de modificación del post.
- **Fecha de Extracción:** es la fecha en la cual se realizó la extracción de la información de Moodle. Esto es un pedido de *Ánima* para poder tener un histórico de las extracciones realizadas y poder realizar futuras comparaciones. Esta hoja contiene solamente una columna, la cual contiene la siguiente información:

- **FECHA DE EXTRACCIÓN:** fecha en la cual se realizó la extracción de la información de Moodle.

7.2. Sistemas existentes

Se decidió utilizar los archivos excel que Ánima utilizaba como punto de partida de la solución planteada, dado que en la mayoría de los sistemas existentes Ánima estaba utilizando la información a través de los proveedores de los sistemas informáticos o exportando los formularios de Google a archivos excel. Cabe destacar que por no estar dentro del alcance del proyecto no se investigaron otras formas de extracción en los sistemas existentes, pero serían fácilmente adaptadas si existieran.

A continuación, en la Tabla 7.1 se presentan los archivos excel obtenidos de los sistemas utilizados por Ánima a ser utilizados en la extracción de los datos para su posterior transformación y carga al Data Warehouse.

Nombre Excel	Descripción	Sistema
Base de datos Docente	En este excel se encuentran los datos de los docentes de la institución. Se crea un nuevo excel por año. Contiene el nombre y cédula de los docentes, las asignaturas, las notas de exámenes y entregas con que calificaron a sus alumnos, su carga horaria, entre otros.	Excel Interno
MapeoAlumno Ci-Mail	Contiene el mapeo de cédulas y emails de Moodle de los alumnos de la institución.	Excel Interno
MapeoDocente Ci-Mail	Mapeo de cédulas y emails de Moodle de los docentes de la institución	Excel Interno
PlanillaNotas	Tiene las calificaciones de los alumnos y se obtiene en cada entrega de boletín por cada grupo de la institución. Se tienen los alumnos identificados con cédula y nombre completo, y la calificación que obtuvieron en cada materia.	SIGED
Año - Reporte de inasistencias V1	En este caso se tienen las inasistencias de los alumnos. Se presenta en varios formatos, pudiéndose ver todas en una lista, por semana y por grupo. En todos los casos se tiene al alumno identificado por nombre y cédula, y la materia a la que corresponde la inasistencia con el tipo de la misma.	SIGED

Base de datos Social	Información recabada por <i>Ánima</i> al momento de ingresar los alumnos a la institución para conocer su contexto social. Como información relevante se destaca el Índice de Nivel Socio Económico (INSE).	Google Form
Sistematización Información Egresados (Respuestas)	Datos básicos sobre los egresados de la institución, así como alguna información de seguimiento que se le realiza a alguno de ellos. Por ejemplo, si el egresado continuó con sus estudios, si consiguió trabajo, entre otras cosas.	Google Form
ANIMA Mes	Se encuentran las intervenciones que recibe un alumno. Las mismas están en una lista y se indica el alumno, quién hizo la intervención, y la descripción de la misma con la fecha en que se realizó.	INOA
transversales - Nro. entrega	Se concentra la información de competencias transversales de los alumnos de <i>Ánima</i> que son evaluados en sus pasantías laborales.	Evaluador

Cuadro 7.1: Archivos Excel utilizados por *Ánima*

7.3. Carga de Data Warehouse

Dado que quedó fuera del alcance de este proyecto la construcción del Data Warehouse diseñado y lograr implementar el proceso ETL que cargue los datos al mismo, se decidió implementar diagramas en los cuales se explique de forma detallada de dónde se deberían obtener los datos para cargar con información las distintas tablas de la base de datos del Data Warehouse. Esta decisión fue tomada ya que el equipo del proyecto dedicó mucho tiempo a analizar los distintos archivos excel de los sistemas informáticos en conjunto con los cuadros de mando que *Ánima* realiza en base a ellos, y esto podría servir para que en un futuro no se tuviera que analizar nuevamente con tanto detalle toda esta información, en el caso de que el proyecto sea retomado por otro equipo.

Un ejemplo de estos diagramas se puede observar en la Figura 7.1, donde muestra de qué archivos se puede obtener la información requerida por la tabla Usuario. Los mismos se realizaron porque el diseño de las tablas de la base de datos está basado en la información que *Ánima* tenía de los archivos excel de sus distintos sistemas informáticos y también de los nuevos requerimientos sobre Moodle. En ellos es donde se representa al centro del diagrama lo que sería el proceso ETL que se debe encargarse de tomar la información de los distintos archivos excel, transformarla al formato requerido por la base de datos y por último realizar la carga de la información resultante. Por esto es que a la izquierda se

muestran los archivos excel a utilizar para cargar la información necesaria a la tabla que se representa a la derecha del diagrama. Además, a cada archivo excel utilizado se le agrega una etiqueta con la cual se identifica al sistema informático que pertenece.

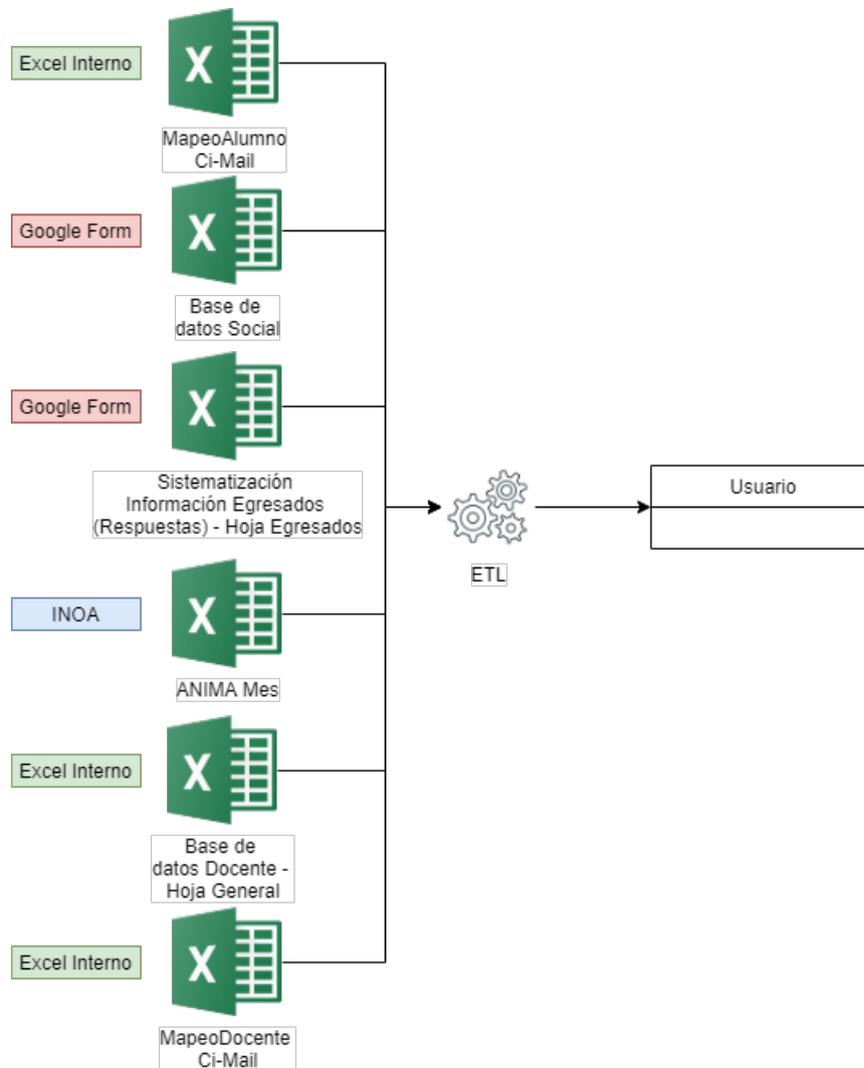


Figura 7.1: Proceso de ETL de Usuarios

El diseño del resto de los procesos ETL para las tablas que conforman la base de datos del Data Warehouse se pueden encontrar en la sección del Apéndice E.3.

Capítulo 8

Verificación y validación de resultados

En este capítulo se presenta por un lado la verificación del trabajo realizado para el segundo 8.1 y tercer 8.2 objetivo del proyecto, los cuales son la obtención de datos de Moodle y el diseño del Data Warehouse respectivamente. Por otro lado, se presenta la validación del proyecto en general 8.3 por parte de los usuarios finales.

8.1. Verificación de la obtención de datos de Moodle

Para la verificación de los datos obtenidos de Moodle se realizaron las pruebas detalladas en la Sección 5.5. A partir del trabajo de verificación realizado se puede concluir que, para el conjunto de datos utilizados para las pruebas, la extracción de los datos que se obtiene dentro del excel se realiza de forma correcta ya que son encontrados dentro de la plataforma de Moodle. Al mismo tiempo se concluye que para el conjunto de datos mencionado anteriormente para realizar las pruebas, toda la información presente en Moodle también se encuentra en el excel. Es decir, no hay información faltante dentro de los datos obtenidos por lo que se realiza la extracción de manera completa en estos casos.

8.2. Verificación del diseño del Data Warehouse

Para la verificación del diseño del Data Warehouse se realizó el prototipo detallado en la Sección 6.5. Teniendo en consideración que dentro del prototipo se utilizaron solamente los datos extraídos de la plataforma de Moodle, se concluye que el Data Warehouse fue diseñado satisfactoriamente para dar soporte al menos a esta gran parte de datos. Dado que el diseño realizado para los datos que se extraerían del resto de los sistemas que utilizan en Anima es similar al de Moodle, esto porque tienen en común las mismas Dimensiones, se concluye también que el diseño propuesto será de gran utilidad para la implementación total del Data Warehouse.

8.3. Validación de los usuarios

Para obtener una validación de parte de los referentes de Anima se realizó una encuesta de satisfacción, la cual fue entregada a los dos referentes de Anima

que participaron del proceso de dicho proyecto. Uno de los referentes que tuvo participación en el proyecto lleva el cargo de Referente pedagógico, mientras que el otro referente se encarga de la manipulación de datos.

Formulario de satisfacción

Pregunta 1: ¿Qué les pareció la experiencia de trabajar con alumnos de la Facultad de Ingeniería en un proyecto para el uso de ANIMA?

Respuesta referente 1: Excelente. Se identificó el problema y la respuesta se ajustó a las necesidades.

Respuesta referente 2: Muy buena.

Pregunta 2: Desde su punto de vista, ¿se encuentran cumplidos de buena manera los requerimientos obtenidos durante la etapa de análisis del proyecto?

Respuesta referente 1: Sí.

Respuesta referente 2: En buena parte sí.

Pregunta 3: Viendo el trabajo terminado, ¿qué tan satisfechos se encuentran con los resultados obtenidos?

Respuesta referente 1: Muy satisfecho.

Respuesta referente 2: Satisfecha.

Pregunta 4: Pensando a futuro, ¿piensan que van a darle uso a la herramienta desarrollada y que les será de utilidad?

Respuesta referente 1: Seguramente. Y a su vez, sirvió como insumo para próximos proyectos institucionales.

Respuesta referente 2: Pienso que sí.

Como conclusión a la encuesta realizada se puede decir que los referentes de *Ánima* quedaron satisfechos con el proyecto realizado y su intención es continuar utilizando el prototipo de extracción de Moodle. De ser posible en un futuro estarían dispuestos a continuar con el desarrollo para obtener una solución que abarque en su totalidad los datos manejados por la institución. Dado que la experiencia de haber desarrollado un proyecto en conjunto con alumnos de la Facultad de Ingeniería resultó satisfactoria, es que sería una posibilidad factible el poder continuar con el desarrollo del proyecto por parte de otro grupo de alumnos de grado.

Capítulo 9

Conclusiones y trabajo a futuro

En este capítulo se tratarán las conclusiones finales en base al prototipo de extracción de Moodle construido, al diseño logrado y los resultados obtenidos para los objetivos del proyecto, así como las posibles extensiones y trabajos a futuro.

9.1. Conclusiones generales

En cuanto a los objetivos planteados desde un principio para dicho proyecto se puede concluir que se obtuvieron resultados satisfactorios para todos ellos. Como resultado del primer objetivo se encuentra un relevamiento de la realidad de *Ánima* y las necesidades que afronta la institución, destacando entre ellas la importancia que posee el análisis de los datos en los sistemas informáticos para la toma de decisiones en la misma. A partir de los requerimientos obtenidos mediante las reuniones con los referentes de *Ánima* en el transcurso del proyecto, se planteó una solución la cual fue desarrollada a fin de cumplir los mismos. Con respecto al segundo objetivo del proyecto, debido a que no contaban con la capacidad de obtener los datos de Moodle, la cual había aumentado su uso a partir del comienzo de la pandemia, se implementó un prototipo denominado *Data Ánima* que se encarga de la extracción, transformación y carga de los datos de la plataforma. Dado que en la misma se genera una gran variedad de información, se relevó y priorizó aquella que era de mayor relevancia para la institución. A partir de esta lista de prioridad se organizó el orden en el cual se iba a implementar la extracción de los distintos elementos. Finalmente se llegó a implementar la extracción de todos los elementos relevados que estaban disponibles para la versión de Moodle utilizada en *Ánima*. Por otro lado, en cuanto al tercer objetivo del proyecto, debido a que la problemática principal de la institución es la integración de la información con la que trabajan, se planteó y diseñó un *Data Warehouse* a partir de dicha información.

La extracción de los datos por medio de *Data Ánima* fue verificada para un conjunto de datos de prueba, a su vez se implementó un prototipo de *Data Warehouse* acotado a los datos extraídos de Moodle para verificar que su diseño fuera correcto. Por último, se realizó una encuesta para saber la satisfacción de los usuarios que harán uso de los resultados obtenidos en el proyecto, los cuales indicaron estar muy conformes al respecto.

Cabe destacar que con las implementaciones realizadas se puede contemplar el camino completo desde que son extraídos los datos desde Moodle hasta su alojamiento en el prototipo de *Data Warehouse*, para su posterior análisis mediante

reportes creados en Google Data Studio. A su vez, considerando que esta solución soportaría también los datos de los demás sistemas informáticos, se concluye que el trabajo realizado es de gran valor para *Ánima* ya que brinda un análisis detallado del diseño que permite integrar todas sus fuentes de datos y sería de gran utilidad en la eventual implementación de la misma.

9.2. Trabajo a futuro

En base a lo logrado en dicho proyecto se pueden realizar varios trabajos a futuro, los cuales surgen de los requerimientos no conseguidos en esta instancia de trabajo, como también de posibles extensiones que puedan surgir a partir de la solución general planteada a lo largo de este proyecto.

- **Actualizar la versión de Moodle:** Durante la realización del proyecto se encontró una limitante a la hora de realizar los requerimientos de obtención de datos de Moodle ya que en *Ánima* no contaban con la última versión de este. Esta actualización permitiría obtener más información de Moodle a través de los siguientes requerimientos:
 - Retroalimentación de las entregas
 - Entregas en fecha
 - Aprobación de tareas
 - Accesos de los usuarios a Moodle
- **Generación de proceso automático:** la creación de un proceso automático que realice la obtención de la información de Moodle cada cierto tiempo configurable por *Ánima* sería de gran ayuda para la institución para realizar actualizaciones de la información periódicas. Un ejemplo puede ser que se ejecute dicho proceso antes de realizar cada entrega de boletines a los alumnos y que dicho proceso le notifique por correo electrónico a los interesados de la institución.
- **Desarrollar el proceso ETL de todos los sistemas informáticos:** Elaborar dicho proceso facilitaría el manejo de la información que se hace hoy en día de los sistemas informáticos en *Ánima* ya que se manipulan varios archivos excel para unificar toda la información y poder ser visualizada en reportes. Este proceso ETL manipularía todos los archivos excel provenientes de los distintos sistemas informáticos utilizados en *Ánima* y cargar la información resultante en el Data Warehouse diseñado en este proyecto.
- **Utilización de Learning Analytics:** En base a la información obtenida del Data Warehouse se puede incorporar Learning Analytics para obtener variados análisis y alertas sobre distintos criterios que se desee. Dichos análisis complementarían los ya existentes en *Ánima* y las alertas notificarían a docentes, alumnos y administradores de *Ánima*. Un ejemplo sería notificar a los docentes si un alumno no entregó cierta cantidad de tareas a lo largo del curso para que esté enterado de la situación.

Bibliografía

- [1] *Ánima: Quienes somos*. url: <https://anima.edu.uy/quienes-somos/>. (accedido: 13.03.2021).
- [2] *Axios*. url: <https://www.npmjs.com/package/axios>. (accedido: 15.04.2021).
- [3] F. Carpani. «CMDM: un método conceptual para la especificación de bases multidimensionales. Tesis de maestría. Universidad de la República (Uruguay). Facultad de Ingeniería. Instituto de Computación – PEDECIBA.» En: (2000). url: <https://www.colibri.udelar.edu.uy/jspui/handle/20.500.12008/2915>.
- [4] *Configuración Servicios Moodle*. url: https://docs.moodle.org/34/en/Using_web_services. (accedido: 26.04.2021).
- [5] *Date Format*. url: <https://www.npmjs.com/package/dateformat>. (accedido: 15.04.2021).
- [6] *Excel 4 Node*. url: <https://www.javascript.com/>. (accedido: 15.04.2021).
- [7] *Express*. url: <https://expressjs.com/>. (accedido: 15.04.2021).
- [8] *File System*. url: <https://nodejs.org/api/fs.html>. (accedido: 15.04.2021).
- [9] *Flujo de trabajo de Gitflow*. url: <https://www.atlassian.com/es/git/tutorials/comparing-workflows/gitflow-workflow>. (accedido: 01.04.2021).
- [10] *Foro Moodle*. url: <https://docs.moodle.org/all/es/Foros>. (accedido: 11.05.2021).
- [11] M. Heredia M. Rydel M. Saúl y G. Severi. «Extracción y procesamiento de datos para modelado de trayectorias académicas en cursos universitarios. Tesis de grado. Universidad de la República (Uruguay). Facultad de Ingeniería.» En: (2019). url: <https://www.colibri.udelar.edu.uy/jspui/handle/20.500.12008/20932>.
- [12] *Git*. url: <https://git-scm.com/>. (accedido: 01.04.2021).
- [13] *GitHub*. url: <https://github.com/>. (accedido: 14.06.2021).
- [14] *Google Data Studio*. url: <https://marketingplatform.google.com/intl/es/about/data-studio/>. (accedido: 01.04.2021).
- [15] *Google Drive*. url: <https://www.google.com/intl/es/drive/>. (accedido: 01.04.2021).
- [16] *Google Meet*. url: <https://meet.google.com/>. (accedido: 01.04.2021).
- [17] *Grafana*. url: <https://grafana.com/>. (accedido: 22.05.2021).
- [18] *Heroku*. url: <https://www.heroku.com/>. (accedido: 14.06.2021).
- [19] *Heroku CLI*. url: <https://devcenter.heroku.com/articles/heroku-cli>. (accedido: 19.06.2021).

- [20] INOA. url: <https://inoaedu.com/>. (accedido: 13.03.2021).
- [21] Javascript. url: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>. (accedido: 01.04.2021).
- [22] *Learning Analytics*. url: <https://online.ucam.edu/blog/que-es-learning-analytics>. (accedido: 23.03.2021).
- [23] *Material-ui*. url: <https://material-ui.com/>. (accedido: 15.04.2021).
- [24] *Moodle web service API functions*. url: https://docs.moodle.org/dev/Web_service_API_functions. (accedido: 17.04.2021).
- [25] *MoodleCloud*. url: https://moodlecloud.com/app/en/signup/chooseuser?plan=lms_1000. (accedido: 06.05.2021).
- [26] Regina Motz. «DIIA: Descubrimiento de Interacciones que Impactan en el Aprendizaje - Creación de un ambiente de software para descubrir patrones semánticos de interacción». En: *ANII* (2018). url: <http://www.diia.edu.uy/>.
- [27] *Node Fetch*. url: <https://www.npmjs.com/package/node-fetch>. (accedido: 15.04.2021).
- [28] *Node JS*. url: <https://nodejs.org/es/>. (accedido: 01.04.2021).
- [29] Zacharoula Papamitsiou y Anastasios A. Economides. «Learning Analytics for Smart Learning Environments: A Meta-Analysis of Empirical Research Results from 2009 to 2015». En: *Learning, Design, and Technology: An International Compendium of Theory, Research, Practice, and Policy*. Ed. por Michael J Spector, Barbara B Lockee y Marcus D. Childress. Cham: Springer International Publishing, 2016, págs. 1-23. isbn: 978-3-319-17727-4. doi: [10.1007/978-3-319-17727-4_15-1](https://doi.org/10.1007/978-3-319-17727-4_15-1). url: https://doi.org/10.1007/978-3-319-17727-4_15-1.
- [30] *Path*. url: <https://nodejs.org/api/path.html>. (accedido: 15.04.2021).
- [31] *Pentaho*. url: <https://www.hitachivantara.com/en-us/products/data-management-analytics.html?source=pentaho-redirect>. (accedido: 22.05.2021).
- [32] *PostgreSQL*. url: <https://www.postgresql.org/>. (accedido: 10.06.2021).
- [33] *Postman*. url: <https://www.postman.com/>. (accedido: 06.05.2021).
- [34] *Query String*. url: <https://nodejs.org/api/querystring.html>. (accedido: 15.04.2021).
- [35] *RapidMiner*. url: <https://rapidminer.com/>. (accedido: 22.05.2021).
- [36] *React*. url: <https://reactjs.org/>. (accedido: 15.04.2021).
- [37] Shaker H. Ali El-Sappagh, Abdeltawab M. Ahmed Hendawi y Ali Hamed El Bastawissy. «A proposed model for data warehouse ETL processes». En: *Journal of King Saud University - Computer and Information Sciences* 23.2 (2011), págs. 91-104. issn: 1319-1578. doi: <https://doi.org/10.1016/j.jksuci.2011.05.005>. url: <https://www.sciencedirect.com/science/article/pii/S131915781100019X>.
- [38] George Siemens y Phil Long. «Penetrating the Fog: Analytics in Learning and Education». En: *EDUCAUSE Review* 5 (ene. de 2011), págs. 30-32. doi: [10.17471/2499-4324/195](https://doi.org/10.17471/2499-4324/195).

-
- [39] *SIGED - Introducción*. url: <https://sigid.com.uy/web/index.php/productos/sigid/sistema-de-gestiducativa-mainmenu-31>. (accedido: 13.03.2021).
- [40] *Sobre Moodle*. url: https://docs.moodle.org/310/en/About_Moodle. (accedido: 13.03.2021).
- [41] *Trello*. url: <https://trello.com/es>. (accedido: 01.04.2021).
- [42] *Typescript*. url: <https://www.typescriptlang.org/>. (accedido: 01.04.2021).
- [43] Alejandro Vaisman y Esteban Zimányi. *Data Warehouse Systems: Design and Implementation*. Heidelberg: Springer, 2014. isbn: 978-3-642-54654-9. doi: [10.1007/978-3-642-54655-6](https://doi.org/10.1007/978-3-642-54655-6).

Apéndice A

Manual de usuario

En este apéndice se desarrolla el manual de usuario para el sistema construido, a pesar de ser bastante intuitivo y fácil de usar.

A.1. Data Ánima

Para acceder al sitio de Data Ánima se debe acceder desde un navegador a la siguiente URL <http://52.247.119.4:3080/> la cual desplegará la pantalla principal del mismo que se puede ver en la Figura A.1. La URL del sitio no cuenta con un dominio asignado por lo que se debe acceder a través de la IP del servidor, esto no es muy amigable al usuario, pero se soluciona asignando un dominio en la configuración del servidor de Amazon provisto por Ánima.

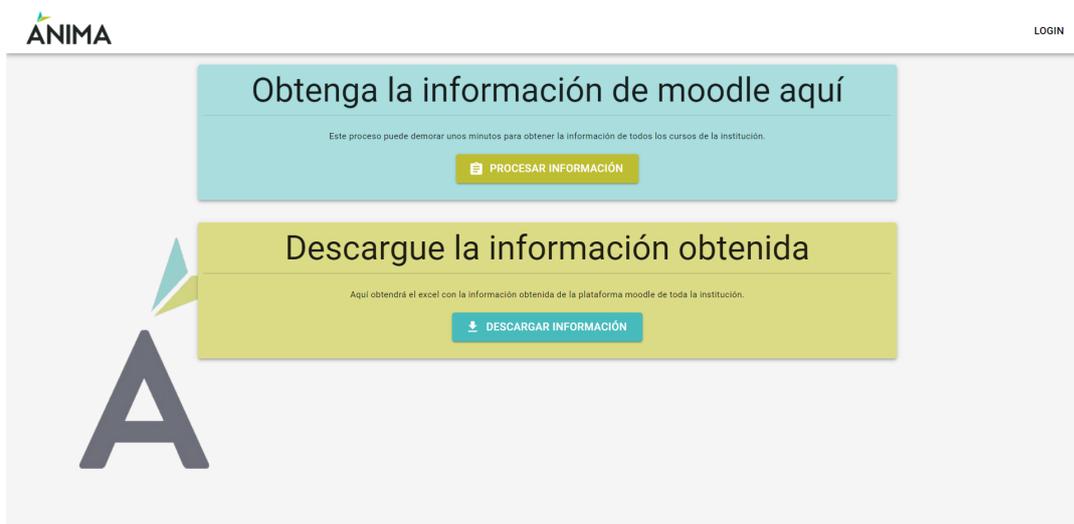


Figura A.1: Data Ánima - Página principal

Una vez habiendo accedido al sitio web se debe proseguir a ingresar en la aplicación como se ve en la Figura A.2.

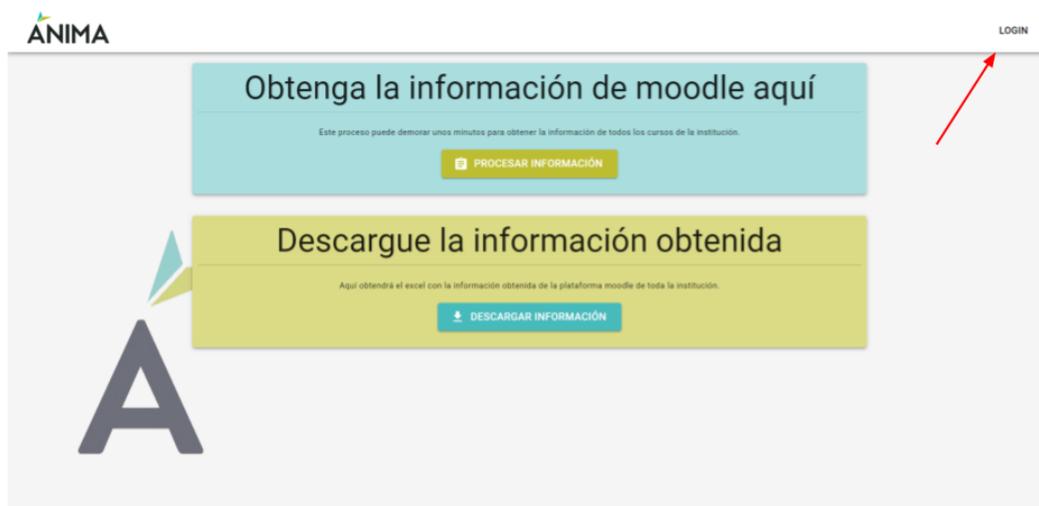


Figura A.2: Data Ánima - Página principal con indicación de Login

Esto desplegará una pantalla donde se deben ingresar los datos del usuario como se ve en la Figura A.3. Aquí se deben ingresar los datos de un usuario de Moodle que cuente con permisos para extraer la información. Estos permisos se pueden conocer más en detalle en el Apéndice B en la Sección B.3.

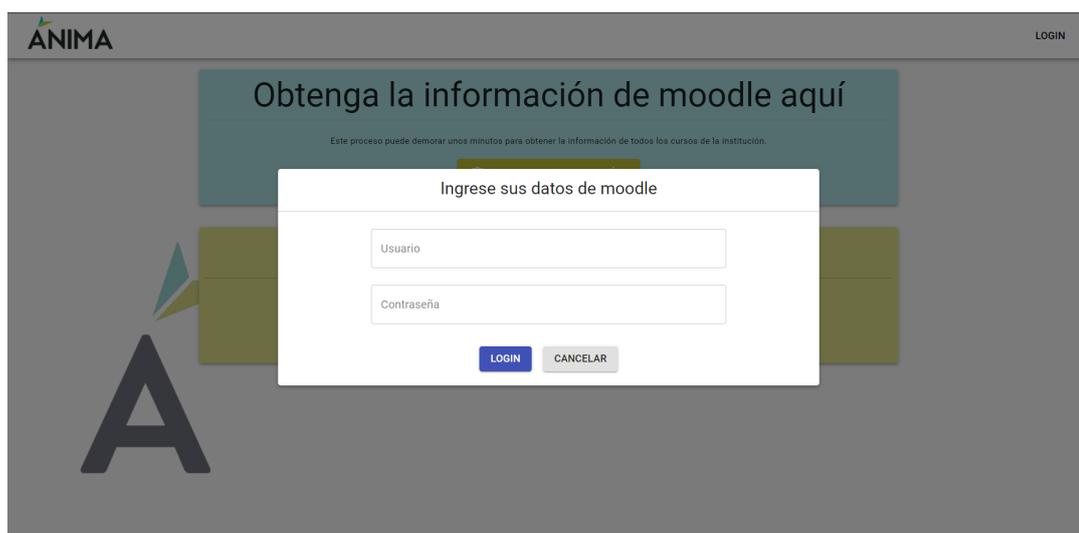
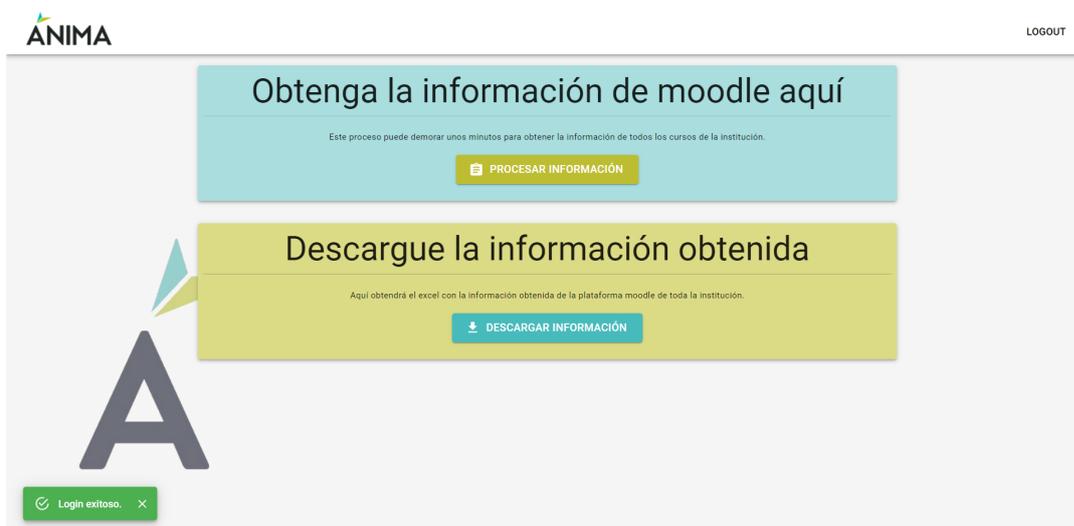
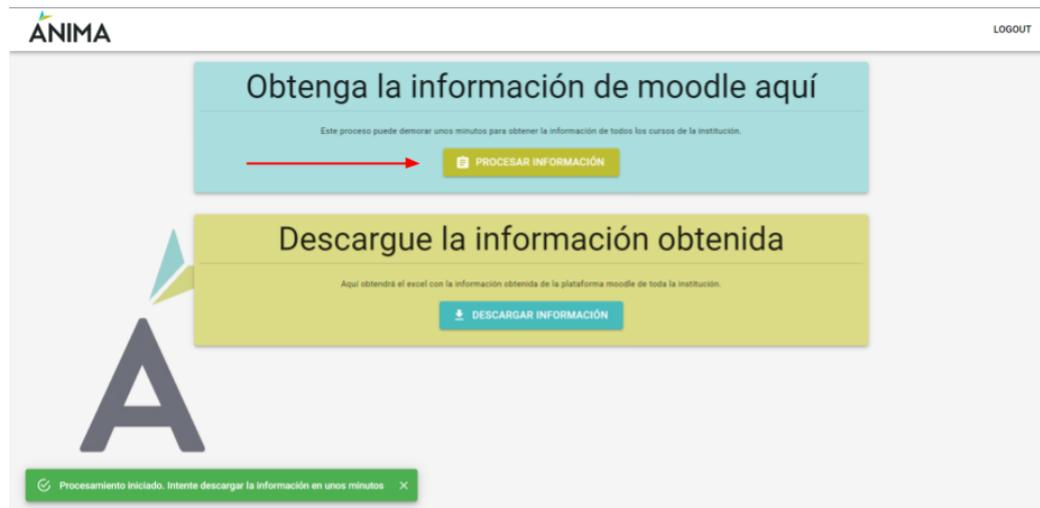


Figura A.3: Data Ánima - Login

Una vez ingresado los datos correctamente la aplicación le advertirá si se logró ingresar correctamente como se ve en la Figura A.4

Figura A.4: Data *Ánima* - Mensaje de Login Exitoso

A continuación, se puede proseguir a iniciar el proceso de obtención de los datos de Moodle presionando en el botón "OBTENER INFORMACIÓN" como se marca en el Figura A.5. Además, también se puede observar que la aplicación alerta si el proceso se inició correctamente.

Figura A.5: Data *Ánima* - Mensaje de procesamiento exitoso

Luego de transcurrido unos minutos en el que el proceso de extracción haya finalizado se puede proceder a descargar el archivo generado por la aplicación presionando en el botón "DESCARGAR INFORMACIÓN" como se ve en la Figura A.6. También se puede ver que el sistema alerta si la descarga fue exitosa.

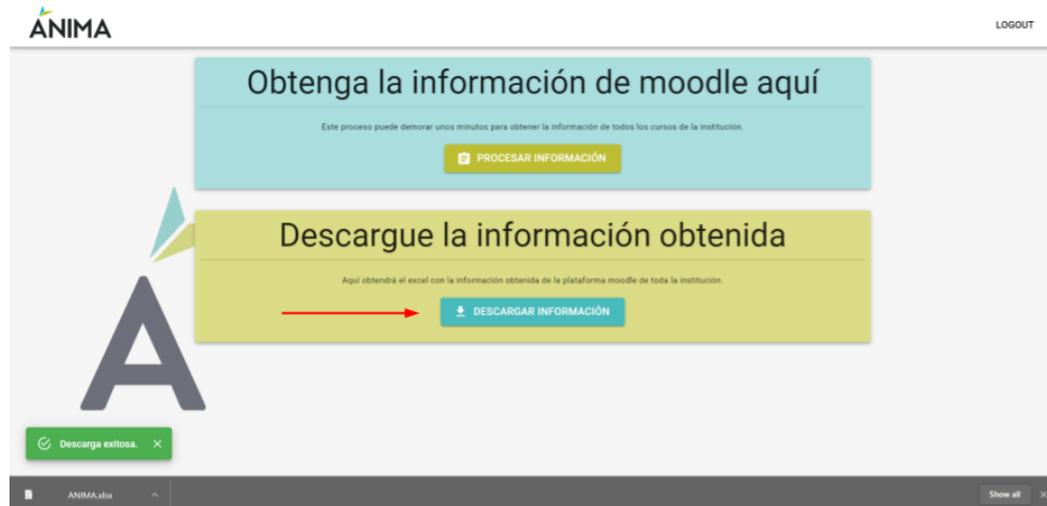


Figura A.6: Data Ánima - Mensaje de descarga exitosa

En este punto ya se cuenta con la última información extraída desde Moodle con la cual ya se puede comenzar a utilizar para generar los cuadros de mando deseados.

A.2. Consideraciones para tener en cuenta

Una consideración para tener en cuenta es que el sitio le advertirá que debe ingresar previamente antes de utilizarla. Por ejemplo, si se quiere obtener la información de Moodle o descargar la información le advertirá a través de un mensaje como se puede ver en la Figura A.7

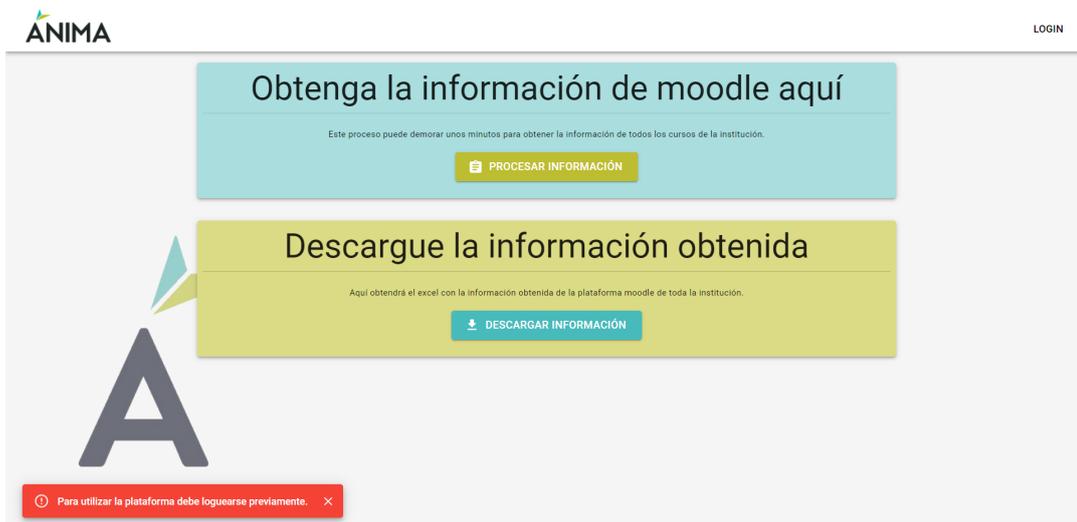


Figura A.7: Data Ánima - Mensaje de error sobre ingresar nuevamente

Otra consideración que se debe tener en cuenta a la hora de utilizar el sistema es que si se presiona el botón de “DESCARGAR INFORMACIÓN” previo a presionar el botón de “PROCESAR INFORMACIÓN” se descargará un excel con la información obtenida la última vez que se haya procesado la información de

Moodle. Esto quiere decir que puede que no esté actualizada y falten algunos datos.

Por otro lado, otra cosa para tener en cuenta es que cuando se presiona el botón de “PROCESAR INFORMACIÓN” y si se presiona luego el botón de “DESCARGAR INFORMACIÓN” antes de que el proceso de obtención de información de Moodle haya terminado el sitio web le desplegará un aviso informando que el proceso no ha finalizado y que vuelva a intentar descargar la información más tarde, como se puede ver en el Figura A.8.

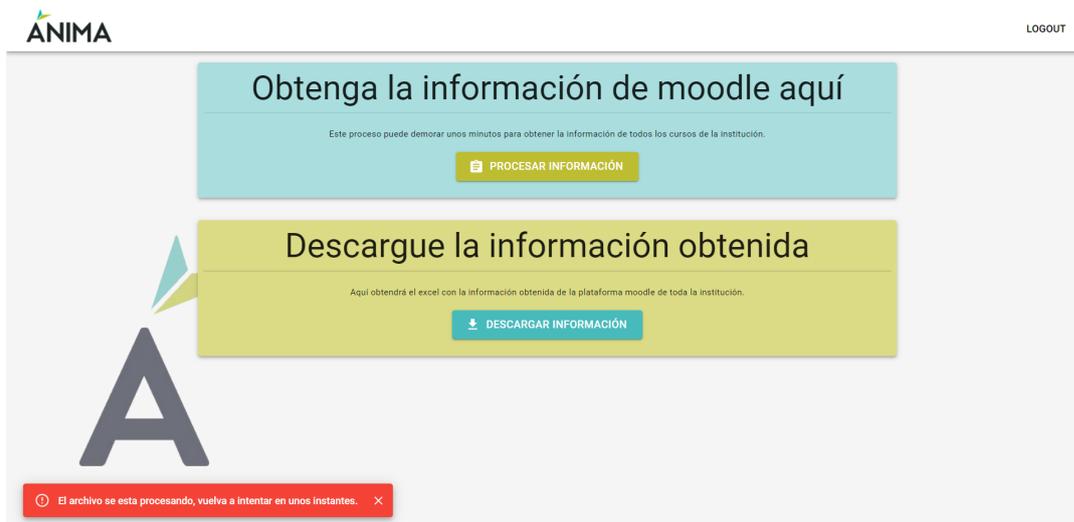


Figura A.8: Data Ánima - Mensaje de error sobre procesamiento en progreso

Por último, en caso de que se haya cambiado la configuración de permisos de un usuario en Moodle en los Tokens que tiene habilitados, el prototipo informará al usuario que el Token es inválido y debe ingresar nuevamente las credenciales al sitio como se puede ver en la Figura A.9.

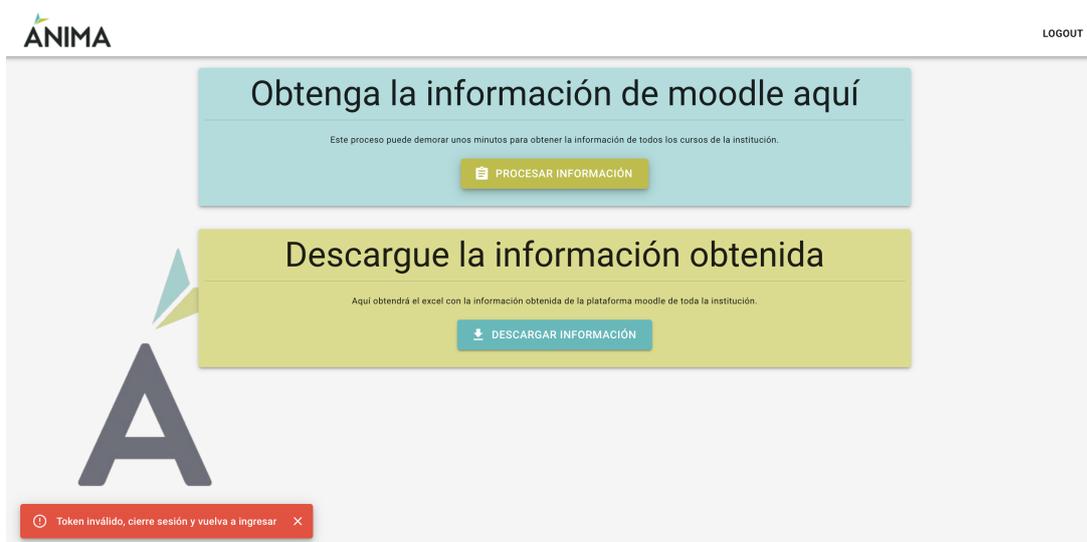


Figura A.9: Data Ánima - Mensaje de error sobre Token inválido

Apéndice B

Manual de instalación de Data Ánima

En este manual de instalación se detallan los pasos a seguir para instalar y configurar el prototipo de extracción de Moodle en su totalidad. El manual se divide en tres secciones. La primera en la que se menciona cómo conectarse al servidor y configurar las distintas partes de la solución. En la segunda sección se detalla cómo habilitar los servicios web de Moodle a través de la API que utiliza el prototipo desarrollado. Por último, se detalla cómo habilitar los permisos para los usuarios de Moodle para que puedan hacer uso de la API.

B.1. Configuración del servidor

El servidor con el que se trabajó durante el proyecto es una instancia EC2 de Amazon con sistema operativo Ubuntu, el cual fue brindado por Ánima para hacer las pruebas y dejar allí instalado el prototipo desarrollado.

B.1.1. Conexión al servidor e instalación de Data Ánima

Para realizar la conexión con el servidor se debe contar con el archivo “DataAnima_key.pem” el cual funciona como clave para acceder al servidor. Para lograr conectarse al servidor se deben ejecutar los siguientes comandos en una consola ubicada en la carpeta donde se encuentre el archivo “.pem”:

```
$ chmod 400 DataAnima_key.pem  
$ ssh -i DataAnima_key.pem ubuntu@52.247.119.4
```

El primer comando que se detalla se utiliza para quitarle los permisos al usuario sobre el archivo “.pem” y solamente permitirle leer el mismo, esto es necesario para realizar la conexión SSH al servidor de Amazon. Si la conexión se realizó correctamente se debería observar en la consola algo similar a lo que se muestra en la Figura B.1.

```

antony.maciel@UY-IT01237 Desktop % ssh -i DataAnima_key.pem ubuntu@52.247.119.4
Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 5.4.0-1026-azure x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Tue Apr 20 21:56:03 UTC 2021

System load:  0.08          Processes:            122
Usage of /:   15.4% of 28.90GB Users logged in:      0
Memory usage: 46%          IP address for eth0: 10.0.0.4
Swap usage:   0%

 * Pure upstream Kubernetes 1.21, smallest, simplest cluster ops!

https://microk8s.io/

 * Canonical Livepatch is available for installation.
- Reduce system reboots and improve kernel security. Activate at:
https://ubuntu.com/livepatch

63 packages can be updated.
0 updates are security updates.

*** System restart required ***
Last login: Thu Mar 25 02:48:08 2021 from 179.25.38.160
ubuntu@DataAnima:~$

```

Figura B.1: Conexión al servidor

Una vez corroborada la conexión al servidor se debe proseguir a desconectar-se del mismo y transferir el archivo comprimido “anima-master.zip” que contiene el código fuente del prototipo de extracción de Moodle. Para realizar esto se debe tener el archivo comprimido en la misma carpeta que el archivo “.pem”. Para esto se debe ejecutar el siguiente comando:

```

$ exit
$ scp -i DataAnima_key.pem anima-master.zip ubuntu@52.247.119.4:/tmp

```

Una vez transferido el archivo con el código fuente del prototipo de extracción de Moodle se debe realizar la conexión nuevamente al servidor y se deben instalar algunas herramientas necesarias para ejecutar el proyecto. Se debe instalar NodeJs para compilar y ejecutar el proyecto, a su vez se debe instalar npm que es el manejador de paquetes de NodeJs y por último la librería PM2 que se encarga de manejar los procesos en el servidor. Dichas herramientas se instalan con los siguientes comandos:

```

$ ssh -i DataAnima_key.pem ubuntu@52.247.119.4
$ curl -sL https://deb.nodesource.com/setup_14.x | sudo -E bash -
$ sudo apt-get install -y nodejs
$ npm install pm2 -g
$ sudo apt install unzip

```

Teniendo todas las herramientas necesarias instaladas para ejecutar el prototipo, se puede proceder a compilar el mismo. Para esto se deben ejecutar los siguientes comandos:

```

$ cd ../../tmp
$ mv anima-master.zip ../home/ubuntu
$ cd ../home/ubuntu
$ unzip anima-master.zip
$ cd anima-master/front
$ npm install
$ npm run build

```

```
$ cd ../api
$ npm install
$ npm run build
```

Una vez compilado el prototipo se puede proceder a ponerlo en marcha con el manejador de procesos anteriormente instalado, con los siguientes comandos:

```
$ pm2 stop 0
$ pm2 delete 0
$ pm2 start dist/app.js
```

Hecho esto se puede ingresar en <http://52.247.119.4:3080/> que es la url en la que queda público para su acceso.

B.1.2. Back end

Para la configuración del Back end de Data Ánima se debe tener en cuenta que se puede modificar la url de la plataforma Moodle de la cual obtener la información. Esta configuración se debe realizar en el archivo ubicado en “anima-master/api/src/env.ts”, modificando la propiedad:

```
postTokenURL = 'url_moodle_anima';
```

Siendo **url_moodle_anima** la url de la plataforma de Moodle de Ánima para extraer la información del mismo. De realizarse algún tipo de cambio de configuración se deben seguir los pasos descritos en la Subsección B.1 para reflejar los cambios realizados en el servidor.

B.1.3. Front end

Dado que en Data Ánima se utilizan los usuarios y contraseñas de la plataforma Moodle de Ánima para autenticarse y poder hacer uso del mismo, es que se puede configurar la url del sitio Moodle. Esta url debe coincidir con la que se haya configurado para el Back end mencionada en la Sección B.1.2.

Para esto se debe modificar el archivo ubicado en “anima-master/front/src/api/urls.js”, dentro de éste se encuentra la propiedad:

```
postTokenURL = 'url_moodle_anima/login/token.php';
```

Donde **url_moodle_anima** es la url de la plataforma de Moodle de Ánima. Esto hace que sea una solución genérica y que si en algún momento se modificara el acceso al Moodle de Ánima se pueda modificar de manera sencilla. Si se realiza un cambio de configuración se deben seguir los pasos detallados en la Subsección B.1 para que el servidor se actualice y tome los cambios realizados.

B.2. Configuración de servicios en Moodle

Para la configuración de los servicios de Moodle es necesario habilitar y configurar servicios dentro de la plataforma, ya que por defecto estos vienen deshabilitados para mejorar la seguridad del sistema. A continuación, se detalla cada una de las configuraciones necesarias, cabe destacar que para realizar todos los pasos descritos a continuación hay que estar ingresado en el sistema con un usuario con permisos de administrador. [4]

B.2.1. Habilitar servicios web

El primer paso es habilitar los servicios web, para esto se tiene que realizar los siguientes pasos:

- Administración >Administración del sitio >Características avanzadas.
- Seleccionar “Habilitar servicios web” y después el botón “Guardar cambios”.

En la Figura B.2 se ilustran los pasos mencionados anteriormente.

The screenshot shows the Moodle administration interface. On the left, there is a navigation menu with 'ADMINISTRACIÓN' expanded to 'Administración del sitio', where 'Características avanzadas' is selected. The main content area is titled 'Características avanzadas' and contains several settings:

- Habilitar resultados** (enableoutcomes): Valor por defecto: No. Apoyo para los Resultados (también conocidos como 'competencias', 'objetivos', 'estándares' o 'criterios'). Significa que calificamos usando una o más escalas que están conectadas a las declaraciones de resultados. La activación de los resultados posibilita esta forma especial de calificación en todo el sitio.
- Habilitar comentarios** (usecomments): Valor por defecto: Sí. Habilitar comentarios.
- Activar funcionalidad de las marcas** (usetags): Valor por defecto: Sí. ¿Habilitar la funcionalidad de marcas a través del sitio?
- Habilitar notas** (enablenotes): Valor por defecto: Sí. Habilitar almacenamiento de notas sobre usuarios individuales.
- Portafolios disponibles** (enableportfolios): Valor por defecto: No. Esta opción permite a los administradores configurar sistemas remotos para que los usuarios exporten contenidos a ellos.
- Habilitar servicios web** (enablewebservices): Valor por defecto: No. Los servicios Web permiten otros sistemas para acceder a este Moodle y realizar operaciones. Para mayor seguridad esta característica debe ser desactivada a menos que realmente la esté utilizando.

Figura B.2: Habilitar servicios web

B.2.2. Habilitar protocolos

En este paso se habilitan los protocolos que van a ser utilizados para consumir los servicios web, para esto se tiene que realizar los siguientes pasos:

- Administración >Administración del sitio >Extensiones >Servicios web >Administrar protocolos.
- Seleccionar el protocolo que se quiere habilitar (protocolo REST como se ve en la Figura B.3) en la columna “Habilitar” y después el botón “Guardar cambios”.

Administrar protocolos

Protocolos activos de servicio web

Protocolo	Versión	Habilitar	Configuración
Protocolo AMF	2015051100		
Protocolo REST	2015051100		
Protocolo SOAP	2015051100		
Protocolo XML-RPC	2015051100		

Por razones de seguridad, solo se habilitarán los protocolos que vayan a ser utilizados.

Documentación de servicios web Valor por defecto: No

enablewsdocumentation

Habilitar la auto-generación de documentación de servicios web. Un usuario puede tener acceso a su propia documentación en su página de claves de seguridad [Más detalles](#). Solo se muestra la documentación de los protocolos habilitados.

Guardar cambios

Figura B.3: Habilitar protocolos

B.2.3. Crear un servicio

Para la creación de un servicio que incluya las funciones utilizadas por Data Ánima se tiene que realizar los siguientes pasos:

- Administración > Administración del sitio > Extensiones > Servicios web > Servicios externos.
- Seleccionar Agregar
 - Escribir un nombre (“custom_web_services” como se puede ver en la Figura. B.4).
 - Seleccionar la casilla “Habilitado”.
 - Seleccionar el botón “Guardar cambios”.

Servicios externos

Información

Un servicio es un conjunto de funciones. A dicho servicio pueden acceder todos los usuarios o sólo algunos especificados.

Servicios incluidos

Servicio externo	Extensión	Funciones	Usuarios	Editar
Moodle mobile web service	moodle	Funciones	Todos los usuarios	Editar

Servicios personalizados

Servicio externo	Borrar	Funciones	Usuarios	Editar
custom_web_services	Borrar	Funciones	Todos los usuarios	Editar

[Agregar](#)

Figura B.4: Crear servicio

B.2.4. Agregar funciones a un servicio

Luego de creado un servicio es necesario agregar las funciones que se quieran utilizar en este, para agregar un servicio se tiene que realizar los siguientes pasos:

- Seleccionar el enlace “funciones” en la fila del servicio al que se quiere agregar una función.
- Seleccionar “Agregar funciones”.
- Seleccionar las funciones que se quieran agregar (como se puede ver en la Figura B.5) y luego el botón “Agregar funciones”.

Agregar funciones al servicio "custom_web_services"

Función	Descripción	Permisos requeridos	Editar
core_user_get_users_by_field	Retrieve users information for a specified unique field - If you want to do a user search, use core_user_get_users()	moodle/user:viewdetails, moodle/user:viewhiddendetails, moodle/course:useremail, moodle/user:update	Eliminar
core_course_get_courses	Return course details	moodle/course:view,moodle/course:update,moodle/course:viewhiddencourses	Eliminar
core_course_get_contents	Get course contents	moodle/course:update,moodle/course:viewhiddencourses	Eliminar
mod_assign_get_assignments	Returns the courses and assignments for the users capability		Eliminar
mod_assign_get_submissions	Returns the submissions for assignments		Eliminar
core_enrol_get_enrolled_users	Get enrolled users by course id.	moodle/user:viewdetails, moodle/user:viewhiddendetails, moodle/course:useremail, moodle/user:update, moodle/site:accessallgroups	Eliminar
mod_forum_get_forums_by_courses	Returns a list of forum instances in a provided set of courses, if no courses are provided then all the forum instances the user has access to will be returned.	mod/forum:viewdiscussion	Eliminar
mod_forum_get_forum_discussions_paginated	Returns a list of forum discussions optionally sorted and paginated.	mod/forum:viewdiscussion, mod/forum:viewqandawithoutposting	Eliminar
mod_forum_get_forum_discussion_posts	Returns a list of forum posts for a discussion.	mod/forum:viewdiscussion, mod/forum:viewqandawithoutposting	Eliminar
mod_assign_get_grades	Returns grades from the assignment		Eliminar

[Agregar funciones](#)

Figura B.5: Agregar funciones

B.3. Configuración de permisos de usuarios en Moodle

Para que un usuario pueda acceder a los servicios deben de tener los permisos de administrador, para poder asignar estos permisos a un usuario se tiene que realizar los siguientes pasos:

- Administración >Administración del sitio >Usuarios >Permisos >Administradores del sitio.
- Seleccionar en el buscador derecho, el usuario que se quiere agregar.
- Seleccionar el botón “Agregar”.

Además, se debe agregar un Token para el usuario, el cual permitirá tener acceso al servicio “custom_web_services” creado en los pasos anteriores. Esto se puede realizar con los siguientes pasos:

- Administración del sitio >Extensiones >Servicios web >Administrar tokens.
- Seleccionar el botón “Agregar”.
- Ingresar el nombre del usuario (nombre de usuario Moodle).
- Seleccionar el servicio web “custom_web_services” (como se puede ver en la Figura B.6).
- Seleccionar el botón “Guardar cambios”.

Crear ficha

▼ Ficha

Nombre de usuario / ID de Usuario*

Servicio*

Restricción de IP

Válido hast Habilitar

En este formulario hay campos obligatorios *.

Figura B.6: Agregar token

Apéndice C

Manuales del prototipo de Data Warehouse

En este manual se darán los detalles para instalar el prototipo de Data Warehouse implementado. Cabe destacar que este prototipo posee la misma estructura que el prototipo de Data Ánima por lo que muchas configuraciones son similares. A su vez se brinda un manual de usuario para la utilización del prototipo.

C.1. Instalación y configuración de la base de datos

A fin de instalar la base de datos se debe realizar la conexión al servidor, en este caso el servidor utilizado es la misma instancia EC2 de Amazon con sistema operativo Ubuntu que es utilizada para la instalación de Data Ánima.

Esto se realiza a través del siguiente comando en la consola estando en el directorio que contiene el archivo “.pem” para la conexión al servidor.

```
$ ssh -i DataAnima_key.pem ubuntu@52.247.119.4
```

Luego de realizada la conexión al servidor se debe proceder a la instalación de una base de datos, en este caso se utiliza una base de datos PostgreSQL [32]. Se debe ingresar los siguientes comandos en la consola a fin de instalar dicha base de datos:

```
$ sudo apt update
$ sudo apt install postgresql postgresql-contrib
$ sudo service postgresql start
```

Una vez instalada la base de datos, se debe configurar para que pueda ser accedida remotamente fuera del servidor, para ello se deben seguir los pasos a continuación. Primero, ingresar los siguientes comandos en la consola a fin de crear un nuevo usuario de la base de datos:

```
$ sudo su postgres
$ psql -U postgres -c "CREATE ROLE ubuntu;"
$ psql -U postgres -c "ALTER ROLE ubuntu WITH LOGIN;"
$ psql -U postgres -c "ALTER USER ubuntu CREATEDB;"
$ psql -U postgres -c "ALTER USER ubuntu WITH PASSWORD '<<pass>>';"
```

Siendo «**pass**» la contraseña elegida para el usuario creado, debiéndose reemplazar por la que se va a utilizar.

En segundo lugar, se debe modificar algunas líneas de los siguientes archivos de configuración de la base de datos para que pueda ser accedida remotamente. En el archivo “postgresql.conf” se debe modificar la siguiente configuración:

- A fin de acceder al archivo ingresar el siguiente comando en la consola:

```
$ sudo nano /etc/postgresql/10/main/postgresql.conf
```

- Dentro del archivo debe realizarse el siguiente cambio:

```
listen_addresses = '*'
```

En el archivo “pg_hba.conf” se deben realizar lo siguiente:

- Acceder al archivo ingresando el siguiente comando en la consola:

```
$ sudo nano /etc/postgresql/10/main/pg_hba.conf
```

- Dentro del archivo se deben modificar las siguientes secciones:

- En la sección ipv4 realizar el cambio:

```
host    all             all             0.0.0.0/0      md5
```

- En la sección ipv6 realizar el cambio:

```
host    all             all             :::/0          md5
```

C.2. Ejecución del Script en la base de datos

Luego de instalada la base de datos se procede a ejecutar el Script que contiene las tablas propuestas en el diseño del Data Warehouse realizado. Para ello se debe enviar al servidor el archivo que contiene el Script de la base de datos, al cual llamaremos “DataAnima.sql”. Para ello se ingresa a la consola dentro del directorio donde se encuentra el archivo de conexión al servidor “.pem” y el archivo con el Script de la base de datos, y se ingresa el siguiente comando:

```
$ scp -i DataAnima_key.pem DataAnima.sql ubuntu@52.247.119.4:/tmp
```

Una vez enviado el Script de la base de datos al servidor, se deben ejecutar los siguientes comandos a fin de conectarse al servidor, crear una nueva base de datos y ejecutar el Script:

```
$ ssh -i DataAnima_key.pem ubuntu@52.247.119.4
$ cd ../../tmp
$ mv DataAnima.sql ../home/ubuntu
$ cd ../home/ubuntu
$ createdb dataAnima
$ psql dataAnima < DataAnima.sql
```

C.3. Instalación del prototipo de Data Warehouse

Para instalar el prototipo se deben seguir pasos similares a la instalación de Data Anima con las siguientes salvedades.

Por un lado, se debe usar el archivo “anima-data-warehouse.zip” que contiene la implementación correspondiente al prototipo del Data Warehouse. A su vez los pasos no deben realizarse dentro del mismo servidor brindado por Anima en este

proyecto, esto debido a que no posee la suficiente memoria RAM. Por esta razón, se debe instalar en otro servidor.

Por otro lado, antes de proceder con la instalación se debe agregar a la implementación del prototipo las credenciales de la base de datos creada anteriormente.

A fin de agregar las credenciales de la base de datos en el prototipo se debe descomprimir el archivo “.zip” y acceder a la ruta “anima-data-warehouse/api/src”, donde se encuentra el archivo “env.ts” que contiene la configuración de la base de datos. Se debe editar el archivo con la siguiente información:

```
DB_USER: "ubuntu",
DB_HOST: "52.247.119.4",
DB_NAME: "dataAnima",
DB_PASSWORD: "<<pass>>",
DB_PORT: "5432",
```

Siendo «**pass**» la contraseña elegida para el usuario creado en la base de datos cuando se configuró la misma.

Finalmente, para realizar la instalación, se debe modificar el archivo que contiene la URL del servidor del prototipo a fin de que interactúe con el sitio web correctamente. Para ello se debe acceder a la ruta “anima-data-warehouse/front/api” y modificar el archivo “urls.js” para que contenga la siguiente línea:

```
const serverUrl = '<<url>>';
```

En donde «**url**» será el dominio del servidor en el que se instaló, debiéndose usar el puerto 3080 (en caso de que no se posea un dominio propio se puede usar la URL <http://localhost:3080>).

A continuación, se presentan algunas alternativas de instalación dependiendo del ambiente de ejecución a utilizarse.

C.3.1. Instalación del prototipo para pruebas y desarrollo

Para realizar la instalación del prototipo en un ambiente de desarrollo se debe abrir dos consolas en el directorio donde se encuentra la carpeta “anima-data-warehouse”, esto es para iniciar el sitio web y el servidor respectivamente. En una consola ingresar los siguientes comandos a fin de iniciar el sitio web:

```
$ cd anima-data-warehouse/front
$ npm install
$ npm run build
$ npm start
```

En la otra consola ingresar los siguientes comandos a fin de iniciar el servidor:

```
$ cd anima-data-warehouse/api
$ npm install
$ npm run build
$ npm start
```

Luego se podrá acceder al sitio web desde un navegador con el dominio del servidor en el puerto 5000 (sino se posee un dominio propio puede acceder a la URL <http://localhost:5000/>).

C.3.2. Instalación del prototipo en Heroku

Debido a que en este proyecto se utilizó un servidor Heroku [18], se detallan los pasos para realizar la instalación en un servidor de esta plataforma, el cual es gratuito y puede ser accedido remotamente.

Para ello se debe utilizar el usuario provisto a *Ánima* con permisos de acceso a la instancia Heroku [18] utilizada, a su vez se debe crear un usuario en la plataforma Github [13]. También debe instalarse la herramienta de Heroku CLI [19] a fin de realizar la conexión a la plataforma.

Luego estando dentro del directorio donde se encuentra la carpeta “*anima-data-warehouse*” ingresar los siguientes comandos en la consola:

```
$ heroku login
$ cd anima-data-warehouse/api
$ git init
$ git remote add heroku https://git.heroku.com/data-anima.git
$ git add .
$ git commit -m "deploy app"
$ git push heroku master -f
$ cd ../front
$ git init
$ git remote add heroku https://git.heroku.com/data-anima-web.git
$ git add .
$ git commit -m "deploy app"
$ git push heroku master -f
```

Luego la aplicación podrá ser accedida en la URL <http://data-anima-web.herokuapp.com>.

C.4. Manual de usuario

En esta sección se detalla un manual de usuario para el prototipo del Data Warehouse. A fin de acceder a la aplicación se debe abrir un navegador e ingresar a la URL <http://data-anima-web.herokuapp.com>. En esta página podrá visualizar la interfaz de usuario como se ve en la Figura C.1, la cual requerirá de una autenticación previa para su utilización.

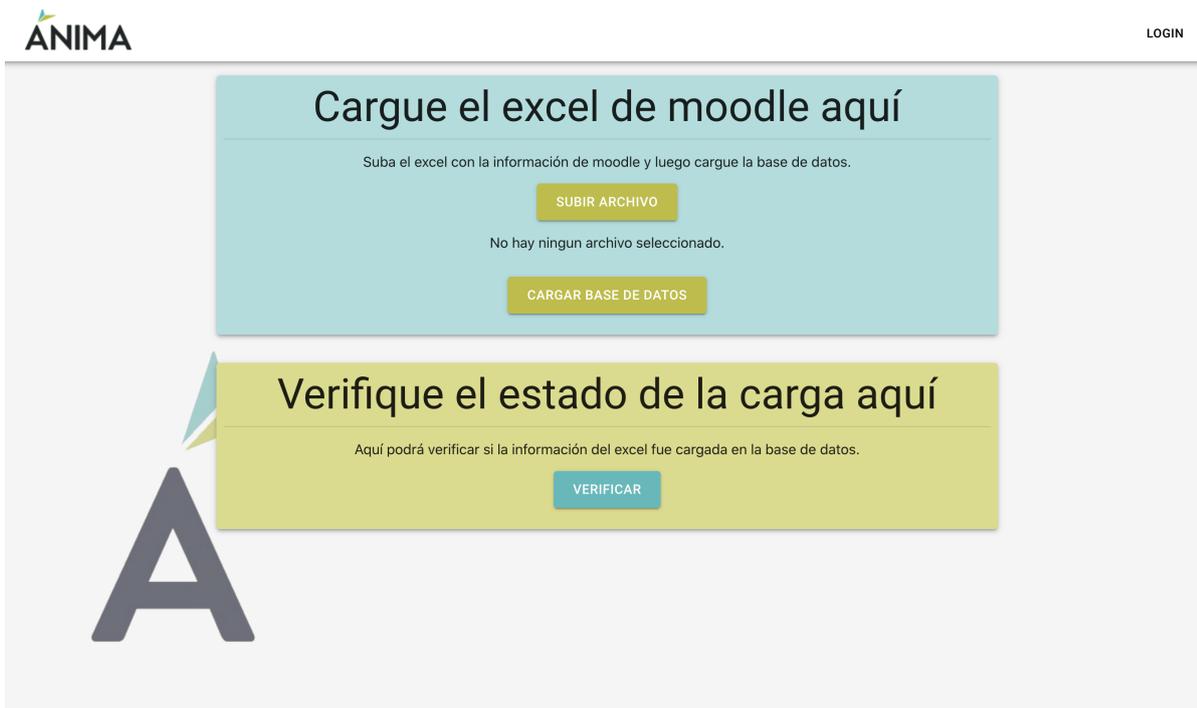


Figura C.1: Página principal prototipo Data Warehouse

Para autenticarse se debe ingresar las credenciales de Moodle de Ánima, donde el usuario deberá tener los mismos permisos que se requieren para la utilización del prototipo de extracción de Moodle.

Luego se debe cargar el archivo excel obtenido en Data Ánima presionando el botón “SUBIR ARCHIVO” y seleccionando el mismo como se ve a continuación en la Figura C.2.

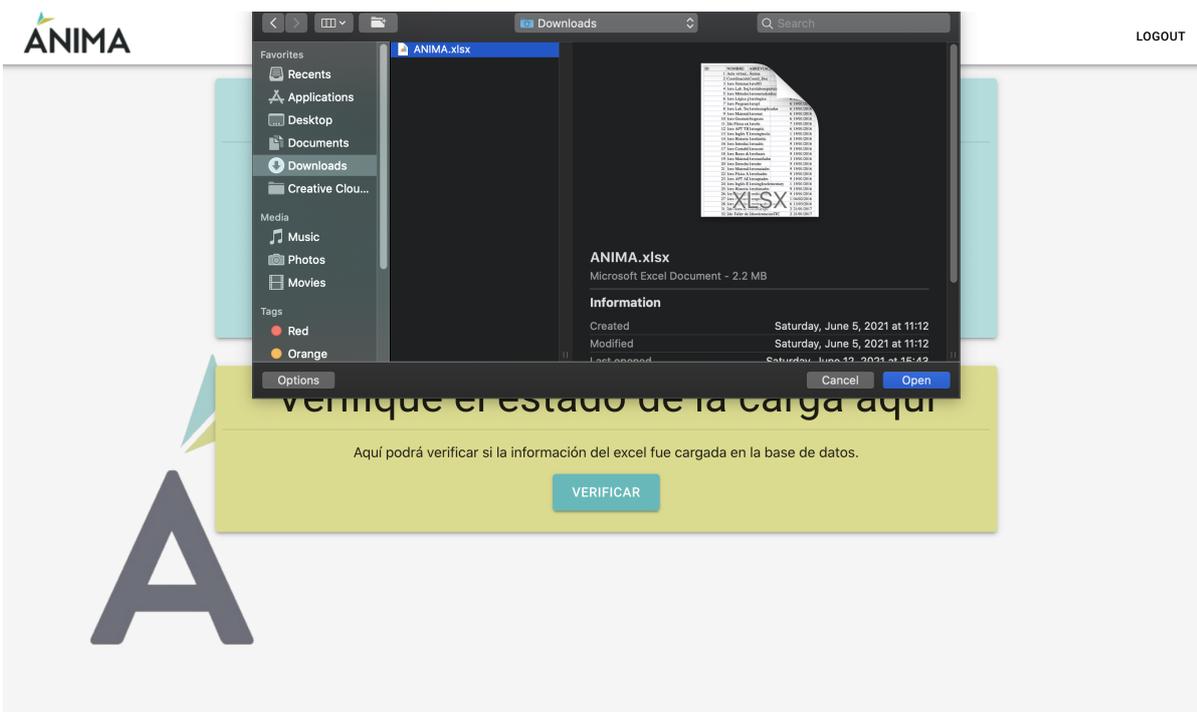


Figura C.2: Prototipo Data Warehouse - Subir Archivo

Una vez cargado el archivo se debe presionar el botón “CARGAR BASE DE DATOS” para iniciar la carga de la base de datos del Data Warehouse. Se puede verificar el progreso de la carga con el botón “VERIFICAR”, en caso de que no haya finalizado se muestra el mensaje de la Figura C.3 y en caso de que haya finalizado se muestra el mensaje de la Figura C.4.

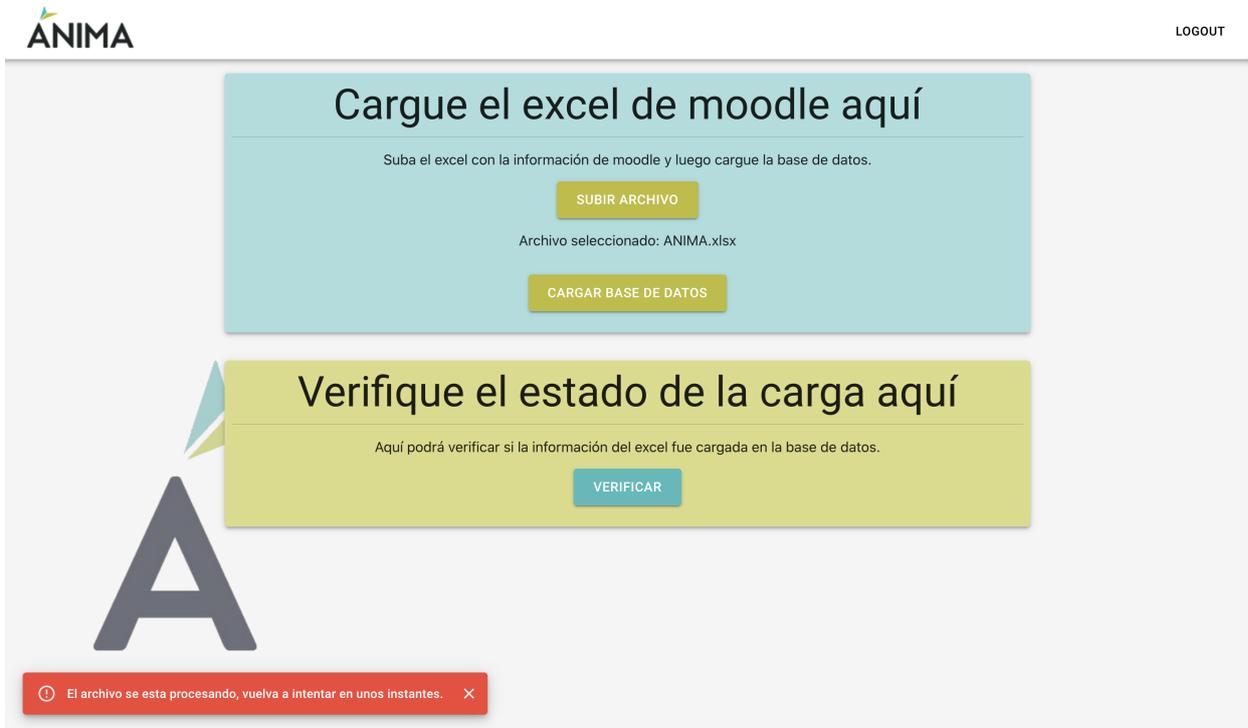


Figura C.3: Prototipo Data Warehouse - Carga no finalizada

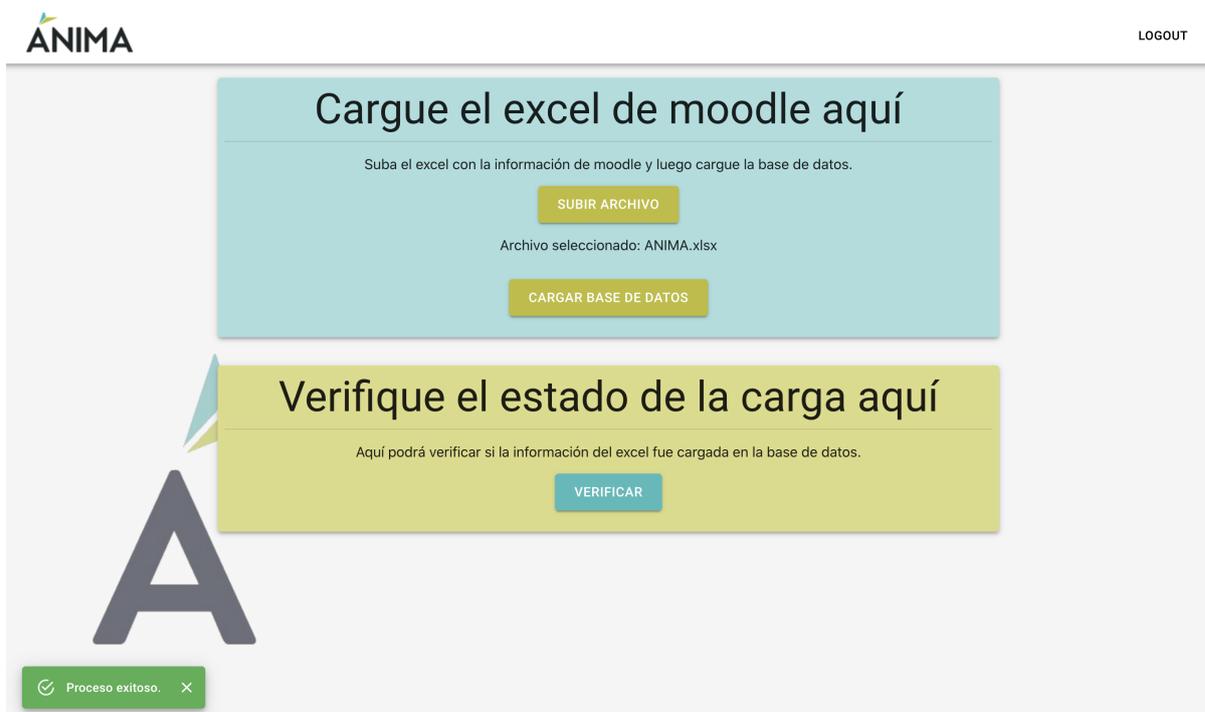


Figura C.4: Prototipo Data Warehouse - Carga finalizada

Apéndice D

Servicios web de Moodle

En este apéndice se va a describir en detalle la forma de ejecutar las funciones hacia la API de Moodle y un ejemplo de la respuesta de cada una de estas. Cierta información fue eliminada para respetar la privacidad de la organización.

D.1. Obtención del Token

D.1.1. Envío

```
1 http://aulavirtual.anima.edu.uy/login/token.php
2   ? username = <usuario_anima>
3   & password = <contraseña_usuario_anima>
4   & service = custom_ws
```

D.1.2. Respuesta

```
1 {
2   "token": "<anima_token>"
3 }
```

D.1.3. Información utilizada

- **token:** identificador de autenticación provisto al usuario para que a través del mismo acceda a los servicios web de la plataforma.

D.2. Obtención de información de un usuario

D.2.1. Consulta

Envío

```
1 http://aulavirtual.anima.edu.uy/webservice/rest/server.php
2   ? wstoken = <anima_token>
3   & wsfunction = core_user_get_users_by_field
4   & moodlesrestformat = json
5   & field = <atributo>
6   & values[<indice>] = <atributo_usuario>
```

Respuesta

```
1  [
2    {
3      "id": 474,
4      "username": "<nombre_usuario_anima>",
5      "firstname": "<nombre_usuario>",
6      "lastname": "apellido_usuario",
7      "fullname": "<nombre_apellido_usuario>",
8      "email": "<email_usuario>",
9      "department": "<departamento_usuario>",
10     "firstaccess": <primer_acceso>,
11     "lastaccess": <ultimo_acceso>,
12     "description": "<descripcion_usuario>",
13     "descriptionformat": <formato_descripcion>,
14     "profileimageurlsmall": "<imagen_usuario_chica>",
15     "profileimageurl": "<imagen_usuario_original>",
16     "preferences": [
17       {
18         "name": "assign_filter",
19         "value": ""
20       },
21       {
22         "name": "assign_perpage",
23         "value": "50"
24       },
25       {
26         "name": "assign_quickgrading",
27         "value": ""
28       },
29       {
30         "name": "auth_forcepasswordchange",
31         "value": "0"
32       },
33       {
34         "name": "auth_manual_passwordupdatetime",
35         "value": "1600871734"
36       },
37       {
38         "name": "email_bounce_count",
39         "value": "1"
40       },
41       {
42         "name": "email_send_count",
43         "value": "1"
44       },
45       {
46         "name": "flextable_mod_assign_grading",
```

```

47         "value": "{\\"collapse\\":[],\\"sortby\\":{\\"userid\\":4}
           ,\\"i_first\\":\\"\\",\\"i_last\\":\\"\\",\\"textsort\\":
           []}"
48     },
49     {
50         "name": "grade_report_showonlyactiveenrol",
51         "value": "1"
52     },
53     {
54         "name": "login_failed_count_since_success",
55         "value": "15"
56     },
57     {
58         "name": "userselector_autoselectunique",
59         "value": "0"
60     },
61     {
62         "name": "userselector_preserveselected",
63         "value": "0"
64     },
65     {
66         "name": "userselector_searchanywhere",
67         "value": "0"
68     },
69     {
70         "name": "_lastloaded",
71         "value": 1619134031
72     }
73 ]
74 }
75 ]

```

D.3. Obtención de información de los cursos

Consulta

D.3.1. Envío

```

1 http://aulavirtual.anima.edu.uy/webservice/rest/server.php
2   ? wstoken = <anima_token>
3   & wsfunction = core_course_get_courses
4   & moodlesrestformat = json

```

D.3.2. Respuesta

```

1 [
2   {

```

```
3      "id": <identificador>,  
4      "shortname": "<curso_nombre_corto>",  
5      "categoryid": <identificador_categoria>,  
6      "categorysortorder": <orden_categoria>,  
7      "fullname": "<curso_nombre_completo>",  
8      "idnumber": "",  
9      "summary": "<resumen_curs>",  
10     "summaryformat": 1,  
11     "format": "site",  
12     "showgrades": 1,  
13     "newsitems": 3,  
14     "startdate": 0,  
15     "numsections": 1,  
16     "maxbytes": 0,  
17     "showreports": 0,  
18     "visible": 1,  
19     "groupmode": 0,  
20     "groupmodeforce": 0,  
21     "defaultgroupingid": 0,  
22     "timecreated": 1441205091,  
23     "timemodified": 1590787325,  
24     "enablecompletion": 0,  
25     "completionnotify": 0,  
26     "lang": "",  
27     "forcetheme": "",  
28     "courseformatoptions": [  
29         {  
30             "name": "numsections",  
31             "value": 1  
32         }  
33     ]  
34 }  
35 ]
```

D.3.3. Información utilizada

- **id:** identificador del curso auto generado por el sistema de Moodle, este es almacenado para luego poder realizar consultas sobre los cursos, poder identificar y relacionar los mismos dentro de la plataforma Google Data Studio.
- **shortname:** abreviación del nombre asignado al curso por parte de Ánima.
- **categoryid:** identificador de la categoría a la que pertenece el curso, una categoría puede incluir varios cursos y dependiendo de cómo esta se defina puede ser útil al momento de realizar filtros en la plataforma de Google Data Studio.
- **fullname:** nombre asignado al curso por parte de Ánima.

- **timecreated:** es la fecha en la que se creó el curso en Moodle.

D.4. Obtención de los contenidos de un curso

D.4.1. Consulta

Envío

```
1 http://aulavirtual.anima.edu.uy/webservice/rest/server.php
2   ? wstoken = <anima_token>
3   & wsfunction = core_course_get_contents
4   & moodlesrestformat = json
5   & courseid = <identificador_curso>
```

Respuesta

```
1 [
2   {
3     "id": 1,
4     "name": "Sitio",
5     "visible": 1,
6     "summary": "",
7     "summaryformat": 1,
8     "modules": [
9       {
10        "id": 62,
11        "url": "<curso_modulo_url>",
12        "name": "<modulo_nombre>",
13        "instance": 13,
14        "description": "<modulo_descripcion>",
15        "visible": 1,
16        "modicon": "<modulo_icono>",
17        "modname": "url",
18        "modplural": "URLs",
19        "indent": 0,
20        "contents": [
21          {
22            "type": "url",
23            "filename": "<documento_nombre>",
24            "filepath": null,
25            "filesize": 0,
26            "fileurl": "documento_url",
27            "timecreated": null,
28            "timemodified": 1456494691,
29            "sortorder": null,
30            "userid": null,
31            "author": null,
32            "license": null
```

```
33     }
34   ]
35 }
36 ]
37 }
38 ]
```

D.4.2. Información utilizada

Secciones

- **id:** identificador de una sección de un curso, este es asignado por Moodle.
- **name:** nombre de la sección asignado por un usuario de Anima.
- **summary:** resumen del contenido de la sección.
- **visible:** valor que indica si la sección es visible o no para los usuarios suscritos al curso.

Elementos

- **id:** identificador del elemento de Moodle.
- **instance:** identificador de la instancia del elemento, este es necesario para realizar las consultas de los detalles de estos elementos.
- **name:** nombre del elemento.
- **description:** descripción del elemento.
- **visible:** valor que indica si el elemento es visible o no para los usuarios suscritos al curso.
- **modname:** valor que identifica el tipo de elemento de Moodle, este puede ser de tipo “forum”, “url”, “assign”, entre otros.

D.5. Obtención de información de las entregas

D.5.1. Consulta

Envío

```
1 http://aulavirtual.anima.edu.uy/webservice/rest/server.php
2   ? wstoken = <anima_token>
3   & wsfunction = mod_assign_get_submissions
4   & moodlesrestformat = json
5   & assignmentids[<indice>] = <identificador_tarea>
```

Respuesta

```
1 {
2   "assignments": [
3     {
4       "assignmentid": 2,
5       "submissions": [
6         {
7           "id": 28,
8           "userid": 2,
9           "attemptnumber": 0,
10          "timecreated": 1455547242,
11          "timemodified": 1455547242,
12          "status": "new",
13          "groupid": 0,
14          "plugins": [
15            {
16              "type": "onlinetext",
17              "name": "Texto en linea",
18              "fileareas": [
19                {
20                  "area": "submissions_onlinetext"
21                }
22              ],
23              "editorfields": [
24                {
25                  "name": "onlinetext",
26                  "description": "Comentarios de
27                    la entrega",
28                  "text": "",
29                  "format": 0
30                }
31              ]
32            },
33            {
34              "type": "file",
35              "name": "Archivos enviados",
36              "fileareas": [
37                {
38                  "area": "submission_files"
39                }
40              ]
41            },
42            {
43              "type": "comments",
44              "name": "Comentarios de la entrega"
45            }
46          ]
47        }
48      ]
49    }
50  ]
51 }
```

```

46         }
47     ]
48 }
49 ]
50 }

```

D.5.2. Información utilizada

- **id:** identificador de la entrega realizada, es asignado por Moodle.
- **userid:** identificador del usuario que realizó la entrega, este es utilizado para luego relacionar a los usuarios con sus entregas.
- **attemptnumber:** número de intentos de entrega realizados por el alumno para una misma tarea.
- **timecreated:** fecha de creación del primer intento de entrega.
- **timemodified:** fecha de modificación del último intento de entrega.
- **status:** estado de la entrega, éste puede ser “new”, “draft” o “submitted”, como se describieron en la Sección 7.1.2.
- **groupid:** en caso de que la tarea sea grupal se asigna un identificador al grupo de alumnos que realizan la entrega.
- **assignmentid:** identificador de la tarea a la cual se está realizando la entrega.

D.6. Obtención de la calificación de las entregas

D.6.1. Consulta

Envío

```

1 http://aulavirtual.anima.edu.uy/webservice/rest/server.php
2   ? wstoken = <anima_token>
3   & wsfunction = mod_assign_get_grades
4   & moodlesrestformat = json
5   & assignmentids[<indice>] = <identificador_tarea>

```

Respuesta

```

1 {
2   "assignments": [
3     {
4       "assignmentid": 1,
5       "grades": [
6         {
7           "id": 369,

```

```
8         "userid": 168,  
9         "attemptnumber": 0,  
10        "timecreated": 1490873284,  
11        "timemodified": 1492961584,  
12        "grader": 18,  
13        "grade": "8.00000"  
14    }  
15 ]  
16 }  
17 ],  
18 "warnings": []  
19 }
```

D.6.2. Información utilizada

- **grade:** calificación de la entrega.
- **userid:** identificador del usuario que realizó la entrega.
- **assignmentid:** identificador de la tarea asociada a la entrega.

D.7. Obtención de información de los usuarios en un curso

D.7.1. Consulta

Envío

```
1 http://aulavirtual.anima.edu.uy/webservice/rest/server.php  
2   ? wstoken = <anima_token>  
3   & wsfunction = core_enrol_get_enrolled_users  
4   & moodlesrestformat = json  
5   & courseid = <identificador_curso>
```

Respuesta

```
1 [  
2   {  
3     "id": <identificador_usuario>,  
4     "username": "<nombre_usuario_anima>",  
5     "firstname": "<nombre_usuario>",  
6     "lastname": "<apellido_usuario>",  
7     "fullname": "<nombre_apellido_usuario>",  
8     "email": "<email_usuario>",  
9     "department": "",  
10    "firstaccess": 1441205126,  
11    "lastaccess": 1520878643,  
12    "description": "",
```

```
13     "descriptionformat": 1,  
14     "city": "Montevideo",  
15     "country": "UY",  
16     "profileimageurlsmall": "<imagen_chica_usuario>",  
17     "profileimageurl": "<imagen_usuario>",  
18     "groups": [],  
19     "roles": [  
20         {  
21             "roleid": 1,  
22             "name": "",  
23             "shortname": "<rol_usuario_curso>",  
24             "sortorder": 0  
25         }  
26     ],  
27     "enrolledcourses": [  
28         {  
29             "id": 27,  
30             "fullname": "<nombre_curso>",  
31             "shortname": "<nombre_corto_curso>"  
32         }  
33     ]  
34 }  
35 ]
```

D.7.2. Información utilizada

Usuario

- **id:** identificador de usuario asignado por Moodle.
- **username:** nombre de alias del usuario en Moodle.
- **firstname:** nombre del usuario.
- **lastname:** apellido del usuario.
- **fullname:** nombre y apellido del usuario.
- **email:** correo electrónico del usuario.

Roles

- **roleid:** identificador de un rol dentro de Moodle.
- **name:** nombre del rol en Moodle.
- **shortname:** abreviación del nombre del rol en Moodle.

D.8. Obtención de información de los foros de un curso

D.8.1. Consulta

Envío

```
1 http://aulavirtual.anima.edu.uy/webservice/rest/server.php
2   ? wstoken = <anima_token>
3   & wsfunction = mod_forum_get_forums_by_courses
4   & moodlesrestformat = json
5   & courseids[<indice>] = <identificador_curso>
```

Respuesta

```
1 [
2   {
3     "id": 1,
4     "course": "2",
5     "type": "news",
6     "name": "Novedades",
7     "intro": "Novedades",
8     "introformat": 1,
9     "assessed": 0,
10    "assesstimestart": 0,
11    "assesstimefinish": 0,
12    "scale": 100,
13    "maxbytes": 0,
14    "maxattachments": 1,
15    "forcesubscribe": 1,
16    "trackingtype": 1,
17    "rsstype": 0,
18    "rssarticles": 0,
19    "timemodified": 1444512042,
20    "warnafter": 0,
21    "blockafter": 0,
22    "blockperiod": 0,
23    "completiondiscussions": 0,
24    "completionreplies": 0,
25    "completionposts": 0,
26    "cmid": 2,
27    "numdiscussions": 1
28  }
29 ]
```

D.8.2. Información utilizada

- **id**: identificador del foro asignado por Moodle.

- **course:** identificador del curso al cual pertenece el foro.
- **type:** tipo de foro.
- **name:** nombre del foro.
- **intro:** descripción de uso del foro.
- **timemodified:** fecha de modificación del foro.

D.9. Obtención de información de las discusiones de un foro

D.9.1. Consulta

Envío

```
1 http://aulavirtual.anima.edu.uy/webservice/rest/server.php
2   ? wstoken = <anima_token>
3   & wsfunction = mod_forum_get_forum_discussions_paginated
4   & moodlesrestformat = json
5   & forumid = <identificador_foro>
```

Respuesta

```
1 {
2   "discussions": [
3     {
4       "id": 1,
5       "name": "Primeros pasos en Moodle",
6       "groupid": -1,
7       "timemodified": 1444512214,
8       "usermodified": 4,
9       "timestart": 0,
10      "timeend": 0,
11      "discussion": 1,
12      "parent": 0,
13      "userid": 4,
14      "created": 1444512214,
15      "modified": 1444512214,
16      "mailed": 1,
17      "subject": "Primeros pasos en Moodle",
18      "message": "<mensaje_discusion>",
19      "messageformat": 1,
20      "messagetrust": 0,
21      "attachment": "",
22      "totalscore": 0,
23      "mailnow": 1,
24      "userfullname": "Ismael Burone",
```

```
25     "usermodifiedfullname": "Ismael Burone",
26     "userpictureurl": "",
27     "usermodifiedpictureurl": "",
28     "numreplies": "0",
29     "numunread": 0
30   }
31 ],
32 "warnings": []
33 }
```

D.9.2. Información utilizada

- **discussion:** identificador de la discusión dentro del foro.
- **number:** identificador del foro.
- **name:** nombre del foro.
- **created:** fecha de creación de la discusión.
- **modified:** fecha de la última modificación de la discusión.
- **usermodified:** usuario que realizó la última modificación en la discusión.
- **subject:** título de la discusión.
- **message:** contenido de la discusión.

D.10. Obtención de información de los posts de una discusión

D.10.1. Consulta

Envío

```
1 http://aulavirtual.anima.edu.uy/webservice/rest/server.php
2   ? wstoken = <anima_token>
3   & wsfunction = mod_forum_get_forum_discussion_posts
4   & moodlesrestformat = json
5   & discussionid = <identificador_discusion>
```

Respuesta

```
1 {
2   "posts": [
3     {
4       "id": 1,
5       "discussion": 1,
6       "parent": 0,
7       "userid": 4,
```

```
8     "created": 1444512214,  
9     "modified": 1444512214,  
10    "mailed": 1,  
11    "subject": "Primeros pasos en Moodle",  
12    "message": "<mensaje_post>",  
13    "messageformat": 1,  
14    "messagetrust": 0,  
15    "attachment": "",  
16    "totalscore": 0,  
17    "mailnow": 1,  
18    "children": [],  
19    "canreply": true,  
20    "postread": false,  
21    "userfullname": "Ismael Burone",  
22    "userpictureurl": ""  
23  }  
24 ],  
25 "warnings": []  
26 }
```

D.10.2. Información utilizada

- **id:** identificador del post dentro de la discusión.
- **discussionid:** identificador de la discusión a la que pertenece el post.
- **userid:** identificador del usuario que creó el post.
- **created:** fecha de creación del post.
- **modified:** fecha de la última modificación del post.
- **subject:** título del post.
- **message:** contenido del post.

Apéndice E

Diagramas del Diseño del Data Warehouse

En este apéndice se ilustran los diagramas pertenecientes al diseño del Data Warehouse. A su vez se brinda los diagramas relacionados al proceso ETL para cargar los datos al Data Warehouse y ejemplos de consultas SQL para obtener reportes del mismo.

E.1. Dimensiones y Medidas

A continuación, se presentan los diagramas de las Dimensiones y Medidas obtenidas para el Data Warehouse diseñado.

La Figura E.1 contiene la Dimensión “Entrega Tareas”.

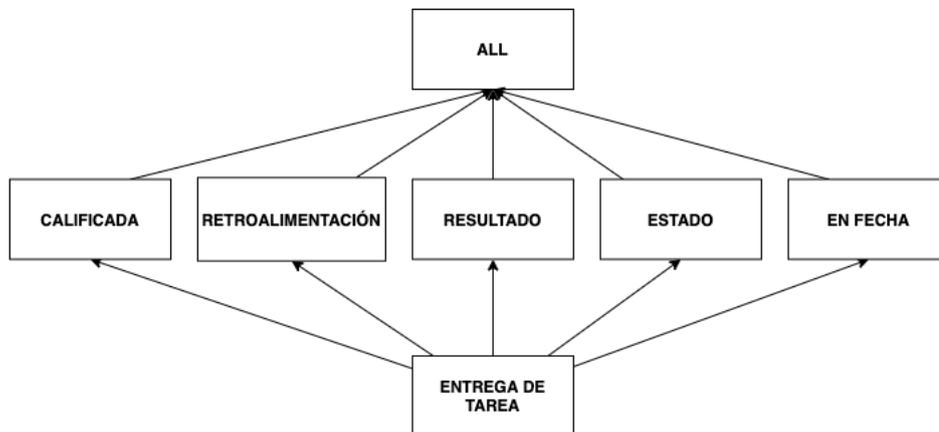


Figura E.1: Dimensión “Entrega Tareas”.

La Figura E.2 contiene la Dimensión “Tiempo”.

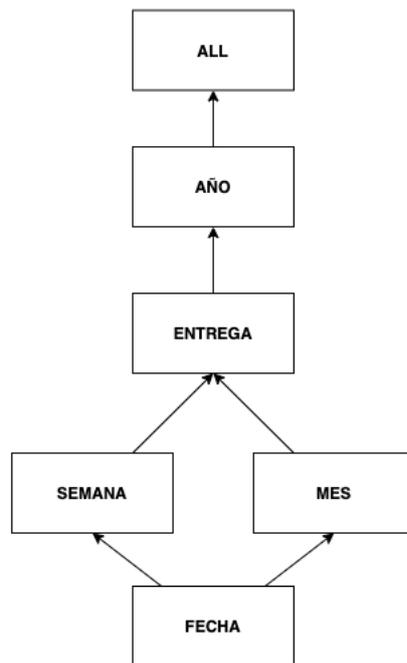


Figura E.2: Dimensión "Tiempo".

La Figura E.3 contiene la Dimensión "Académico".

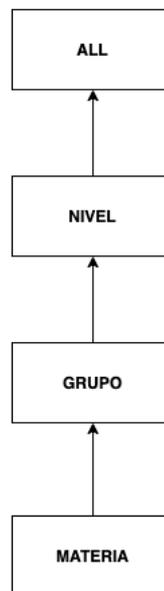


Figura E.3: Dimensión "Académico".

La Figura E.4 contiene la Dimensión "Usuario", en este caso queda implícito que contiene un nivel superior llamado "ALL".

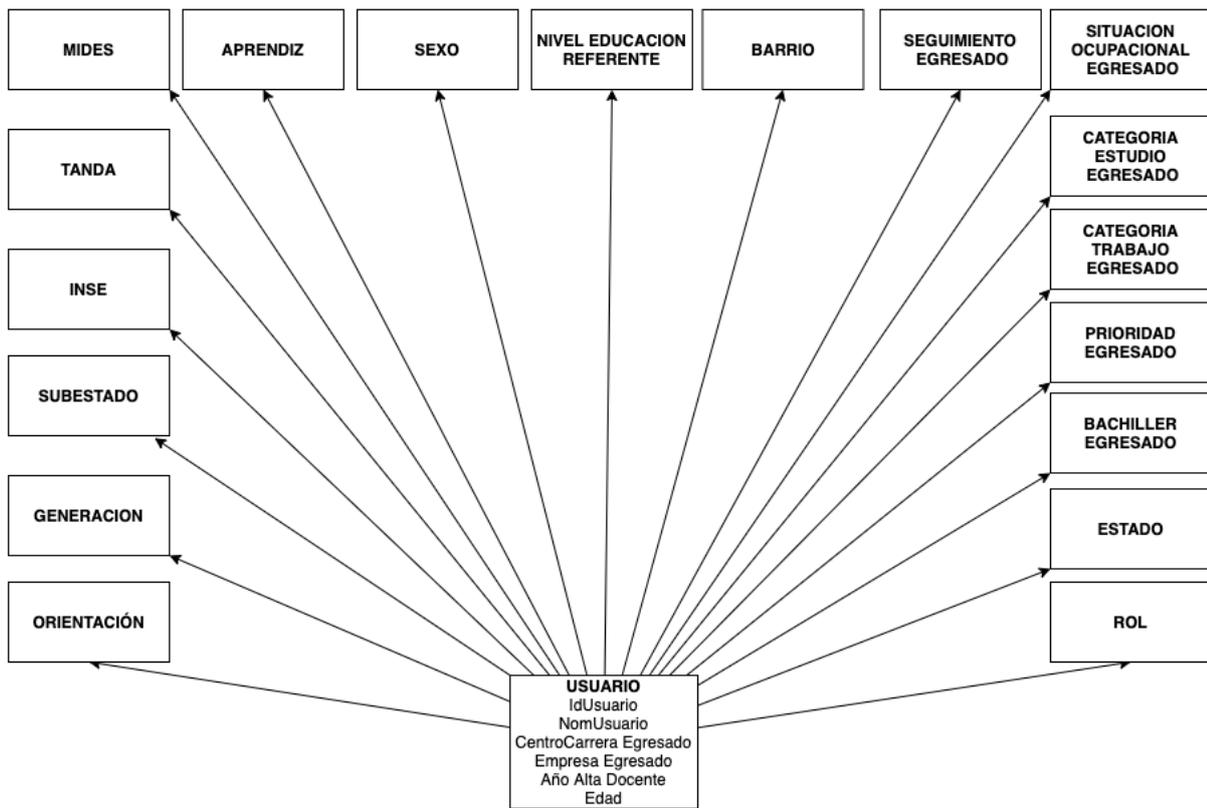


Figura E.4: Dimensión "Usuario".

La Figura E.5 contiene la Dimensión "Elementos Moodle".

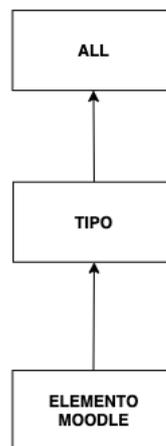


Figura E.5: Dimensión "Elementos Moodle".

La Figura E.6 muestra las Medidas obtenidas para el diseño del Data Warehouse.

M_CantidadAlumnos CantidadAlumnos	M_BoletinAlumno PromedioNotas PromedioBajas PromedioExámenes	M_Competicencias PromedioCompetencias
M_InasistenciasAlumno CantidadInasistencias CantidadHorasTotales	M_InasistenciasClaseDocente CantidadInasistencias CargaHorariaTotal	M_AsignaturasDocente CantidadAsignaturas
M_CantidadEntrevistas CantidadEntrevistas	M_CantidadRecomendaciones CantidadRecomendaciones	M_CantidadAccesosMoodle CantidadAccesos
M_CantidadAcciones CantidadAcciones	M_IntervencionLaboral CantidadIntervenciones	M_CantidadElementosMoodle CantidadElementosCreados
M_CantidadMensajesForoMoodle CantidadMensajesForo	M_InasistenciasCoordinacionesDocente CantidadInasistencias CargaHorariaTotal	M_ResultadoEntregasTareas CantidadEntregasTareas PromedioCalificacion

Figura E.6: Medidas.

E.2. Relaciones Dimensionales

En esta sección se presentan los diagramas de las Relaciones Dimensionales creados a partir de las Dimensiones y Medidas mostradas anteriormente.

La Figura E.7 asocia la Medida “M_ResultadoEntregasTareas” con las Dimensiones “Entrega Tarea”, “Usuario” con rol de alumno y de docente, “Académico” y “Tiempo”.

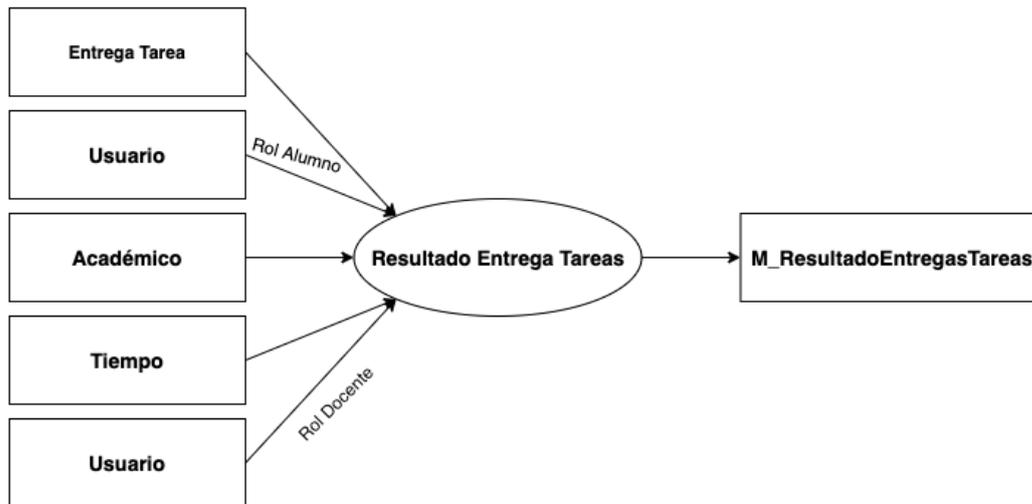


Figura E.7: Relación Dimensional "Resultado Entrega Tareas".

La Figura E.8 asocia la Medida "M_CantidadAlumnos" con las Dimensiones "Usuario" con rol de alumno, "Academico" y "Tiempo".

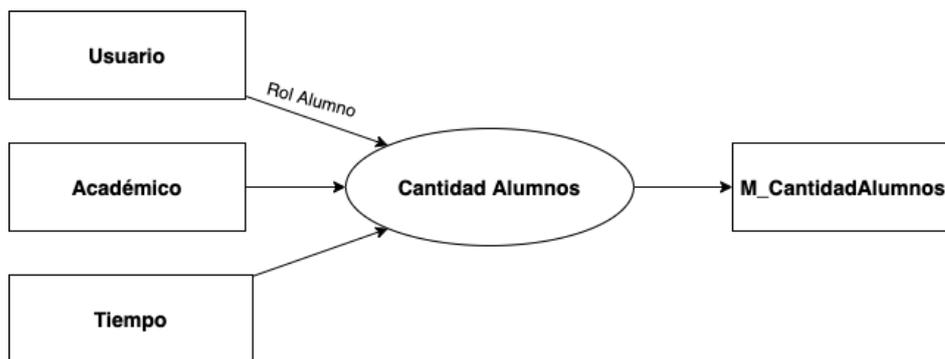


Figura E.8: Relación Dimensional "Cantidad Alumnos".

La Figura E.9 asocia la Medida "M_BoletinAlumno" con las Dimensiones "Usuario" con rol de alumno, "Usuario" con rol de docente, "Academico" y "Tiempo".

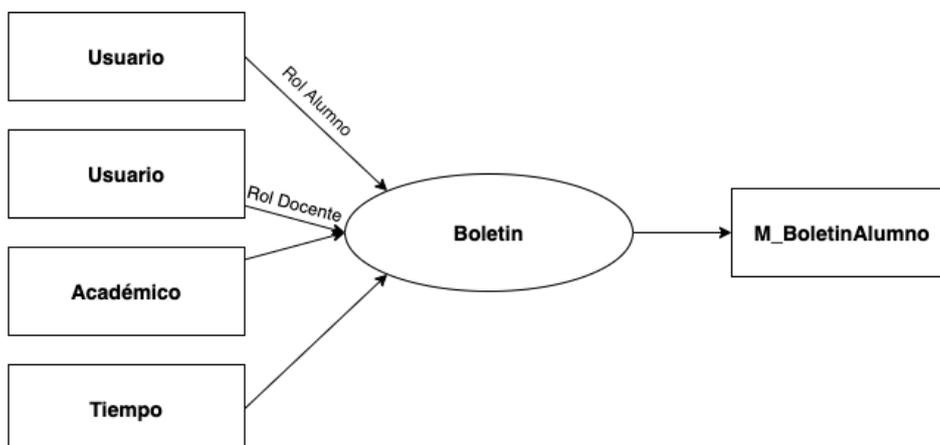


Figura E.9: Relación Dimensional "Boletin".

La Figura E.10 asocia la Medida “M_Competicencias” con las Dimensiones “Usuario” con rol de alumno, “Académico” y “Tiempo”.

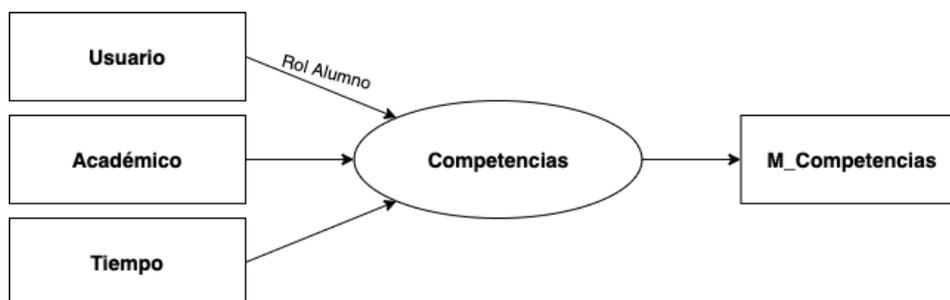


Figura E.10: Relación Dimensional “Competencias”.

La Figura E.11 asocia la Medida “M_InasistenciasAlumno” con las Dimensiones “Usuario” con rol de alumno, “Académico” y “Tiempo”.

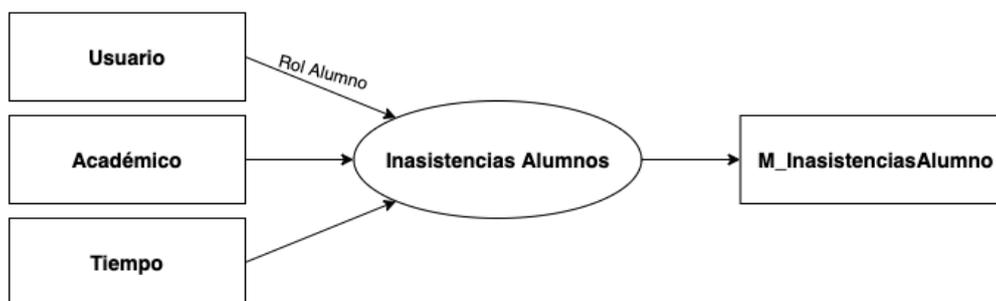


Figura E.11: Relación Dimensional “Inasistencias Alumnos”.

La Figura E.12 asocia la Medida “M_InasistenciasClaseDocente” con las Dimensiones “Usuario” con rol de docente y “Tiempo”.

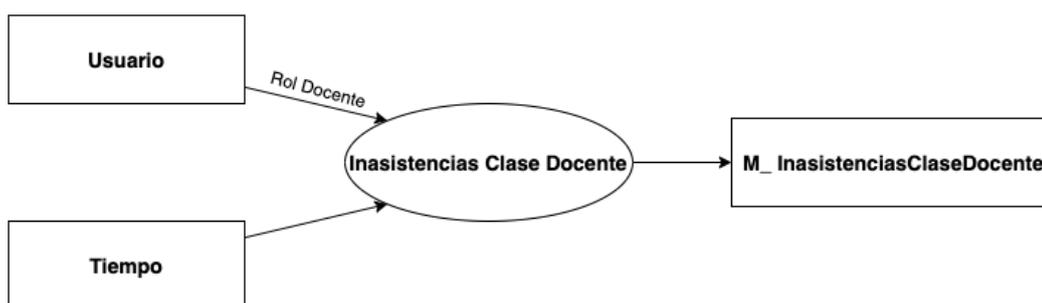


Figura E.12: Relación Dimensional “Inasistencias Clase Docente”.

La Figura E.13 asocia la Medida “M_InasistenciasCoordinacionesDocente” con las Dimensiones “Usuario” con rol de docente y “Tiempo”.

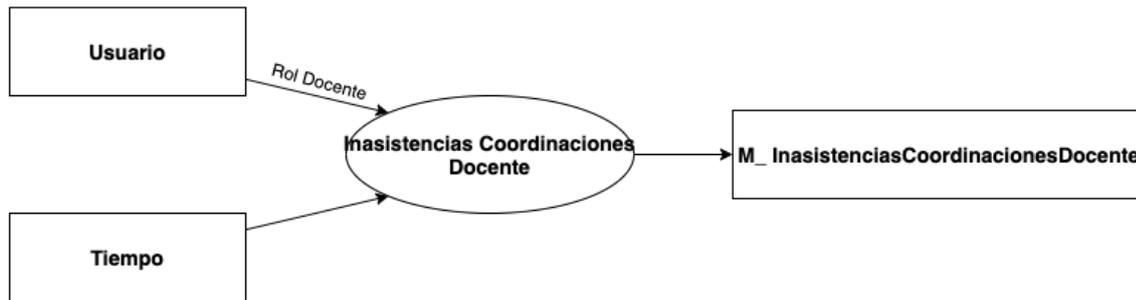


Figura E.13: Relación Dimensional "Inasistencias Coordinaciones Docente".

La Figura E.14 asocia la Medida "M_AsignaturasDocente" con las Dimensiones "Usuario" con rol de docente y "Tiempo".

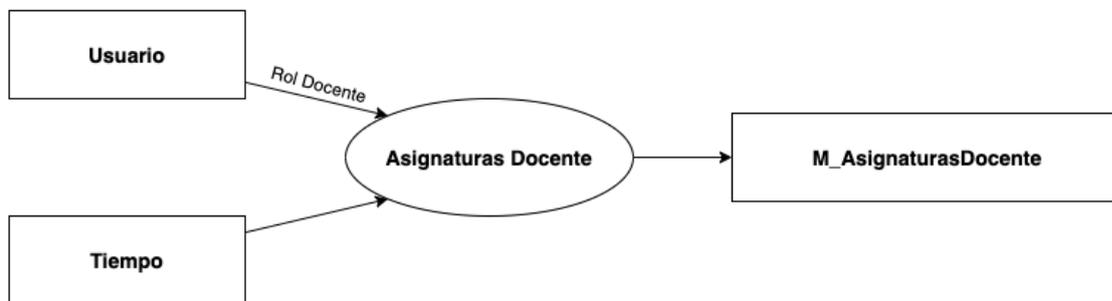


Figura E.14: Relación Dimensional "Asignaturas Docente".

La Figura E.15 asocia la Medida "M_CantidadElementosMoodle" con las Dimensiones "Elementos Moodle", "Usuario" con rol de docente, "Académico" y "Tiempo".

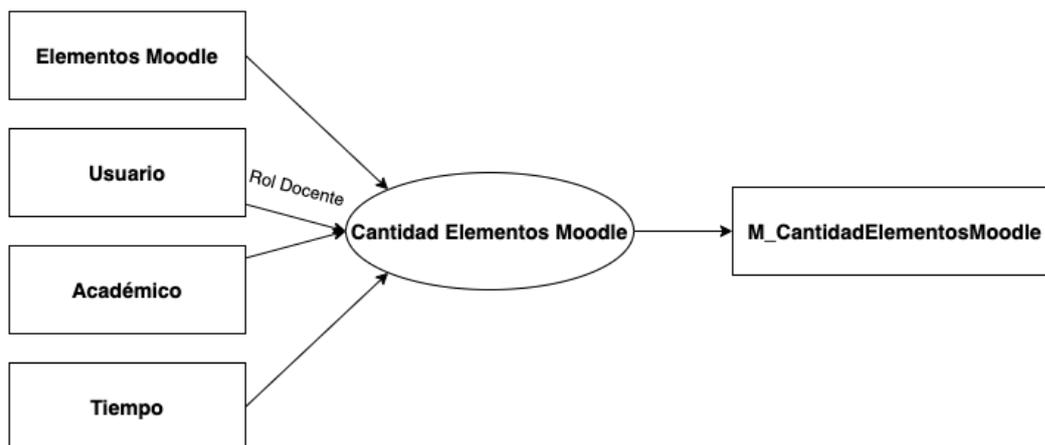


Figura E.15: Relación Dimensional "Cantidad Elementos Moodle".

La Figura E.16 asocia la Medida "M_CantidadEntrevistas" con las Dimensiones "Usuario" con rol de alumno y "Tiempo".



Figura E.16: Relación Dimensional "Cantidad Entrevistas".

La Figura E.17 asocia la Medida "M_CantidadRecomendaciones" con las Dimensiones "Tiempo", "Académico" y "Usuario" con rol de alumno.



Figura E.17: Relación Dimensional "Cantidad Recomendaciones".

La Figura E.18 asocia la Medida "M_CantidadAccesosMoodle" con las Dimensiones "Tiempo", "Académico" y "Usuario" con rol de docente y de usuario que accede.

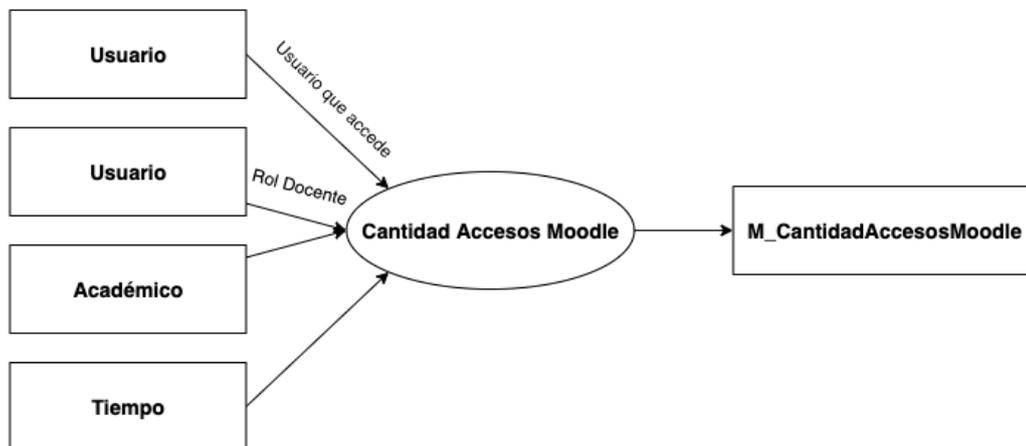


Figura E.18: Relación Dimensional "Cantidad Accesos Moodle".

La Figura E.19 asocia la Medida "M_CantidadAcciones" con las Dimensiones "Usuario" con el rol de alumno y "Tiempo".



Figura E.19: Relación Dimensional "Cantidad Acciones".

La Figura E.20 asocia la Medida “M_IntervencionLaboral” con las Dimensiones “Usuario” con el rol de alumno y “Tiempo”.



Figura E.20: Relación Dimensional “Cantidad Intervención Laboral”.

La Figura E.21 asocia la Medida “M_CantidadMensajesForoMoodle” con las Dimensiones “Académico”, “Tiempo” y “Usuario” como autor del mensaje y como docente.

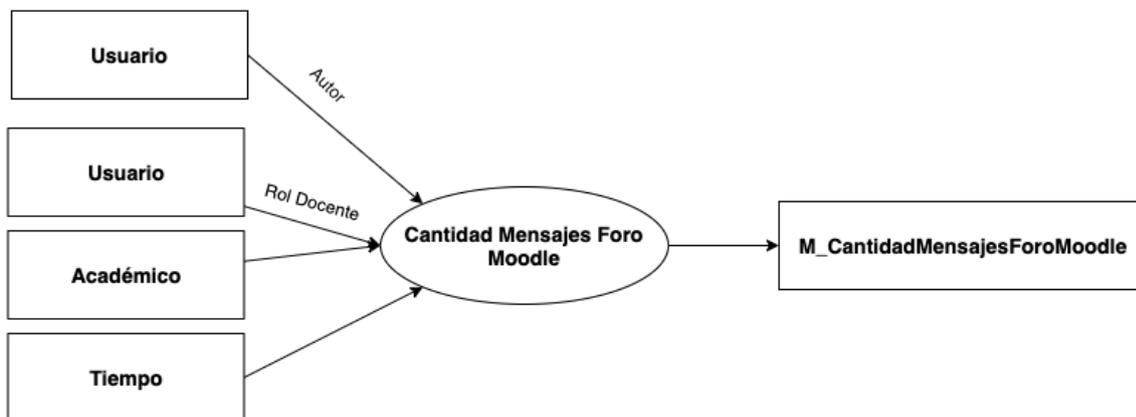


Figura E.21: Relación Dimensional “Cantidad Mensajes Foro Moodle”.

E.3. ETL

A continuación, se pueden observar los diagramas para el proceso ETL del Data Warehouse.

La Figura E.22 detalla desde dónde se deben tomar los datos para cargar la información a la tabla Usuario.

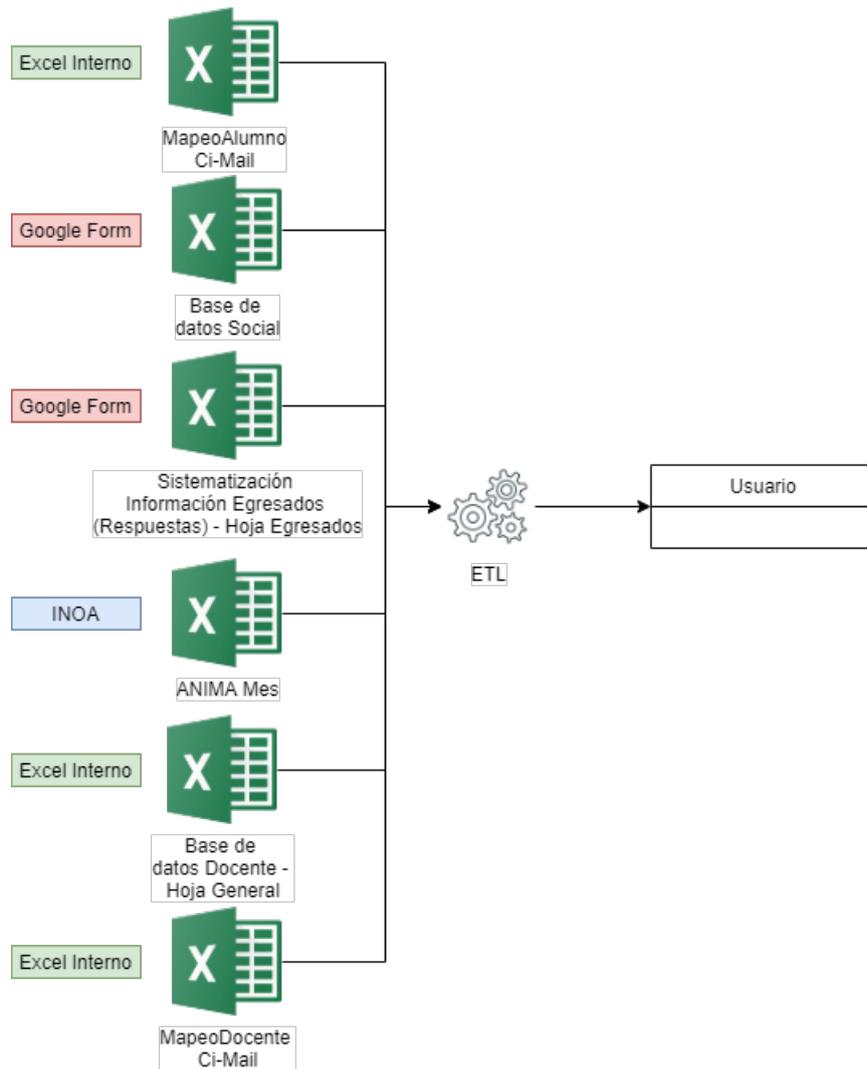


Figura E.22: Proceso de ETL de Usuarios

La Figura E.23 detalla desde dónde se deben tomar los datos para cargar la información de la tabla Académico.

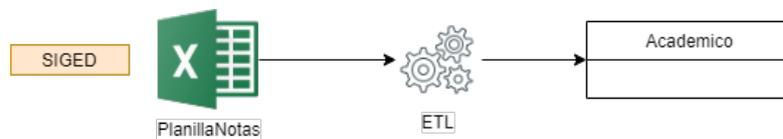


Figura E.23: Proceso de ETL de Académico

La Figura E.24 detalla desde dónde se deben tomar los datos para cargar la información de la tabla AsignaturaDocente, así como la tabla Tiempo.

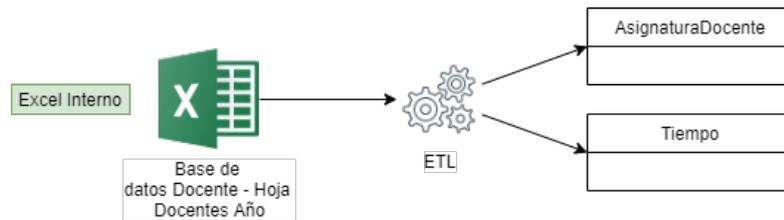


Figura E.24: Proceso de ETL Asignatura Docente

La Figura E.25 detalla desde dónde se deben tomar los datos para cargar la información de la tabla Boletin, así como la tabla Tiempo.

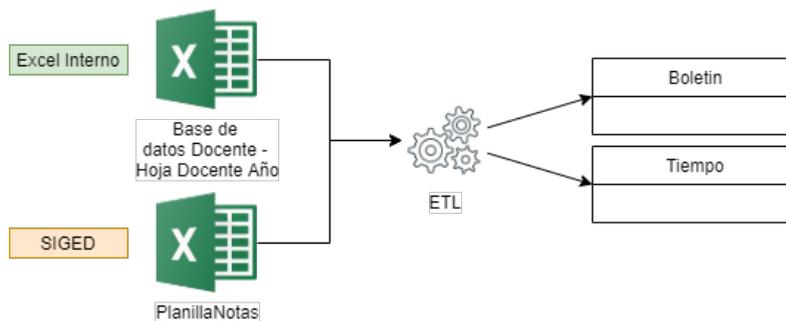


Figura E.25: Proceso de ETL Boletin

La Figura E.26 detalla desde dónde se deben tomar los datos para cargar la información de la tabla CantidadAccesosMoodle, así como la tabla Tiempo.

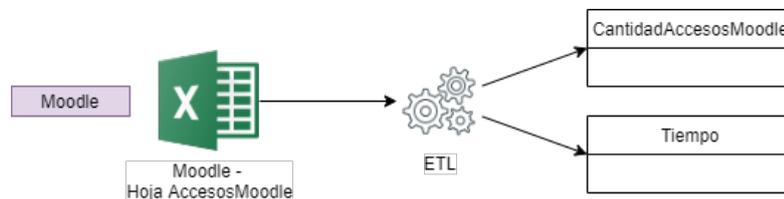


Figura E.26: Proceso de ETL Cantidad Acceso Moodle

La Figura E.27 detalla desde dónde se deben tomar los datos para cargar la información de la tabla CantidadAcciones, así como la tabla Tiempo.

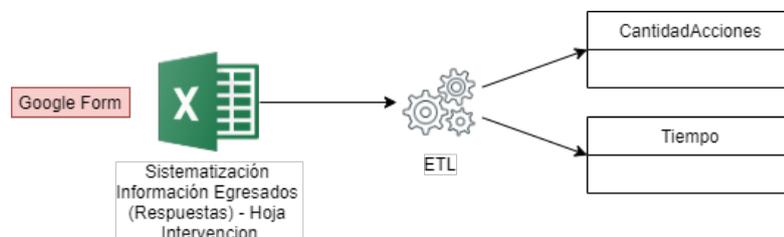


Figura E.27: Proceso de ETL Cantidad Acciones

La Figura E.28 detalla desde dónde se deben tomar los datos para cargar la información de la tabla CantidadAlumnos, así como la tabla Tiempo.

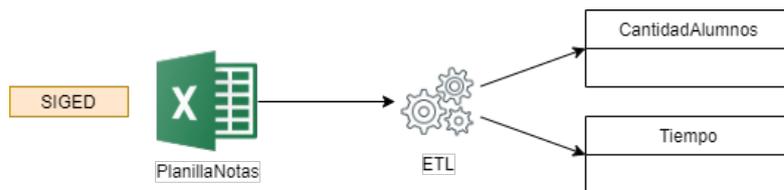


Figura E.28: Proceso de ETL Cantidad Alumnos

La Figura E.29 detalla desde dónde se deben tomar los datos para cargar la información de la tabla CantidadElementosMoodle, así como la tabla Tiempo.

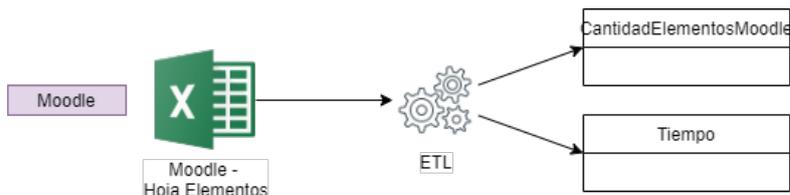


Figura E.29: Proceso de ETL Cantidad Elementos Moodle

La Figura E.30 detalla desde dónde se deben tomar los datos para cargar la información de la tabla CantidadEntrevistas, así como la tabla Tiempo.

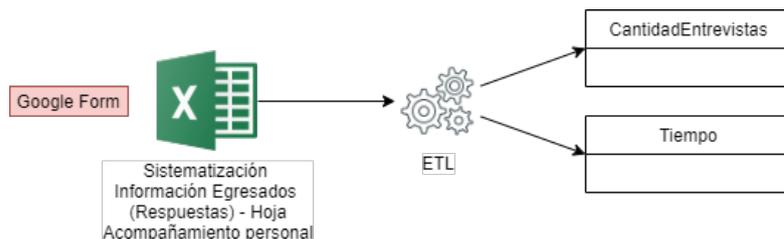


Figura E.30: Proceso de ETL Cantidad Entrevistas

La Figura E.31 detalla desde dónde se deben tomar los datos para cargar la información de la tabla CantidadIntervencionLaboral, así como la tabla Tiempo.

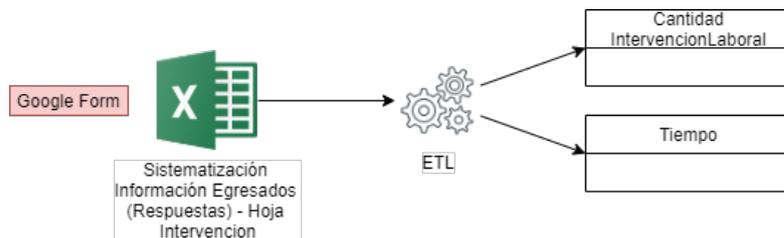


Figura E.31: Proceso de ETL Cantidad Intervencion Laboral

La Figura E.32 detalla desde dónde se deben tomar los datos para cargar la información de la tabla CantidadMensajesForoMoodle, así como la tabla Tiempo.

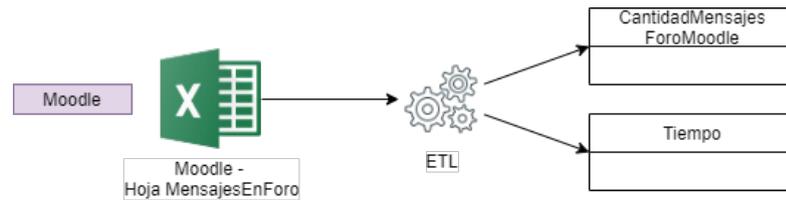


Figura E.32: Proceso de ETL Cantidad Mensajes Foros Moodle

La Figura E.33 detalla desde dónde se deben tomar los datos para cargar la información de la tabla CantidadRecomendaciones, así como la tabla Tiempo.

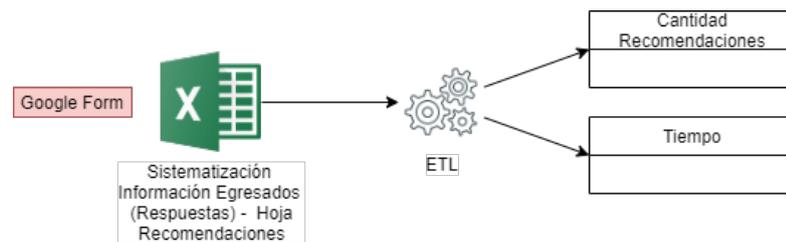


Figura E.33: Proceso de ETL Cantidad Recomendaciones

La Figura E.34 detalla desde dónde se deben tomar los datos para cargar la información de la tabla Competencias, así como la tabla Tiempo.

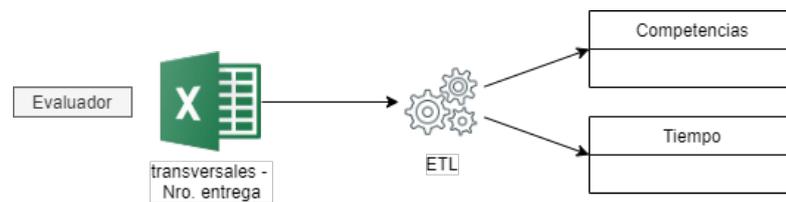


Figura E.34: Proceso de ETL Competencias

La Figura E.35 detalla desde dónde se deben tomar los datos para cargar la información de la tabla ElementoMoodle.

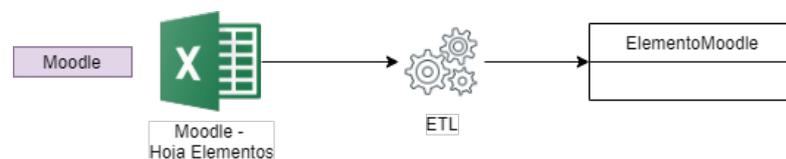


Figura E.35: Proceso de ETL Elemento Moodle

La Figura E.36 detalla desde dónde se deben tomar los datos para cargar la información de la tabla EntregaTareas.

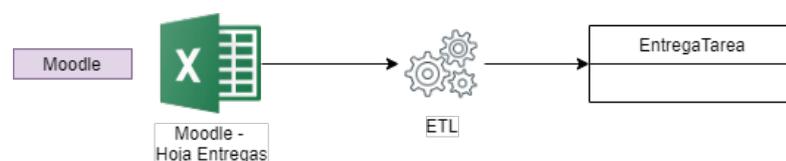


Figura E.36: Proceso de ETL Entrega Tareas

La Figura E.37 detalla desde dónde se deben tomar los datos para cargar la información de la tabla InasistenciasAlumno, así como la tabla Tiempo.

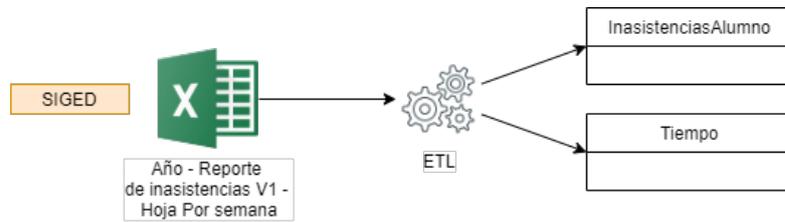


Figura E.37: Proceso de ETL Inasistencias Alumnos

La Figura E.38 detalla desde dónde se deben tomar los datos para cargar la información de la tabla InasistenciasClaseDocente, así como la tabla Tiempo.

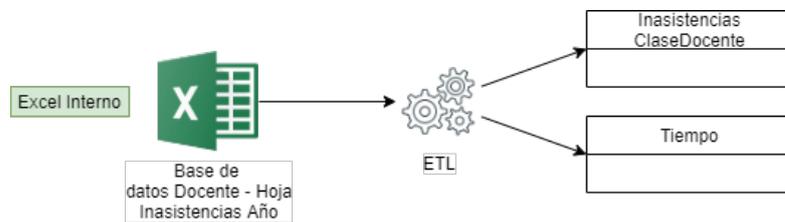


Figura E.38: Proceso de ETL Inasistencias Clase Docente

La Figura E.39 detalla desde dónde se deben tomar los datos para cargar la información de la tabla InasistenciasCoordinacionDocente, así como la tabla Tiempo.

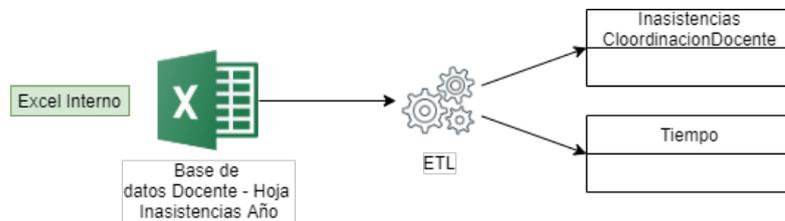


Figura E.39: Proceso de ETL Inasistencias Coordinación Docente

La Figura E.40 detalla desde dónde se deben tomar los datos para cargar la información de la tabla ResultadoEntregaTareas, así como la tabla Tiempo.

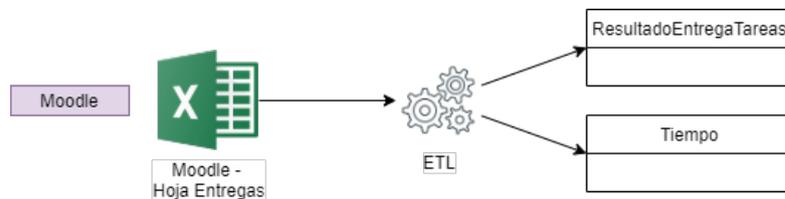


Figura E.40: Proceso de ETL Resultado Entrega Tarea

E.4. Consultas SQL Prototipo

En los siguientes fragmentos se muestran las consultas SQL utilizadas en el prototipo realizado.

E.4.1. Consulta de cantidad de entregas de tareas en Moodle

Se realiza esta consulta para obtener toda la cantidad de entregas de tareas dentro de la plataforma de Moodle e información de las mismas.

```

SELECT
    alumno.email alumno_email,
    alumno.nombre alumno_nombre,
    academico.nombre_materia materia,
    tiempo.fecha fecha,
    tiempo.anio anio,
    tiempo.mes mes,
    docente.nombre docente_nombre,
    entrega_tarea.estado estado_entrega,
    entrega_tarea.calificacion calificacion_entrega
FROM resultado_entrega_tarea
INNER JOIN entrega_tarea
    ON entrega_tarea.id = resultado_entrega_tarea.id_entrega_tarea
INNER JOIN tiempo
    ON tiempo.id = resultado_entrega_tarea.id_tiempo
INNER JOIN academico
    ON academico.id_materia = resultado_entrega_tarea.id_materia
INNER JOIN usuario alumno
    ON alumno.ci = resultado_entrega_tarea.ci_alumno
INNER JOIN usuario docente
    ON docente.ci = resultado_entrega_tarea.ci_docente;

```

E.4.2. Consulta de cantidad de mensajes en foros de Moodle

Esta consulta es con el fin de obtener los mensajes en los foros y su información relacionada en la plataforma de Moodle.

```

SELECT
    cantidad_mensajes_foro_moodle,
    autor.email usuario_email,
    autor.nombre usuario_nombre,
    academico.nombre_materia materia,
    tiempo.fecha fecha,
    tiempo.anio anio,
    tiempo.mes mes,
    docente.nombre docente_nombre
FROM
    cantidad_mensajes_foro_moodle
INNER JOIN tiempo
    ON tiempo.id = cantidad_mensajes_foro_moodle.id_tiempo
INNER JOIN academico
    ON academico.id_materia = cantidad_mensajes_foro_moodle.id_materia
INNER JOIN usuario autor
    ON autor.ci = cantidad_mensajes_foro_moodle.ci_usuario
INNER JOIN usuario docente

```

```
ON docente.ci = cantidad_mensajes_foro_moodle.ci_docente;
```

E.4.3. Consulta de cantidad de elementos en Moodle

En el siguiente caso se muestra la consulta SQL que obtiene los elementos en la plataforma de Moodle con información relacionada a los mismos.

```
SELECT
    elemento_moodle.nombre nombre_elemento,
    elemento_moodle.tipo tipo_elemento,
    academico.nombre_materia materia,
    tiempo.fecha fecha,
    tiempo.anio anio,
    tiempo.mes mes,
    docente.nombre docente_nombre
FROM
    cantidad_elementos_moodle
INNER JOIN tiempo
    ON tiempo.id = cantidad_elementos_moodle.id_tiempo
INNER JOIN academico
    ON academico.id_materia = cantidad_elementos_moodle.id_materia
INNER JOIN elemento_moodle
    ON elemento_moodle.id = cantidad_elementos_moodle.id_elemento_moodle
INNER JOIN usuario docente
    ON docente.ci = cantidad_elementos_moodle.ci_docente;
```