

Universidades de Burgos, León y Valladolid

Máster universitario

Inteligencia de Negocio y Big Data en Entornos Seguros



Trabajo Fin de Máster

**Plugin KNIME para conexión con
Moodle**

Presentado por Francisco Gil Rodríguez
en la Universidad de Burgos — 1 de febrero de 2024

Tutores: Dr. José Francisco Díez Pastor y
Dr. César Ignacio García Osorio

Universidades de Burgos, León y Valladolid



Máster universitario en Inteligencia de Negocio y Big Data en Entornos Seguros

D. José Francisco Díez Pastor, profesor del departamento de Ingeniería Informática, área de Lenguajes y Sistemas Informáticos.

D. César Ignacio García Osorio, profesor del departamento de Ingeniería Informática, área de Lenguajes y Sistemas Informáticos.

Exponen:

Que el alumno D. Francisco Gil Rodríguez, con DNI 42183724E, ha realizado el Trabajo final de Máster en Inteligencia de Negocio y Big Data en Entornos Seguros titulado «Plugin KNIME para conexión con Moodle».

Y que dicho trabajo ha sido realizado por el alumno bajo la dirección del que suscribe, en virtud de lo cual se autoriza su presentación y defensa.

En Burgos, 1 de febrero de 2024

Vº. Bº. del Tutor:

Vº. Bº. del Tutor:

D. José Francisco Díez Pastor

D. César Ignacio García Osorio

Resumen

KNIME es una plataforma de código abierto orientada al análisis de datos que permite, a través de una interfaz intuitiva y visual, crear análisis complejos y flujos de trabajo mediante componentes especializados en diferentes áreas como la minería de datos, el aprendizaje automático o la visualización de datos. Moodle es una plataforma de aprendizaje (Learning Management System) ampliamente utilizada por instituciones educativas de todo el mundo. En este trabajo se han desarrollado nuevos componentes de KNIME que permiten la integración en flujos de trabajo de KNIME de los datos que Moodle va registrando durante la ejecución de una acción formativa. Con un enfoque orientado al rol de profesor, se pretende que cualquier usuario con este rol dentro de una plataforma Moodle, pueda realizar estudios externos sobre los datos de las acciones formativas a las que tiene acceso. Adicionalmente, y a modo de presentación de los componentes desarrollados, se ha implementado un flujo de trabajo en KNIME sobre datos reales extraídos de Moodle, analizando diferentes técnicas de aprendizaje automático supervisado.

Descriptores

Moodle, KNIME, ETL, aprendizaje automático, aprendizaje supervisado, Java

Abstract

KNIME is an open source platform oriented to data analysis that allows, through an intuitive and visual interface, the creation of complex analysis and *workflows* through specialized components in different areas such as data mining, machine learning or data visualization. Moodle is a learning platform (Learning Management System) widely used by educational institutions around the world. In this work we have developed new KNIME components that allow the integration of the data recorded by Moodle during the execution of a training action into KNIME *workflows*. With an approach oriented to the role of teacher, it is intended that any user with this role within a Moodle platform, can perform external studies on the data of the training actions to which they have access. Additionally, and as a presentation of the developed components, a KNIME workflow has been implemented on real data extracted from Moodle, analyzing different supervised machine learning supervised machine learning techniques.

Keywords

Moodle, KNIME, ETL, machine learning, supervised learning, Java

Índice general

Índice general	iii
Índice de figuras	v
Índice de tablas	vii
 Memoria	 1
1. Introducción	3
1.1. Estructura de la memoria	4
1.2. Estructura de los apéndices	5
2. Objetivos del proyecto	7
2.1. Objetivos generales	7
2.2. Objetivos técnicos	7
2.3. Objetivos personales	8
3. Conceptos teóricos	9
3.1. KNIME	9
3.2. Moodle	12
3.3. ETL	13
3.4. Aprendizaje supervisado	13
4. Técnicas y herramientas	15
4.1. Entorno de desarrollo	15
4.2. Desarrollo	18
4.3. Memoria	20

5. Aspectos relevantes del desarrollo del proyecto	21
5.1. Análisis	21
5.2. Diseño	26
5.3. Implementación	27
5.4. Workflow de KNIME con datos reales	30
6. Trabajos relacionados	45
6.1. UBU Monitor	45
7. Conclusiones y Líneas de trabajo futuras	47
7.1. Conclusiones	47
7.2. Líneas de trabajo futuras	48
 Apéndices	 51
Apéndice A Plan de Proyecto Software	53
A.1. Metodología	53
A.2. Planificación temporal	53
A.3. Estudio de viabilidad	59
Apéndice B Especificación de requisitos	61
B.1. Introducción	61
B.2. Catálogo de requisitos	61
Apéndice C Especificación de diseño	63
C.1. Introducción	63
C.2. Diseño de la extensión KNIME Moodle Integration	63
Apéndice D Documentación técnica de programación	81
D.1. Introducción	81
D.2. Estructura de directorios	81
D.3. Desarrollo de nuevos nodos	84
Apéndice E Documentación de usuario	85
E.1. Introducción	85
E.2. Instalación KNIME	85
E.3. Instalación de la extensión Moodle KNIME Integration	85
E.4. Crear un <i>workflow</i> y usar la extensión	89
 Bibliografía	 93

Índice de figuras

3.1	Ejemplo de Workflow en KNIME.	10
3.2	<i>Workflow</i> de ejemplo con elementos de KNIME.	12
5.3	<i>Workflow</i> : preparación de datos.	32
5.4	<i>Workflow</i> : preparación de datos (Prepare data desplegado).	33
5.5	<i>Workflow</i> : componente Video views count	34
5.6	<i>Workflow</i> : preparación de datos (salida).	35
5.7	<i>Workflow</i> : preparación de datos (diagrama de barras clase).	36
5.8	<i>Workflow</i> completo.	37
5.9	Modelo A: <i>Workflow</i>	37
5.10	Modelo A: Componente AutoML expandido.	38
5.11	Modelo A: Componente AutoML configuración.	39
5.12	Modelo A: Resultados (verde \rightarrow <i>Accuracy</i>).	40
5.13	Modelo B: <i>Workflow</i>	40
5.14	Modelo B: SMOTE	41
5.15	Modelo B: Resultados (verde \rightarrow <i>Accuracy</i>).	42
5.16	Modelo C: <i>Workflow</i>	42
5.17	Modelo C: <i>Best parameters</i>	43
5.18	Modelo C: Tabla de confusión.	43
C.2.1	Nodos implementados en la extensión KNIME Moodle Integration	64
C.2.2	Nodo Moodle Connector . Descripción.	65
C.2.3	Nodo Moodle Connector . Configuración.	66
C.2.4	Nodo Moodle Courses . Descripción.	67
C.2.5	Nodo Moodle Courses . Configuración.	68
C.2.6	Nodo Moodle Courses . Ejemplo de salida.	69
C.2.7	Nodo Moodle Users . Descripción.	71
C.2.8	Nodo Moodle Users . Configuración.	72

C.2.9	Nodo Moodle Users . Salida de ejemplo.	73
C.2.10	Nodo Moodle Reports Logs . Descripción.	74
C.2.11	Nodo Moodle Reports Logs . Salida.	75
C.2.12	Nodo Moodle Reports Grades . Descripción.	76
C.2.13	Nodo Moodle Reports Grades . Salida.	77
C.2.14	Nodo Moodle Reports Quizzes . Descripción.	78
C.2.15	Nodo Moodle Reports Quizzes . Salida.	79
E.3.1	Instalación de la extensión Moodle en KNIME 1.	86
E.3.2	Instalación de la extensión Moodle en KNIME 2.	86
E.3.3	Instalación de la extensión Moodle en KNIME 3.	87
E.3.4	Instalación de la extensión Moodle en KNIME 4.	87
E.3.5	Instalación de la extensión Moodle en KNIME 5.	88
E.4.1	Usando la extensión Moodle . Repositorio de nodos.	89
E.4.2	Usando la extensión Moodle 1.	90
E.4.3	Usando la extensión Moodle 2.	90
E.4.4	Usando la extensión Moodle 3.	91
E.4.5	Usando la extensión Moodle 4.	91
E.4.6	Usando la extensión Moodle 5.	92

Índice de tablas

A.1	Costes de personal.	59
A.2	Costes de hardware.	60
A.3	Costes totales del proyecto.	60
C.1	Tabla de salida del nodo Moodle Courses	69
C.2	Tabla de salida del nodo Moodle Users	72
C.3	Tabla de salida del nodo Moodle Reports Logs	75
C.4	Tabla de salida del nodo Moodle Reports Grades	77
C.5	Tabla de salida del nodo Moodle Reports Quizzes	79

Memoria

Introducción

Moodle es una plataforma de aprendizaje (*LMS, Learning Management System*) ampliamente utilizada por instituciones educativas de todo el mundo, incluyendo universidades. Es, de hecho, el *LMS* que se utiliza en UBUVirtual, plataforma de aprendizaje donde se desarrolla este máster. Durante el tiempo de vida de una acción formativa o curso, Moodle registra todo tipo de información relacionada con la interacción de los participantes con la plataforma. Por ejemplo, se registra cada vez que un usuario consulta un recurso, visita una página, visualiza un vídeo o participa en un foro de discusión. También quedan registradas las calificaciones obtenidas en las diferentes actividades evaluables y los tiempos empleados en la realización de cuestionarios. Esta información puede ser muy variada, ya que depende de la configuración de los componentes integrados dentro de cada curso. Ahora bien, aunque mucha de esta información registrada puede ser consultada por el profesor a través de informes que están disponibles dentro del curso, no es fácilmente exportable para su procesamiento desde herramientas externas.

Utilizando técnicas de aprendizaje automático se puede extraer conocimiento sobre el curso y sus participantes. Este conocimiento, en manos del profesor responsable, puede servir para mejorar el curso y/o tomar decisiones que afecten al desempeño de los estudiantes. Supongamos, por ejemplo, que el profesor dispone de un modelo que le permite predecir qué estudiantes están en riesgo de abandonar o no finalizar el curso. A partir de esta información, el profesor puede tomar las medidas que considere necesarias para intentar «recuperar» a estos estudiantes (enviar un correo electrónico, agendar una tutoría individual, proponer materiales de apoyo, etc.). Además, se podría volver a aplicar el mismo modelo pasado un tiempo para comprobar si los estudiantes han salido de la zona de riesgo o, por el contrario, se requieren acciones adicionales.

En las últimas versiones de Moodle se han ido incorporando algunos modelos predictivos, tanto en el núcleo como a través de *plugins* externos. Estos modelos pueden ser útiles para los casos de uso más comunes, pero siempre estarán limitados con respecto a las posibilidades de analizar los datos de una forma más libre y abierta desde herramientas externas, como es el caso de la herramienta KNIME que abordamos en este trabajo.

KNIME [6] es una plataforma de código abierto orientada al análisis de datos que permite, a través de una interfaz intuitiva y visual, crear análisis complejos y flujos de trabajo mediante componentes especializados en diferentes áreas como la minería de datos, el aprendizaje automático o la visualización de datos. En este trabajo se han desarrollado nuevos componentes de KNIME que permiten la integración en sus flujos de trabajo de los datos que Moodle va registrando durante la ejecución de una acción formativa. Con un enfoque orientado al rol de profesor, se pretende que cualquier usuario con este rol dentro de una plataforma Moodle, pueda realizar estudios externos sobre los datos de las acciones formativas a las que tiene acceso. Adicionalmente, y a modo de presentación de los componentes desarrollados, se ha implementado un flujo de trabajo en KNIME sobre datos reales extraídos de Moodle, analizando diferentes técnicas de aprendizaje automático supervisado.

1.1. Estructura de la memoria

La memoria se compone de los siguientes apartados:

1. **Introducción.** Introducción al proyecto y al trabajo desarrollado.
2. **Objetivos del proyecto.** Se describen los objetivos generales, técnicos y personales establecidos al inicio del proyecto.
3. **Conceptos teóricos.** Se introducen los principales conceptos teóricos abordados en el proyecto.
4. **Técnicas y herramientas.** Se describen las herramientas utilizadas para el desarrollo del proyecto.
5. **Aspectos relevantes del desarrollo.** Recoge los aspectos más interesantes y relevantes del desarrollo del proyecto.

6. **Trabajos relacionados.** Se describen otros trabajos relacionados con el proyecto y el grado de influencia que han tenido en el desarrollo.
7. **Conclusiones y líneas de trabajo futuras.** Conclusiones finales del proyecto y posibles líneas de trabajo que se podrían seguir para ampliar o dar continuidad al proyecto.

1.2. Estructura de los apéndices

También se incluyen los siguientes apéndices anexos al proyecto:

- A. **Plan de proyecto.** Descripción de la metodología utilizada, planificación temporal y viabilidad económica.
- B. **Especificación de diseño.** Incluye la especificación de requisitos y el diseño implementado.
- C. **Documentación técnica de programación.** Descripción de los pasos a seguir para instalar, compilar y ejecutar el proyecto.
- D. **Documentación de usuario.** Manual de usuario de los componentes KNIME desarrollados.

Tanto la memoria como el código desarrollado se puede consultar y descargar desde el [repositorio GitHub del proyecto](https://github.com/frankgil/knime-moodle)¹.

¹<https://github.com/frankgil/knime-moodle>

Objetivos del proyecto

En este apartado se exponen los objetivos del proyecto, diferenciados entre objetivos generales, técnicos y personales.

2.1. Objetivos generales

- Desarrollar un plugin de KNIME que permita importar datos desde una plataforma de teleformación Moodle, para su tratamiento posterior dentro de flujos de trabajo de KNIME.
- Facilitar la incorporación en KNIME de datos de cursos de cualquier plataforma Moodle, sin necesidad de realizar ajustes personalizados dentro de la plataforma de formación.
- Orientar la solución al perfil de profesor, de forma que cualquier usuario con este perfil pueda realizar estudios desde KNIME de los datos relativos a sus cursos.
- Implementar un ejemplo práctico utilizando el plugin de KNIME desarrollado en un flujo de trabajo relacionado con aprendizaje supervisado.

2.2. Objetivos técnicos

- Desarrollar un plugin para KNIME en lenguaje de programación Java.
- Explorar la arquitectura de KNIME para elegir el tipo de componentes a desarrollar que mejor se adapten a la solución requerida.

- Conocer a fondo la metodología de programación en KNIME mediante el estudio de la documentación existente y la inspección de *plugins* de KNIME similares.
- Explorar los nodos y *workflows* de KNIME disponibles en el ámbito del aprendizaje supervisado.

2.3. Objetivos personales

- Aplicar los conocimientos adquiridos dentro del máster en al ámbito educativo y más específicamente en el de la formación online.
- Conocer más a fondo las herramientas que, como KNIME, permiten el análisis de datos desde la interfaz de usuario, tanto a nivel de usuario como a nivel de desarrollador.
- Contribuir a la mejora de la formación online al facilitar que los profesores puedan realizar análisis sobre sus cursos sin conocimientos de programación.

Conceptos teóricos

3.1. KNIME

KNIME es un software de código abierto orientado a la ciencia de datos. Puede ser descargado y utilizado gratuitamente bajo los términos de la Licencia Pública General de GNU versión 3 (GPLv3). A través de una interfaz gráfica intuitiva, permite crear flujos de trabajo de análisis de datos y minería de datos. KNIME fue desarrollado en la Universidad de Konstanz (Alemania) en 2004, y su nombre proviene de su denominación en inglés «*Konstanz Information Miner*»[6] [10].

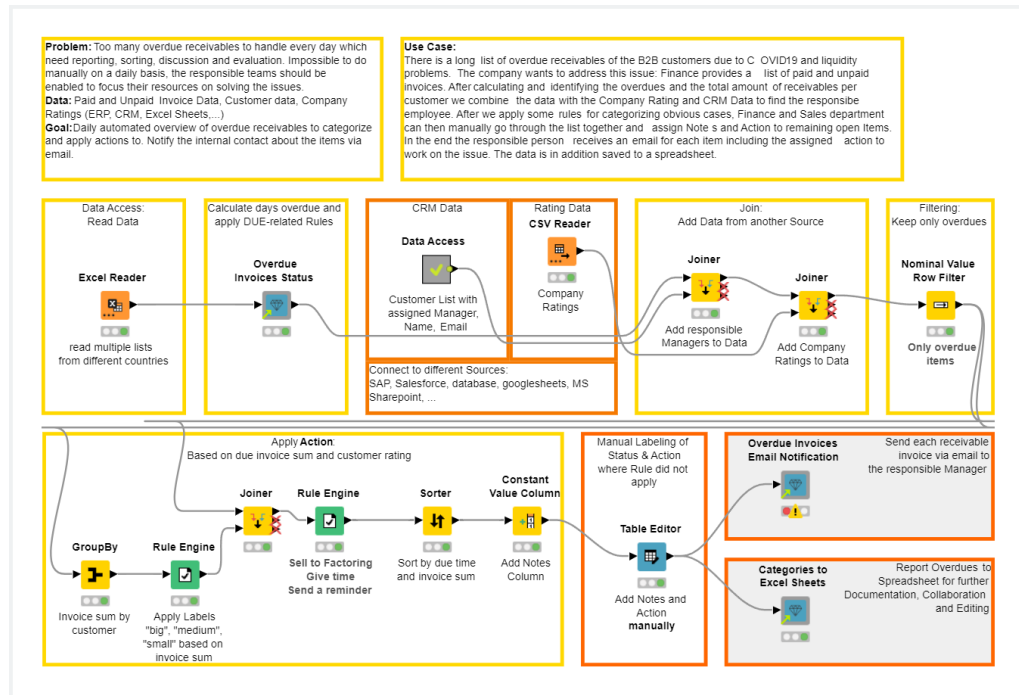


Figura 3.1: Ejemplo de Workflow en KNIME.

KNIME está concebido como una herramienta gráfica y dispone de una serie de nodos, que encapsulan distintos tipos de algoritmos, y flechas, que representan flujos de datos, que se despliegan y combinan de manera gráfica e interactiva.

KNIME cuenta con un [repositorio abierto](https://hub.knime.com/)¹ desde donde se pueden descargar nodos, *workflows*, componentes y extensiones.

KNIME incluye más de 2000 nodos nativos, que puede extenderse con otros más de 4500 nodos disponibles en el *hub* de la comunidad. Cubre prácticamente todos los aspectos de la ciencia de datos, desde recopilar y manipular datos hasta darles sentido con técnicas sofisticadas de modelado y visualización.

El carácter abierto de la herramienta hace posible su extensión mediante la creación de nuevos nodos que implementen algoritmos a la medida del usuario. Además, existe la posibilidad de utilizar llamadas directas a Weka o de incorporar de manera sencilla código desarrollado en R o Python.

¹<https://hub.knime.com/>

Además de la versión *Community*, que es libre y gratuita, KNIME ofrece soluciones de pago para empresas, como *KNIME Server* y *KNIME Business Hub*. Estas soluciones se pueden instalar en los servidores del cliente y están diseñadas para la colaboración en equipo, la automatización, la gestión y el despliegue de flujos de trabajo. En este proyecto nos centraremos únicamente en la versión *Community*.

En la arquitectura de KNIME se compone principalmente de los siguientes elementos:

- **Nodos.** Un nodo representa una función específica que se aplica a los datos. Los nodos realizan operaciones específicas, como lectura de datos, preprocesamiento, análisis, visualización o exportación de datos.
- **Puertos.** Los puertos son las conexiones de entrada y salida que los nodos pueden tener para conectarse y comunicarse entre ellos.
- **Workflows.** Un *workflow* o flujo de trabajo es una representación visual de un conjunto de operaciones que se realiza mediante nodos interconectados.
- **Componentes.** Los componentes son nodos que contienen un subflujo de trabajo, lo que nos permite encapsular funcionalidad para reutilizar en diferentes flujos de trabajo. Pueden tener su propio cuadro de diálogo de configuración, lo que nos permite ajustar la funcionalidad sin necesidad de modificar los componentes que contiene.
- **Metanodos.** Los metanodos se utilizan para simplificar los flujos de trabajo. Se trata de seleccionar una parte del flujo, compuesta por varios nodos y sus relaciones, y agruparlos en un nodo especial llamado metanodo, con lo que ocultamos esa parte del flujo y lo simplificamos visualmente. A diferencia de los componentes, los metanodos se utilizan solo como contenedores para mejorar la visualización de un flujo complejo, teniendo siempre que expandirlos para realizar modificaciones en esa parte del flujo de trabajo.
- **Extensiones.** Las extensiones son *plugins* adicionales que podemos añadir a KNIME para ampliar las funcionalidades que no están incluidas en el núcleo. Generalmente las extensiones incluyen nodos agrupados dentro de una misma temática.

En la Figura 3.2 se muestra un *workflow* muy básico con dos nodos interconectados entre sí. El nodo **Table Creator** permite definir una tabla de datos con varias columnas y filas. Este nodo no tiene puerto de entrada. El puerto de salida devuelve los datos, y se conecta a un nodo de tipo **Sorter** a través de su puerto de entrada. Este otro nodo ordena los datos en función de la columna indicada en su configuración. El puerto de salida del nodo **Sorter** será una tabla similar a la definida en **Table Creator**, pero ordenada según lo especificado.

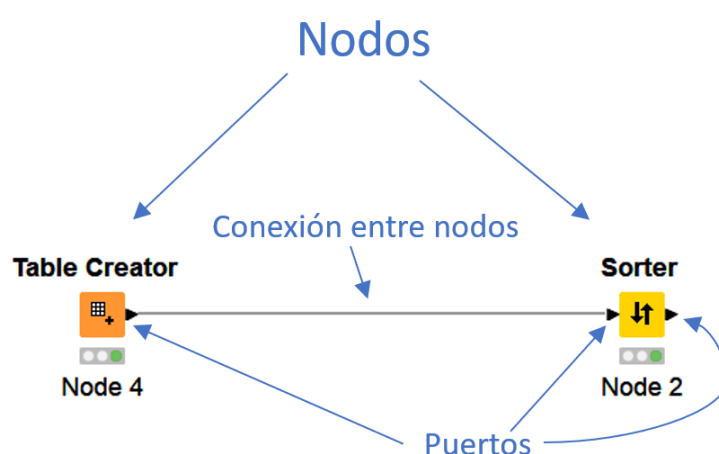


Figura 3.2: *Workflow* de ejemplo con elementos de KNIME.

3.2. Moodle

Moodle es una plataforma de gestión del aprendizaje (*LMS*, *Learning Management System*) de código abierto diseñada para crear y administrar cursos en línea. Moodle se ha convertido en una de los LMS más populares y ampliamente utilizados en todo el mundo, con una importante presencia en el ámbito universitario.

Moodle registra en sus *logs* una gran variedad de información. Para este proyecto nos centraremos únicamente en la información que Moodle recopila relacionada con el proceso de enseñanza-aprendizaje, como puede ser:

- Registro de actividad del usuario. En estos *logs* se registra la actividad de los usuarios, como cuándo acceden a la plataforma, cuándo ven o

completan módulos de cursos, cuándo participan en foros de discusión y cuándo envían tareas o cuestionarios.

- Registro de calificaciones en tareas y cuestionarios.
- Registro de mensajes y comunicación. Los *logs* pueden registrar la comunicación entre usuarios, como mensajes internos, mensajes de foros de discusión y chat en vivo.

3.3. ETL

ETL (*extract, transform, load*) [1] es un proceso de tres fases en el que los datos se extraen, se transforman y se cargan en un contenedor de datos de salida. Los datos pueden obtenerse de una o varias fuentes y enviarse a uno o varios destinos.

- **Extracción.** En esta primera fase se extraen los datos en bruto de diferentes fuentes. En este proyecto los datos provienen principalmente de Moodle aunque son extraídos usando diferentes técnicas (servicios web y *web scraping*).
- **Transformación.** En esta fase se aplican una serie de transformaciones para preparar y limpiar los datos antes de ser cargados para su uso. En este proyecto se realizarán transformaciones como la anonimización de datos personales, obtención del género del usuario, obtención de información de los *logs* de eventos, etc.
- **Carga.** En esta última fase se cargan los datos en un almacén o destino final. En este proyecto ese destino final son los nodos de KNIME, que almacenan la información ya transformada y facilitan los datos en forma de tabla a través de un puerto de salida.

3.4. Aprendizaje supervisado

El aprendizaje supervisado es un tipo de enfoque en el campo del aprendizaje automático, que es una rama de la inteligencia artificial. En el aprendizaje supervisado, un algoritmo o modelo se entrena utilizando un conjunto de datos etiquetados. Estos datos etiquetados consisten en ejemplos de entrada junto con las salidas o etiquetas deseadas. El objetivo del aprendizaje supervisado es generar un modelo que permita predecir las etiquetas de nuevas entradas de datos no etiquetadas.

De forma simplificada, podemos decir que el proceso de aprendizaje supervisado consta de las siguientes etapas [4]:

1. **División de datos para entrenamiento y pruebas.** Se recopila un conjunto de datos que contiene ejemplos de entrada junto con las etiquetas correctas. El conjunto de datos etiquetados se divide en dos subconjuntos principales: el conjunto de entrenamiento (*training set*) y el conjunto de prueba (*test set*).
2. **Entrenamiento del modelo.** Se utiliza el conjunto de datos de entrenamiento (generalmente sobre el 70-80 % de los datos) para entrenar el modelo de aprendizaje automático, en busca de patrones y relaciones en los datos que le permitan hacer predicciones precisas.
3. **Prueba y evaluación.** Una vez que el modelo está entrenado, se prueba con la parte de los datos etiquetados que no se usaron en el entrenamiento del modelo (conjunto de *test* o prueba). Esto permite evaluar la capacidad del modelo para hacer predicciones. En esta etapa se calcula la precisión del modelo, obtenida mediante la comparación de las predicciones del modelo con las etiquetas reales en los datos de prueba.
4. **Uso en la predicción.** Después de que el modelo ha sido entrenado y evaluado satisfactoriamente, puede utilizarse para hacer predicciones sobre nuevas entradas desconocidas o no etiquetadas.

Técnicas y herramientas

4.1. Entorno de desarrollo

En este apartado se describen las principales técnicas y herramientas utilizadas para el desarrollo del proyecto.

4.1.1. Eclipse

Eclipse¹ es un entorno de desarrollo integrado (IDE) ampliamente utilizado para programar aplicaciones en Java y otros lenguajes de programación. Eclipse permite gestionar proyectos Java, editar código con resaltado de sintaxis, depurar en tiempo real, etc.

Los nodos de KNIME se desarrollan en Java utilizando el IDE de programación Eclipse. KNIME nos facilita instrucciones de instalación del entorno de desarrollo con Eclipse para desarrollar nuestros nodos personalizados.

4.1.2. KNIME SDK

KNIME SDK² (*Software Development Kit*) es el conjunto de herramientas facilitadas por KNIME que nos permiten desarrollar extensiones y nodos personalizados que pueden integrarse en los flujos de trabajo de KNIME.

KNIME SDK se integra con Eclipse para desarrollar componentes de KNIME.

¹<https://www.eclipse.org/ide/>

²<https://github.com/knime/knime-sdk-setup>

4.1.3. Bitnami LMS Virtual Machine

Bitnami¹ es una empresa especializada en la creación de *stacks* de aplicaciones, que son entornos de aplicaciones preconfigurados para facilitar la instalación y puesta en marcha de una aplicación junto con todas sus dependencias.

Bitnami LMS² es una distribución que incluye el LMS Moodle y todas las aplicaciones necesarias para su ejecución (PHP, MariaDB, servidor web Apache, etc.). La distribución está disponible como instalación en la nube o como instalación en local con Docker, Kubernetes o Máquina Virtual.

En este proyecto se ha utilizado la versión en Máquina Virtual (formato OVA) correspondiente a Moodle 3.11.

4.1.4. VirtualBox

VirtualBox³ es un software de código abierto para la virtualización de sistemas, que permite ejecutar máquinas virtuales en cualquier sistema operativo. En este proyecto se ha utilizado VirtualBox 6.1 para ejecutar la máquina virtual facilitada por Bitnami LMS y disponer así de una instancia completamente funcional de Moodle en el equipo local.

4.1.5. Insomnia Rest

Insomnia Rest⁴ es una aplicación de escritorio que nos permite realizar llamadas REST a APIs de aplicaciones basadas en este sistema de comunicación. Una particularidad de Insomnia Rest es que permite realizar pruebas sobre aplicaciones en local, lo que nos ha permitido probar la comunicación con la API de Moodle de nuestra máquina virtual local, antes de incorporar el código a los nodos programados de KNIME.

4.1.6. GitHub

GitHub⁵ es una plataforma de desarrollo de software en la nube que utiliza Git como sistema de control de versiones. GitHub permite a los desarrolladores colaborar en proyectos de programación, alojando y gestionando

¹<https://bitnami.com/>

²<https://bitnami.com/stack/moodle>

³<https://www.virtualbox.org/>

⁴<https://insomnia.rest/>

⁵<https://github.com/>

sus repositorios de código fuente. En este proyecto se ha usado GitHub para almacenar el código fuente tanto de la aplicación desarrollada como de la memoria.

4.2. Desarrollo

En este apartado se describen los lenguajes de programación y librerías utilizados para el desarrollo del proyecto.

4.2.1. Java

Aunque desde la versión 4.6 de KNIME también se permite la implementación de nodos en Python, en este proyecto los nodos se han implementado en Java.

Java¹ es un lenguaje de programación muy extendido que se caracteriza por su programación orientada a objetos y su portabilidad. Las aplicaciones Java se compilan y pueden ser utilizadas en cualquier plataforma o sistema operativo que disponga de la Máquina Virtual Java (JVM).

4.2.2. Moodle API

Moodle cuenta con una REST API que permite la comunicación con el sistema a través de servicios web. El listado completo de servicios web de Moodle puede consultarse en este enlace (**Moodle API**²).

4.2.3. KNIME API

El código del núcleo de KNIME y de otras extensiones está disponible en el **repositorio de KNIME en GitHub**³ para poder ser reutilizado en nuevos desarrollos de extensiones de KNIME.

4.2.4. Bibliotecas externas

Algunas bibliotecas externas que se han incorporado al proyecto son:

4.2.4.1. UBUMonitor

UBUMonitor⁴ [11] es una herramienta implementada en Java que permite que los usuarios con rol de profesor puedan monitorización a sus alumnos en plataformas LSM Moodle. En este proyecto se han incorporado las clases de *login* y extracción de *logs* que implementa UBUMonitor.

¹<https://www.java.com/>

²https://docs.moodle.org/dev/Web_service_API_functions

³<https://github.com/knime/>

⁴<https://github.com/yjx0003/UBUMonitor>

4.2.4.2. Data Faker

Data Faker¹ es una librería para Java que permite crear datos falsos para pruebas en aplicaciones JVM. Se ha utilizado esta librería para anonimizar el nombre y apellidos de los alumnos.

¹<https://www.datafaker.net/>

4.3. Memoria

En el desarrollo de la Memoria del proyecto se han utilizado las siguientes herramientas:

4.3.1. Visual Studio Code

Visual Studio Code¹, también conocido como VS Code, es un editor de código fuente gratuito y de código abierto desarrollado por Microsoft. VS Code se puede extender fácilmente instalando extensiones adicionales, lo que permite que pueda ser utilizado para una amplia variedad de lenguajes de programación. En este proyecto se ha utilizado VS Code para trabajar con \LaTeX .

4.3.2. \LaTeX

\LaTeX ² es un sistema que permite crear documentos estructurados a partir de comandos y etiquetas. \LaTeX produce documentos con un formato profesional, lo que lo hace ideal para desarrollar tesis, artículos académicos, informes técnicos, etc.

¹<https://code.visualstudio.com/>

²<https://www.latex-project.org/>

Aspectos relevantes del desarrollo del proyecto

En este apartado se recogen los aspectos más interesantes del desarrollo del proyecto que se han detectado durante las fases de análisis, diseño e implementación. Se incluye también un apartado con un *workflow* que sirve de ejemplo de uso de los nodos desarrollados.

5.1. Análisis

Como resultado de la fase de análisis se han completado los apartados:

- Plan de proyecto software (Apéndice [A](#))
- Especificación de requisitos (Apéndice [B](#))

Durante la fase de análisis se realiza un estudio previo para determinar el alcance del proyecto y las posibilidades y limitaciones de las herramientas y tecnologías involucradas. También se instala el entorno de trabajo completo y se realizan las pruebas técnicas necesarias para asegurar que se podrán cumplir los objetivos del proyecto. Concretamente, se abordan las siguientes cuestiones:

- Extracción de datos de Moodle. Pruebas de acceso a la API de Moodle y los diferentes *logs* disponibles.
- Estudio previo del desarrollo en KNIME, incluyendo también la implementación de nodo KNIME de prueba y creación de un *workflow*

para ejecutar el nodo desarrollado en combinación con otros nodos del núcleo de KNIME.

- Elección de los datos de estudio y *workflow* a implementar.

5.1.1. Extracción de datos de Moodle

Se realiza un primer estudio de los métodos de acceso a los datos de Moodle. Aunque Moodle tiene una REST API muy completa para acceder a diferente tipo de información a través de servicios web, en este punto se detecta que los servicios web de Moodle no ofrecen acceso a todos los datos que se pueden visualizar desde Moodle.

Buscando métodos de acceso alternativos a Moodle, se localiza la aplicación UBUMonitor, un proyecto desarrollado en Java que sí extrae de Moodle todos los *logs* que nos gustaría incorporar en nuestro proyecto. El código de UBUMonitor nos aporta dos soluciones importantes:

- Código de acceso a Moodle mediante *Web Scraping*, simulando el acceso como si se tratara de un *login* realizado en el sitio web Moodle.
- Código de acceso a los *logs*.

Se decide incorporar directamente a nuestro proyecto las clases relevantes de UBUMonitor, generando el archivo JAR que nos permite importar el código al proyecto.

En resumen, como solución adoptada, se realizará un **doble *login* en Moodle**. El primer *login* se realiza siguiendo el modelo de webservice de la APP de Moodle. Esto implica que **la plataforma Moodle debe tener activa la opción de acceso a la APP de Moodle**. El segundo *login* se realiza a través de *Web Scraping* con las clases aportadas por UBUMonitor. Este doble acceso es transparente para el usuario.

5.1.1.1. Género del usuario

Moodle no almacena información de género del usuario. Para poder realizar estudios teniendo en cuenta el género, se ha utilizado el servicio externo genderize.io¹, que nos indica, a través de su API, el género de una persona a partir de su nombre (*male* o *female*).

¹<https://genderize.io/>

5.1.1.2. Anonimización

Se desea que el profesor que extrae los datos de Moodle pueda tener la información de los usuarios sin anonimizar, ya que los estudios que realice pueden ir orientados a la gestión directa de su acción formativa, lo que requeriría conocer el nombre de los alumnos afectados.

Sin embargo, también se quiere aportar la posibilidad de que el profesor utilice sus datos o resultados para sus proyectos o artículos de investigación, por lo que se quiere facilitar cierto grado de anonimización. Se incorpora al proyecto una anonimización sencilla cambiando el nombre y apellidos del usuario con el servicio **Data Faker**¹. Primero se obtiene el género del usuario y posteriormente se genera un nombre falso apropiado para ese género.

Por complejidad, se escapa del alcance de este proyecto un sistema más seguro de anonimización de los datos, que ofusque también la información relacionada con el usuario como, por ejemplo, su ID. En cualquier caso, KNIME dispone de funcionalidades adicionales de anonimización que se podrían incorporar como parte del flujo de trabajo.

5.1.2. Estudio previo del desarrollo en KNIME

5.1.2.1. Pasos para crear una extensión y nodo de ejemplo en KNIME

Los nodos de KNIME son elementos que aportan una funcionalidad independiente. Estos nodos se pueden combinar dentro de un flujo de trabajo o *workflow* para realizar funciones más complejas. Los nodos vienen empaquetados dentro de extensiones de KNIME, que pueden incluir uno o varios nodos relacionados.

Los pasos para crear una extensión en KNIME se pueden resumir en:

1. Montar el entorno de desarrollo con KNIME SDK y Eclipse.
2. Crear un nuevo proyecto de tipo KNIME Extension.
3. Implementar la extensión.
4. Probar la extensión.
5. Desplegar la extensión.

¹<https://www.datafaker.net/>

KNIME nos facilita una [guía para desarrollar en Java una extensión con un primer nodo de ejemplo](#)¹, llamado **Number Formatter**. Este nodo recibe como entrada una tabla de datos con una columna de números con decimales (del tipo Double de Java), y devuelve una tabla similar con los datos redondeados al número de decimales especificados en la configuración del nodo.

En este punto se estudia el código generado por KNIME para este nodo de ejemplo, además de otros nodos de ejemplo similares a los que queremos desarrollar.

5.1.2.2. Reutilización de nodos en KNIME a nivel de código

Durante la fase de análisis se realizan pruebas para verificar si un nodo de KNIME puede ejecutarse a nivel de código desde dentro de otro nodo. Esto podría ser interesante para reutilizar ciertas funcionalidades que están disponibles en KNIME a través de nodos, como puede ser conversión de datos de unos tipos a otros.

Aunque se podría pensar que es posible embeber o reutilizar la funcionalidad de un nodo desde programación como si se tratara de una función (parámetros de entrada) y recuperar la salida, se descubre que los nodos no se pueden ejecutar a nivel de código. Esto es, no podemos inyectar la funcionalidad completa de un nodo dentro de otro, ejecutándolo internamente. Se confirma a partir de comunicaciones con miembros del equipo de KNIME, que se ha impuesto esta restricción de diseño para evitar la alta dependencia entre nodos y los problemas que ello conllevaría a nivel de mantenimiento cuando un nodo queda obsoleto. Se puede consultar [este hilo del foro de soporte de KNIME](#)² donde se confirma esta restricción impuesta en el desarrollo.

5.1.2.3. Estudio de otros nodos

Las extensiones de KNIME están disponibles con código abierto. De entre las extensiones disponibles en *KNIME Community Hub*, la extensión *KNIME Twitter Connectors*³ dispone de un conjunto de nodos con una funcionalidad parecida a la que queremos implementar, por lo que se revisa el código de implementación como guía para nuestra implementación. En

¹https://docs.knime.com/latest/analytics_platform_new_node_quickstart_guide/index.html#_introduction

²<https://forum.knime.com/t/using-node-without-gui/2044/8>

³<https://hub.knime.com/knime/extensions/org.knime.features.ext.twitter/latest>

general, para futuros desarrolladores de KNIME, se aconseja seguir este procedimiento, localizando nodos de funcionalidad o estructura similar a la que se quiera desarrollar.

5.1.2.4. Reutilización de librerías externas

Al tratarse de un proyecto Java, se pueden importar en el proyecto librerías externas en formato JAR. Las aplicaciones UBUMonitor y Datafaker han sido importadas por esta vía.

5.1.3. Datos de estudio y *workflows* a implementar

Durante el análisis se evalúan las posibles fuentes de datos y posibles *workflows* de trabajo. Para el desarrollo se pueden utilizar los datos de prueba disponibles en la [demo de Moodle Mount Orange School](https://school.moodledemo.net/)¹, que es un Moodle ya montado con cursos, actividades y usuarios.

Se decide implementar un único *workflow* sobre datos reales que utilice diferentes técnicas de aprendizaje supervisado.

¹<https://school.moodledemo.net/>

5.2. Diseño

Como resultado de la fase de diseño se ha completado la Especificación de diseño, que se puede consultar en el Apéndice **C** de esta memoria.

5.2.1. Aspectos relevantes del diseño

A la hora de decidir el Diseño de la extensión y tipos de nodos que incluirá, se evalúan las siguientes estrategias:

1. Crear nodos individuales por cada *web service* de Moodle o pieza de información a extraer. Por ejemplo, un nodo para extraer los *logs* de participación en foros, otro nodo para extraer los *logs* de consulta de actividades, etc.
2. Crear nodos más genéricos que extraen información de varios *web services* relacionados. Por ejemplo, un nodo para extraer toda la información de usuarios, otro para cursos, otro para *logs*, etc.

Se opta por la segunda opción y se plantea el diseño de estos 6 nodos:

- **Moodle Connector.** Establece la conexión con la plataforma.
- **Moodle Courses.** Devuelve información de cursos disponibles en la plataforma.
- **Moodle Users.** Devuelve el listado de usuarios matriculados en los cursos especificados.
- **Moodle Reports Logs.** Devuelve un amplio abanico de *logs* de interacción de los usuarios disponibles en Moodle.
- **Moodle Reports Grades.** Devuelve calificaciones de los alumnos en las actividades indicadas.
- **Moodle Reports Quizzes.** Devuelve información relacionada con los cuestionarios realizados por los alumnos.

5.3. Implementación

Como resultado de la fase de implementación se han completado los apartados:

- Documentación técnica de programación (Apéndice **D**)
- Documentación de usuario (Apéndice **E**)

5.3.1. Problemas encontrados durante la implementación

Se destacan en este apartado algunos de los problemas más relevantes encontrados durante la implementación de la extensión.

5.3.1.1. Versión de KNIME

El desarrollo del proyecto comenzó en la versión KNIME 4.5.2. Durante el desarrollo, se lanzó la versión 4.7 y se actualizó el código y el entorno de desarrollo para que la extensión desarrollara fuera compatible con esta versión.

Antes de la entrega del proyecto se ha publicado una nueva versión de KNIME, 5.1, que no es del todo compatible con los nodos anteriores y la extensión desarrollada no se ha actualizado a esta versión. Sí que se ha comprobado que sigue siendo compatible con la última versión 4.7.x lanzada, que es la versión 4.7.7 (agosto 2023).

5.3.1.2. Entorno de desarrollo

En el apéndice **D** se describen los pasos requeridos para montar en entorno de desarrollo.

Como apunte, es importante establecer una versión fija de KNIME en *Target definition* para descargar los *plugins* correspondientes. Por defecto KNIME establece la versión «*nightly*», con lo que el entorno se irá actualizando automáticamente a la última versión disponible. Esto podría, ocasionalmente, «romper» nuestro entorno de trabajo al producirse un salto de versión mayor (por ejemplo, de 4.6 a 4.7, de 5.1 a 5.2, etc.). Como ya hemos comentado, nosotros trabajaremos en la versión 4.7.7.

5.3.1.3. Integración de bibliotecas externas

Ha sido necesario incorporar algunos proyectos externos, como:

- [UBUMonitor 2.10.2 \(20220426\)](#)¹
- [Datafaker 1.7](#)²
- [JSON in Java \(20220320\)](#)³

Se puede consultar el listado completo de bibliotecas externas en la carpeta `/libraries` de la extensión.

Inicialmente se intentaron utilizar directamente las clases de *logs* de UBUMonitor, pero se encontraron problemas de dependencias que obligaron a extraer las clases directamente a nuestro proyecto. La base del código es similar pero se ha modificado posteriormente para adaptar el código a los *logs* específicos que se incorporaron al proyecto.

5.3.1.4. Login de Moodle

Como ya se comentó previamente, el acceso a los datos de Moodle está limitado, por lo que se ha tenido que utilizar un doble acceso vía *web service* similar al utilizado por la APP de Moodle, combinado con el acceso por *Web Scraping* aportado por UBUMonitor. Esto implica que la implementación de Moodle debe tener activa la opción de acceso de la APP, algo que ya es muy habitual en las plataformas universitarias.

5.3.1.5. Despliegue de la extensión desarrollada

En la [guía para crear extensiones de KNIME en Java \(apartado *Deploy your extension*\)](#)⁴, se explican dos métodos para desplegar la extensión desarrollada de forma que pueda ser importada en cualquier instancia de KNIME sin necesidad de integración con Eclipse:

- **Opción 1. Sitio de actualización local (*Local Update Site*).** Esta es la opción recomendada y la que hemos seguido en este proyecto.

¹<https://github.com/yjx0003/UBUMonitor/releases>

²<https://www.datafaker.net/releases/1.7.0/>

³<https://github.com/stleary/JSON-java>

⁴https://docs.knime.com/latest/analytics_platform_new_node_quickstart_guide/index.html#_deploy_your_extension

Se crea a partir de la extensión desarrollada una «*feature*» que se añade a un sitio de actualización local. La carpeta generada, que se ha compartido en el repositorio del proyecto, se puede utilizar para importar el plugin desde cualquier instancia de KNIME. Las instrucciones de instalación se pueden consultar en el Manual de usuario de esta memoria (apéndice [E](#)).

- **Opción 2. *dropin*.** Se trata de exportar la extensión en formato JAR y añadirla en la carpeta **dropin** de la instalación de KNIME. Aunque no está correctamente documentado, esta opción dejó de funcionar en la versión KNIME 4.1 y se recuperó posteriormente en la versión 4.2. Sin embargo, nunca ha sido un método recomendado porque Eclipse tenía intención de eliminarlo. **En las últimas versiones de KNIME probadas (4.5 y 4.7) esta funcionalidad parece no estar disponible.**

5.4. Workflow de KNIME con datos reales

A continuación presentamos un estudio práctico utilizando la extensión desarrollada dentro de un *workflow* de KNIME aplicado al aprendizaje supervisado.

5.4.1. Datos de estudio

Los datos se corresponden con copias de seguridad de Moodle de la asignatura «Aprendizaje no supervisado» que se imparte en este máster dentro del bloque de Ciencia de Datos. Concretamente se disponen de los datos correspondientes a los cursos 2021/2022 y 2022/2023. Las copias de seguridad son completas, por lo que incluyen todas las interacciones de los alumnos y sus calificaciones parciales y finales. Aunque se disponía también de los datos del curso 2018/2019, finalmente se descartaron porque el curso había cambiado significativamente a nivel estructural, y las actividades y su evaluación no se correspondían con los datos de años posteriores.

Los datos se han importado en una instancia de Moodle local para su acceso desde KNIME con el rol de profesor.

En total disponemos de registros de 56 estudiantes, 26 matriculados en el curso de 2021/2022 y 30 en el de 2022/2023.

5.4.2. Objetivo del estudio

Para entender el siguiente estudio necesitamos conocer la metodología de estudio y evaluación utilizada dentro de la asignatura «Aprendizaje no supervisado».

La asignatura se divide en cuatro unidades (UD1, UD2, UD3 y UD4), con evaluación teórica y práctica. La evaluación teórica consiste en un cuestionario final por bloque (C1, C2, C3 y C4), donde el estudiante tiene un máximo de 3 intentos limitados en tiempo. La calificación final de cada cuestionario será la calificación máxima obtenida en los intentos realizados.

Para preparar estos cuestionarios finales, en cada unidad se dispone de una serie de documentos en PDF y vídeos con **cuestionarios de auto-evaluación** asociados. Estos cuestionarios de autoevaluación no computan para la calificación final y tienen intentos ilimitados. Cada cuestionario solo está disponible cuando se completa y supera el cuestionario inmediatamente anterior, según la calificación mínima requerida en cada cuestionario. Si

el estudiante desea mejorar el resultado, incluso si el cuestionario ya está superado, puede volver a visualizar el vídeo correspondiente y realizar nuevamente el cuestionario. Hay que recalcar que tanto la visualización de los vídeos como la realización de los cuestionarios de autoevaluación es opcional.

Para este estudio se desea implementar un **modelo que sea capaz de predecir la calificación final en los cuestionarios de evaluación (Suspendido, Aprobado, Notable o Sobresaliente), a partir de información de interacción de los usuarios con los recursos de la plataforma y de los resultados obtenidos en los cuestionarios de autoevaluación**. Concretamente extraeremos los siguientes campos:

- **videosviews**. Número total de visualizaciones de vídeos.
- **courseviews**. Número total de visualizaciones de recursos del curso.
- **forumcount**. Número total de visualizaciones de mensajes del foro.
- **autoevaltotalattempts**. Número total de intentos en cuestionarios de autoevaluación.
- **autoevalgrademean**. Calificación media obtenida en los cuestionarios de autoevaluación.

Siendo la clase a predecir:

- **evalgradecategory**. Calificación final obtenida en los cuestionarios de evaluación (Suspendido, Aprobado, Notable, Sobresaliente).

5.4.3. Recopilación, limpieza y preparación de los datos

En la Figura 5.3 se muestra la parte del *workflow* relativa a la recopilación, limpieza y preparación de los datos. De la extensión desarrollada en este proyecto, necesitamos los nodos [Moodle Connector](#), [Moodle Users](#), [Moodle Reports Logs](#) y [Moodle Reports Quizzes](#). Para este caso de estudio no será necesario el nodo [Moodle Reports Grades](#), ya que las calificaciones que necesitamos están relacionadas con cuestionarios y ya están disponibles a través del nodo [Moodle Reports Quizzes](#).

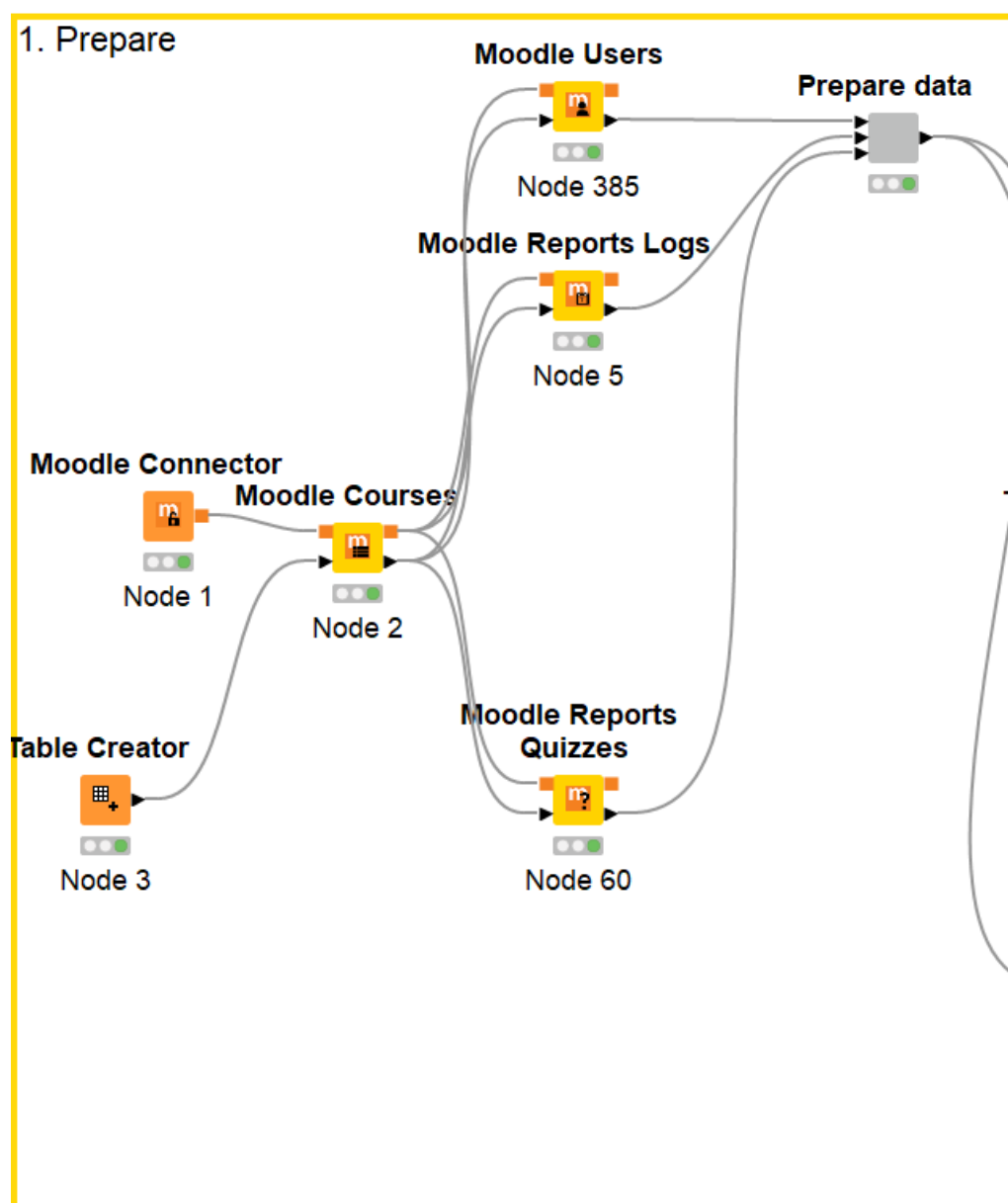


Figura 5.3: *Workflow*: preparación de datos.

Si desplegamos el componente **Prepare data**, veremos componentes adicionales para la extracción y tratamiento de datos específicos (Figura 5.4):

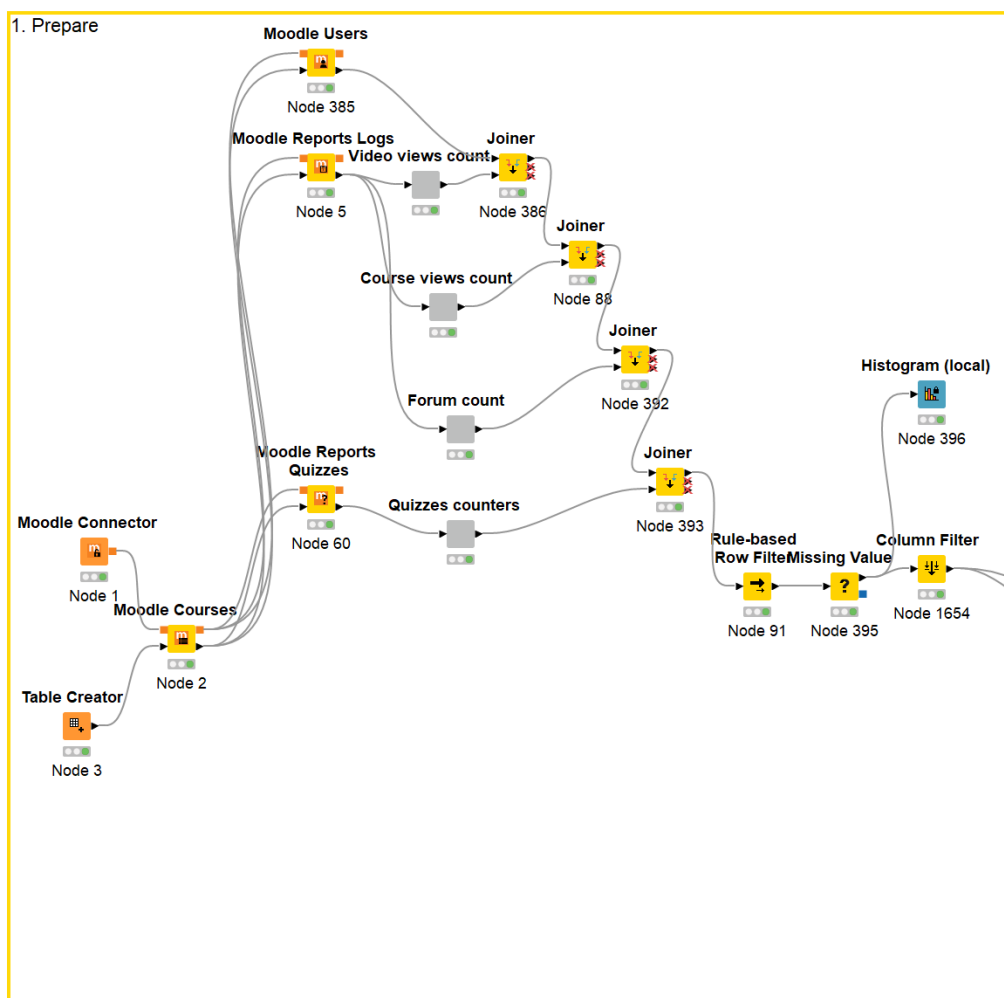


Figura 5.4: Workflow: preparación de datos (Prepare data desplegado).

El componente **Video View Count** 5.5 obtiene los datos de log desde el nodo **Moodle Reports Logs** y filtra aquellos *logs* relacionados con la visualización de vídeos. Utilizando un nodo **GroupBy** realiza un conteo agrupando por curso y usuario, devolviendo el campo **videosviews**.

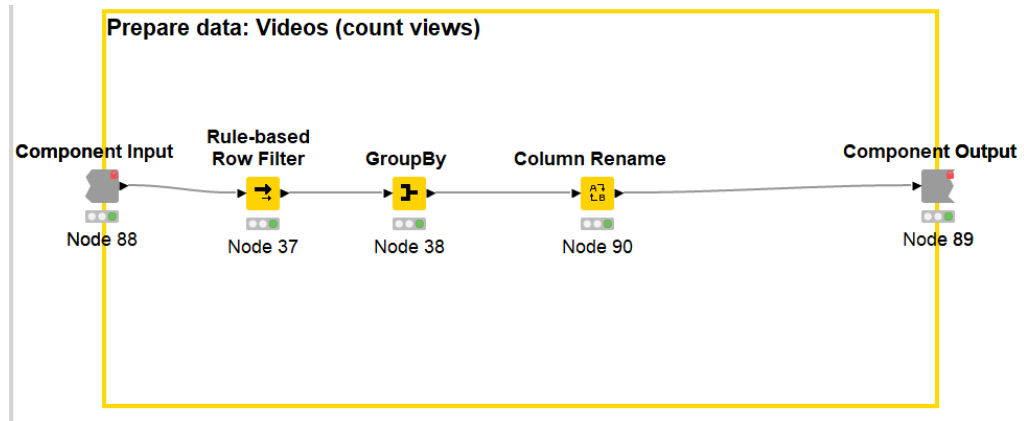


Figura 5.5: *Workflow*: componente Video views count.

Los componentes **Course views count** y **Forum count** funcionan de forma similar al anterior pero filtrando por los *logs* correspondientes a cada caso y devolviendo los campos **courseviews** y **forumcount** respectivamente.

El componente **Quizzes counters** obtiene, a partir del nodo **Moodle Reports Quizzies**, información de los intentos de cuestionarios realizados por los usuarios y agrupa los resultados para obtener los otros campos que necesitamos **autoevaltotalattempts**, **autoevalgrademean** y la clase, **evalgradecategory**.

Se han realizado las siguientes operaciones de preparación y limpieza de los datos:

- Los valores nulos en calificaciones se han considerado 0.
- Se han eliminado los usuarios que no han realizado ningún cuestionario de evaluación.

Tras la limpieza de datos, nos hemos quedado con un conjunto de 53 registros. En la Figura 5.6 se muestra un fragmento de la tabla de salida con las variables necesarias para el estudio.

▲ Filtered table - 8:1654 - Column Filter

File Edit Hilite Navigation View

Table "default" - Rows: 53 Spec - Columns: 6 Properties Flow Variables

Row ID	I videosviews	I courseviews	I forumcount	I autoevaltotalattempts	D autoevalgrademean	S evalgradecategory
Row49_Row2...	6	88	1	18	5.424	Suspendido
Row50_Row3...	16	120	85	26	7.424	Aprobado
Row51_Row4...	33	253	159	44	7.709	Sobresaliente
Row52_Row5...	36	164	53	52	6.83	Notable
Row53_Row6...	40	364	110	76	7.131	Sobresaliente
Row54_Row8...	42	303	118	40	7.853	Sobresaliente
Row55_Row9...	54	423	151	60	7.552	Notable
Row56_Row1...	21	206	91	19	5.787	Aprobado
Row57_Row1...	26	391	120	51	8.357	Sobresaliente
Row58_Row1...	43	211	124	55	8.392	Sobresaliente
Row59_Row1...	37	478	215	51	8.815	Sobresaliente
Row60_Row1...	69	139	67	51	6.959	Sobresaliente
Row61_Row1...	31	131	46	31	8.178	Sobresaliente
Row62_Row1...	31	184	80	63	6.13	Notable

Figura 5.6: *Workflow*: preparación de datos (salida).

Como la clase es una variable cualitativa nominal, mostramos un diagrama de barras (Figura 5.7) con la frecuencia de cada valor posible de calificación (Suspendido, Aprobado, Notable y Sobresaliente).

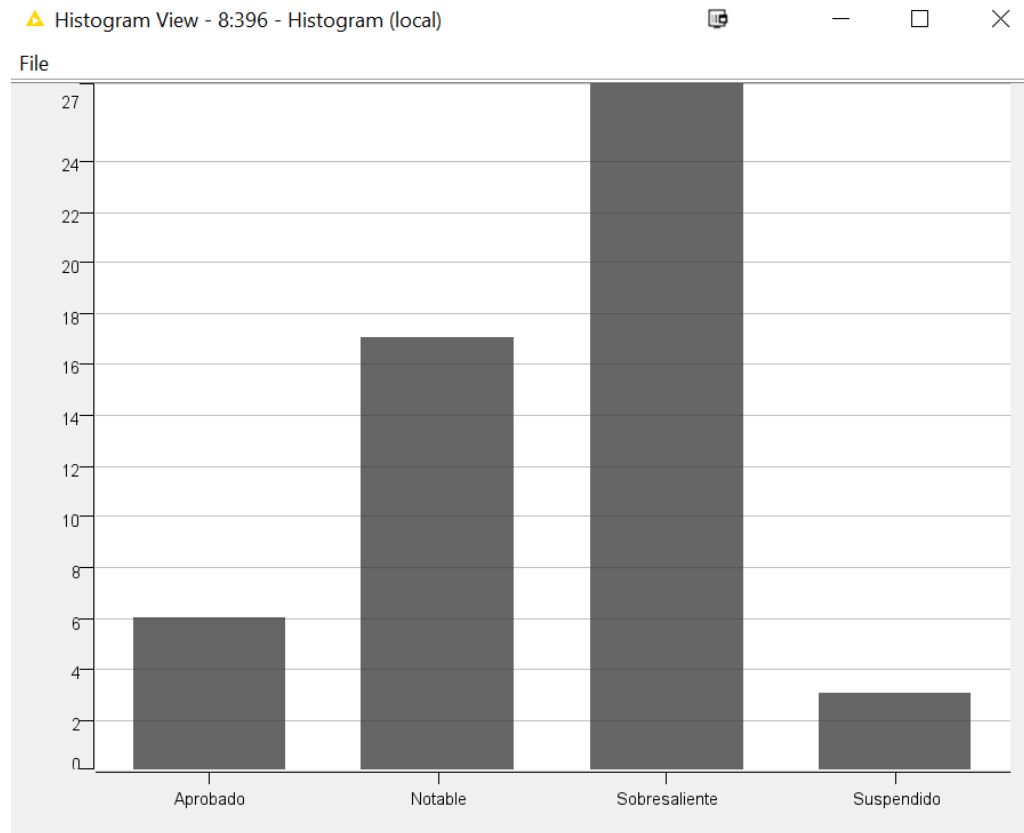


Figura 5.7: *Workflow*: preparación de datos (diagrama de barras clase).

5.4.4. Modelo de aprendizaje

En la Figura 5.8 se muestra el *workflow* completo. Se han implementando tres variantes del modelo con el mismo objetivo, identificadas como Modelo A, Modelo B y Modelo C.

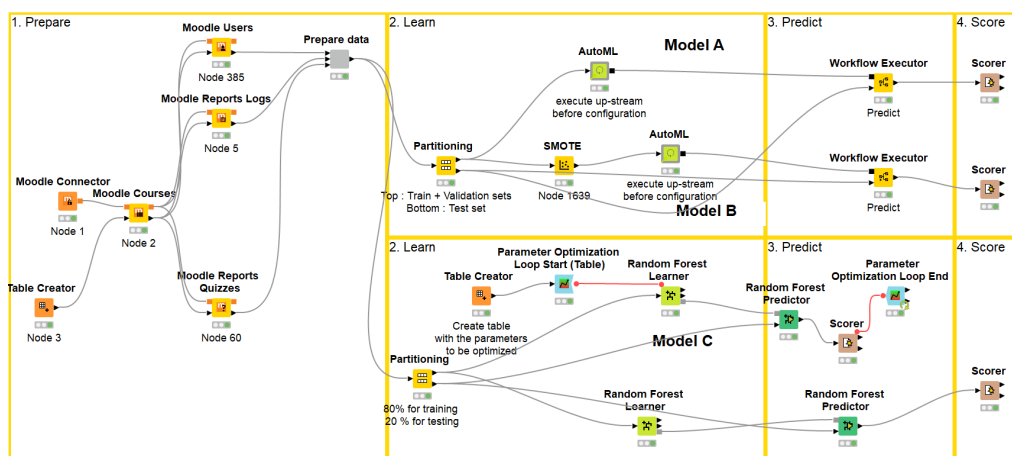


Figura 5.8: Workflow completo.

5.4.4.1. Modelo A

En el modelo A, mostrado en la Figura 5.9 utilizamos el componente [AutoML](#) de KNIME. Este componente entrena automáticamente modelos de aprendizaje automático supervisados para clasificación binaria o multiclase, siendo este segundo nuestro objetivo.

El componente incluye y automatiza todo el ciclo de *Machine Learning*, incluyendo preparación de datos, optimización de parámetros con validación cruzada y evaluación y selección del mejor modelo.

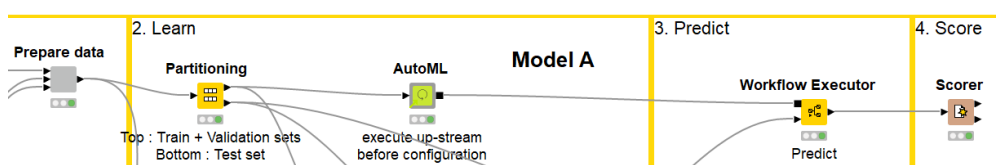


Figura 5.9: Modelo A: Workflow.

No tenemos que ver [AutoML](#) como una caja cerrada que lo hace todo por nosotros. Lo interesante está en el hecho de que [AutoML](#) se ha construido como un componente y no como un nodo, con lo que realmente se trata de un *workflow* más complejo que se ha agrupado como componente. Este componente puede ser expandido y podemos visualizar sus elementos y modificarlos según las necesidades del proyecto.

El componente viene por defecto enlazado con el repositorio de KNIME. Esto permite, por ejemplo, actualizarlo cuando se publican nuevas versiones. Cuando el componente está enlazado al repositorio, podemos expandirlo y ver los elementos que lo componen, pero no podemos modificar su contenido. Para desenlazar el componente y, con ello, desbloquearlo, accederemos al menú contextual (clic con el botón derecho sobre el componente), y seleccionaremos **Component** ▷ **Disconnect Link**.

En la Figura 5.10 se muestra el primer nivel de expansión del componente, que a su vez está formado por otros componetes que también se pueden abrir.

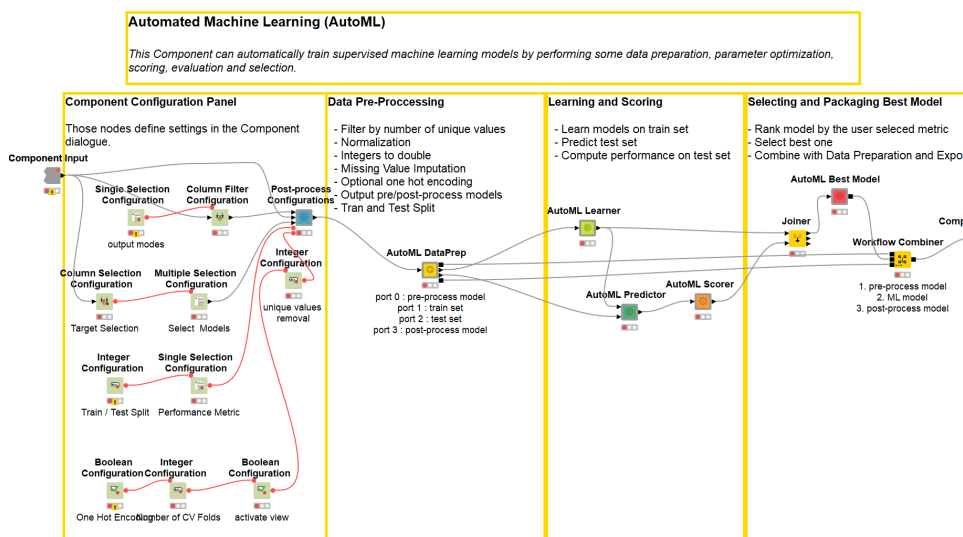
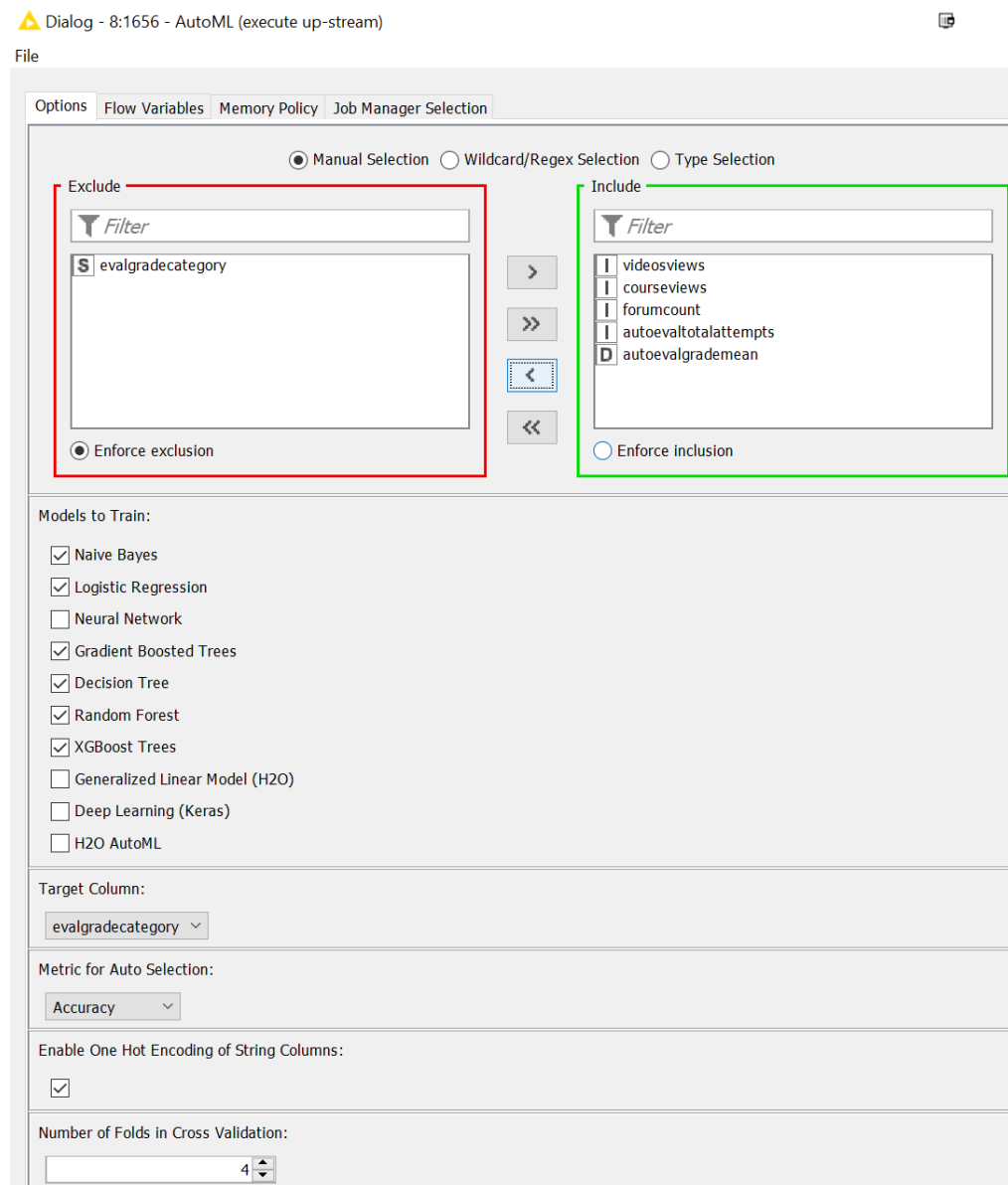


Figura 5.10: Modelo A: Componente **AutoML** expandido.

En la Figura 5.11 se muestra la configuración del componente **AutoML** para nuestro modelo, donde hemos seleccionado los modelos a entrenar:

- Naïve Bayes
- Logistic Regression
- Gradient Boosted Trees
- Decision Tree
- Random Forest
- XGBoost Trees

Figura 5.11: Modelo A: Componente **AutoML** configuración.

Como resultado de la ejecución, **AutoML** nos muestra un listado de los modelos ordenados por *Accuracy* (Figura 5.12). Los modelos *Gradient Boosted Trees*, *Logistic Regression*, *Decision Tree* y *Random Forest* obtienen los valores más altos (*Accuracy* = 0.67)

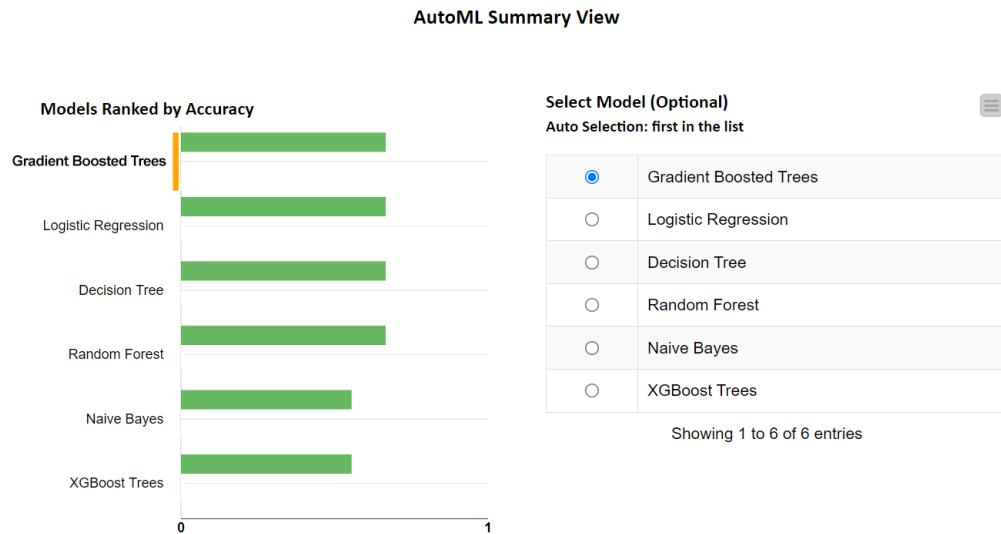


Figura 5.12: Modelo A: Resultados (verde → *Accuracy*).

5.4.4.2. Modelo B

El modelo B (Figura 5.13) es una variante del anterior, usando también *AutoML*, pero añadiendo previamente sobremuestreo a los datos de entrada para enriquecer los datos de entrenamiento e igualar la distribución de clases. Esta técnica se llama *SMOTE* [7] (*Synthetic Minority Over-sampling Technique*) y KNIME dispone de un nodo que la implementa y que hemos configurado para que realice un sobremuestreo de las clases minoritarias. Esta técnica ha sido utilizada en otros estudios similares de predicción de calificaciones de estudiantes [8].

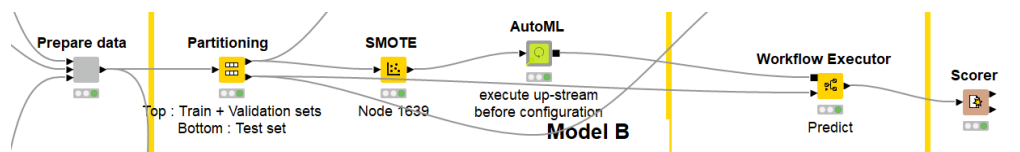


Figura 5.13: Modelo B: *Workflow*.

La distribución de clases ahora se muestra en la Figura 5.14

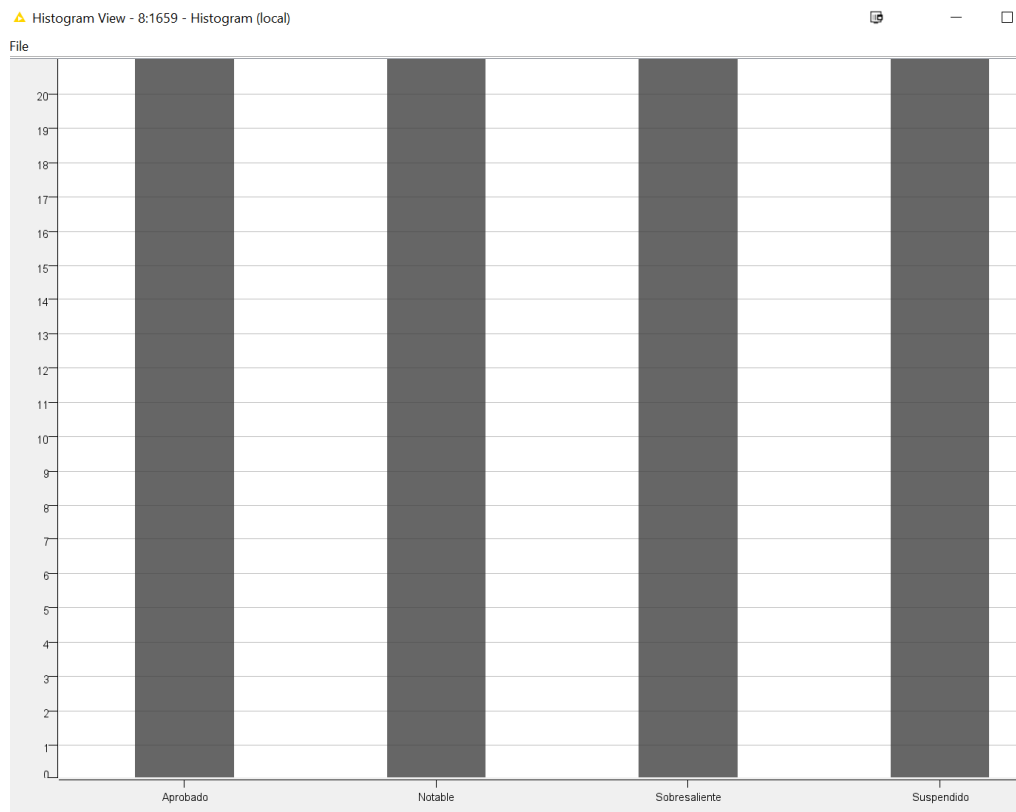


Figura 5.14: Modelo B: **SMOTE**.

Como resultado, vemos que se obtiene un valor mayor de *Accuracy* (0.76) en todos los modelos a excepción de *Naïve Bayes*.

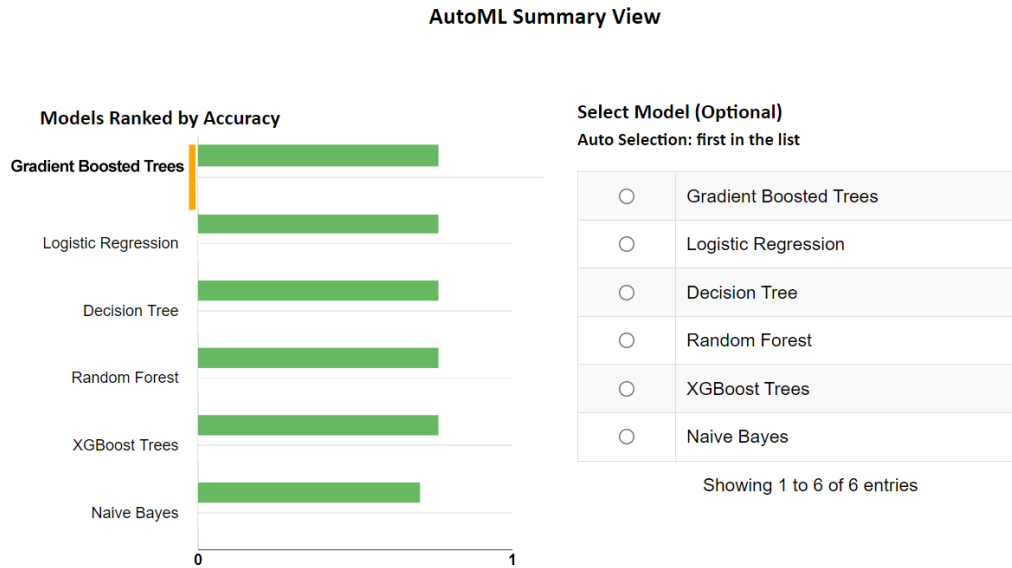


Figura 5.15: Modelo B: Resultados (verde \rightarrow *Accuracy*).

5.4.4.3. Modelo C

En modelo C (Figura 5.16) se entrena un modelo *Random Forest* con optimización de parámetros.

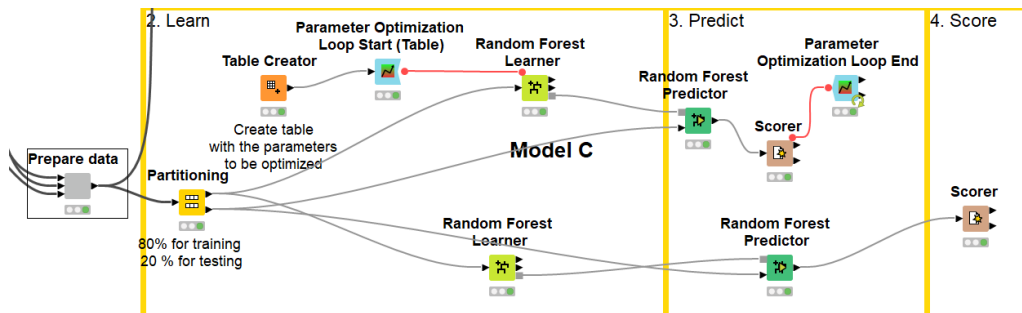


Figura 5.16: Modelo C: *Workflow*.

En la parte superior se observa el flujo desarrollado para ejecutar el modelo con varios parámetros y seleccionar la combinación con la que se obtiene un mejor valor de *Accuracy* (Figura 5.17).

▲ Best parameters - 8:82 - Parameter Optimization Loop End

File Edit Hilite Navigation View

Table "default" - Rows: 1 Spec - Columns: 3 Properties Flow Variables			
Row ID	depth	nmodels	Objective value
Best parameters	8	391	0.818

Figura 5.17: Modelo C: *Best parameters*.

Estos parámetros se utilizarán para configurar el modelo y obtener el modelo final. En la Figura 5.18 se muestra el valor de *Accuracy* de este modelo (0.818) y la tabla de confusión correspondiente.

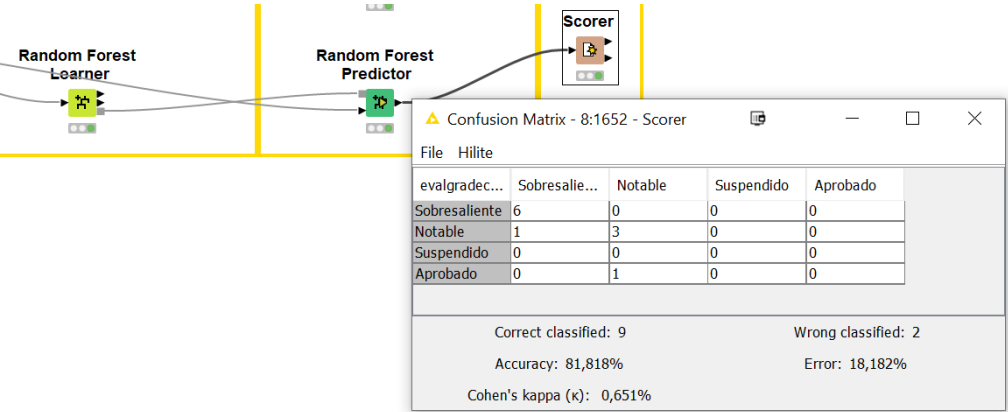


Figura 5.18: Modelo C: Tabla de confusión.

Trabajos relacionados

6.1. UBU Monitor

UBUMonitor es una aplicación de escritorio que permite la conexión a una plataforma Moodle para monitorizar la actividad de los alumnos. UBU-Monitor permite extraer los datos de los *logs* y las calificaciones obtenidas en las actividades, presentando los datos con herramientas de visualización que lo hacen muy atractivo. Además, añade modelos de aprendizaje automático para monitorizar el riesgo de abandono de los alumnos matriculados [5] [11].

Nuestro proyecto comparte mucho con UBUMonitor en cuando al tipo de información con la que trabaja y perfil de usuario al que va destinado, y nos ha resultado muy útil como referencia e incluso reutilización de algunos componentes.

Sin embargo, nuestra aproximación utilizando el ecosistema de KNIME, intenta ser más abierta y flexible, dejando que sea el usuario final el que pueda decidir qué tipo de modelos implementar o utilizar con los datos extraídos de sus asignaturas.

Conclusiones y Líneas de trabajo futuras

7.1. Conclusiones

Los objetivos generales planteados al inicio del proyecto se han podido alcanzar a la finalización del mismo, con lo que se ha conseguido desarrollar una extensión de KNIME muy completa que permite extraer datos de Moodle e incorporarlos en flujos de trabajo más complejos para estudios relacionados con la Ciencia de Datos.

A nivel técnico se ha requerido un profundo estudio del lenguaje de programación Java, no solo orientado al desarrollo específico de extensiones de KNIME, sino más amplio por haberse requerido integración con otras aplicaciones externas. También se ha ahondado en la arquitectura de KNIME, tanto a nivel de núcleo como a nivel de estructura de extensiones y nodos.

También se han podido cumplir los objetivos personales planteados, ampliando los conocimientos adquiridos en el máster y aplicándolos a nuevas herramientas como KNIME y a datos de estudio de mi interés personal y de investigación, relacionados con la formación *online*.

Sin duda, considero que se ha realizado un trabajo amplio y con mucho interés personal en la materia de estudio. Gracias a una tutorización continua por parte de los tutores, el proyecto ha estado muy bien organizado desde su inicio y se ha podido entregar una extensión de KNIME totalmente funcional, pero al mismo tiempo abierta para servir de base para futuras ampliaciones.

7.2. Líneas de trabajo futuras

Debido al alcance limitado del proyecto, se han quedado fuera algunas posibles mejoras y ampliaciones, que se exponen a continuación:

7.2.1. Desarrollo de *plugins* de Moodle para extraer información adicional

En este proyecto solo se ha contemplado la extracción de datos a través de las herramientas que facilita Moodle. Una integración más ambiciosa entre KNIME y Moodle podría conllevar el desarrollo de *plugins* de Moodle para extraer información más específica que no se puede extraer por defecto.

Esta línea requiere no pensar únicamente en el rol de profesor, sino en la intervención por parte de las instituciones educativas para facilitar la integración y el acceso a datos adicionales.

7.2.2. Colección de Flujos de trabajo de KNIME para Moodle

KNIME permite crear flujos de trabajo y compartirlos, de forma que sea fácil su reutilización. Se propone en esta línea crear un set de *workflows* ya preparados para su utilización con la extensión de Moodle desarrollada.

7.2.3. KNIME en servidor

Los flujos de trabajo estudiados en este proyecto se basan en la ejecución local de una instancia de KNIME. La propuesta es utilizar KNIME en *cloud* o servidor para flujos de trabajo que actúan de forma permanente. Por ejemplo, un flujo que revisa constantemente la evolución de un curso para determinar si algún alumno está en riesgo de abandono.

7.2.4. Comunicación bidireccional entre KNIME y Moodle

En este proyecto solo se ha contemplado la extracción de datos desde Moodle hacia KNIME. Moodle permite, a través de determinados servicios web ya incluidos o a través de servicios web personalizados, realizar acciones en el aula. Sería interesante que el resultado de la ejecución de un *workflow* de KNIME pudiera realizar acciones a partir de los resultados obtenidos.

Por ejemplo, enviar un correo a los alumnos clasificados como en riesgo de abandono o publicar un mensaje de aviso en el foro del aula.

7.2.5. Datos Abiertos (Open Data)

Conseguir que las instituciones educativas involucradas en el proyecto faciliten datos abiertos de forma automática con una política previa de anonimización (Open Data). Aunque existen intentos de publicar información basados en la información estrictamente legal (portal de transparencia) y datos preprocesados con estadísticas finales, con el auge de la Ciencia de Datos, se hace cada vez más necesario el acceso a datos en bruto para análisis más variados.

7.2.6. Anonimización

Realizar una anonimización ajustada a la política de anonimización de la institución y que cubra la anonimización de identificadores de usuarios.

7.2.7. Obtener género

Mejorar la API incluyendo varios proveedores externos.

7.2.8. Desvincular de librerías externas

Eliminar las dependencias con las librerías externas. Por ejemplo, eliminar el código dependiente de UBUMonitor desarrollando soluciones propias integradas en la extensión.

7.2.9. Actualización

Actualización a la última versión de KNIME, actualmente la 5.x. Definir un plan de mantenimiento y actualizaciones a versiones futuras. Se debe comprobar si alguna de las librerías y componentes utilizados en nuestro proyecto han quedado obsoletas y requieren ser sustituidas.

7.2.10. Ampliación de funcionalidad

Ampliar funcionalidad extrayendo más información disponible en Moodle. Por ejemplo, la versión actual está limitada en los tipos de logs que se extraen.

7.2.11. Ampliar pruebas

Las pruebas que se han llevado a cabo han estado muy dirigidas hacia un tipo de cursos y una plataforma Moodle concreta. Se considera necesario ampliar las pruebas a otros entornos de Moodle y otros tipos de acciones formativas.

7.2.12. Compartir extensión

Compartir la extensión en el *Community* HUB para que esté disponible para otros usuarios de la comunidad.

Apéndice

Apéndice A

Plan de Proyecto Software

A.1. Metodología

Durante el desarrollo de este proyecto se ha utilizado una versión simplificada de la metodología Scrum. Concretamente se han seguido ciclos o *sprints* de dos semanas, apoyados por reuniones de coordinación con los tutores del proyecto. Tras cada *sprint* se han mostrado y discutido los avances realizados y se han planificado las tareas a realizar en el siguiente *sprint*.

Aunque durante el desarrollo del proyecto se han producido retrasos que han llevado a dilatar la entrega final del proyecto, el orden de ejecución de los *sprints* y el alcance del proyecto se han mantenido según la planificación inicial, que se detalla a continuación.

A.2. Planificación temporal

En este apartado se describen las tareas llevadas a cabo en cada *sprint*.

A.2.1. *Sprint* 0

Sprint de arranque del proyecto dedicado al análisis del problema y el alcance de la solución.

- Preparación del entorno de desarrollo para programar extensiones de KNIME.
- Desarrollo de nodo de prueba para conocer la arquitectura de KNIME.

Fechas: desde 02/02/2022 a 15/02/2022.

A.2.2. *Sprint 1*

En este *sprint* continuamos con la fase de análisis:

- Preparación de plataforma Moodle local con datos de prueba.
- Pruebas de acceso a la información de Moodle a través de *web services* y *web scraping*.
- Estudio de la aplicación UBUMonitor, tanto a nivel de usuario como a nivel de código, para evaluar su posible reutilización.
- Se define el alcance y objetivos del proyecto.
- Memoria: se prepara el documento base de memoria y se completan los apartados de Introducción, Objetivos y Análisis.

Fechas: desde 27/04/2022 a 10/05/2022.

A.2.3. *Sprint 2*

Sprint dedicado al diseño de algunos de los módulos propuestos durante la fase de análisis.

- Diseño de nodo [Moodle Connection](#).
- Diseño de nodo [Moodle Courses](#).
- Diseño de nodo [Moodle Users](#).
- Memoria: documentación parcial de los apartados de Técnicas y herramientas y Diseño.

Fechas: desde 11/05/2022 a 24/05/2022.

A.2.4. *Sprint 3*

Implementación parcial de los nodos diseñados en el *sprint* anterior. En este *sprint* los nodos implementados son operativos pero no se ha añadido aún configuración.

- Implementación parcial del nodo [Moodle Connection](#).
- Implementación parcial del nodo [Moodle Courses](#).
- Implementación parcial del nodo [Moodle Users](#).
- *Workflow* básicos para probar los nodos implementados con datos de prueba de Moodle.
- Memoria: documentación parcial del apartado de Implementación.

Fechas: desde 25/05/2022 a 07/06/2022.

A.2.5. *Sprint 4*

Ampliación de los nodos implementados en el *sprint* anterior añadiendo configuración.

- Implementación de configuración del nodo [Moodle Connection](#).
- Implementación de configuración del nodo [Moodle Courses](#).
- Implementación de configuración del nodo [Moodle Users](#).
- Memoria: actualización del apartado de Implementación.

Fechas: desde 08/06/2022 a 21/06/2022.

A.2.6. *Sprint 5*

Sprint dedicado al diseño de otros módulos propuestos durante la fase de análisis.

- Diseño del nodo Moodle Reports Logs y primeras pruebas de implementación.
- Memoria: actualización del apartado de Diseño.

Fechas: desde 22/06/2022 a 05/07/2022.

A.2.7. *Sprint 6*

Sprint dedicado a la implementación del nodo [Moodle Reports Logs](#).

- Implementación del nodo [Moodle Reports Logs](#).
- Mapeo de *logs* de Moodle para extraer variables de los textos de *log*.
- Memoria: actualización del apartado de Implementación.

Fechas: desde 06/07/2022 a 07/08/2022.

A.2.8. *Sprint 7*

Sprint dedicado al diseño de otros módulos propuestos durante la fase de análisis.

- Diseño del nodo [Moodle Reports Grades](#) y primeras pruebas de implementación.
- Memoria: actualización del apartado de Diseño.

Fechas: desde 25/01/2023 a 06/02/2023.

A.2.9. *Sprint 8*

Sprint dedicado a la implementación del nodo [Moodle Reports Grades](#).

- Implementación del nodo [Moodle Reports Grades](#).
- Memoria: actualización del apartado de Implementación.

Fechas: desde 23/08/2023 a 05/09/2023.

A.2.10. *Sprint 9*

Sprint dedicado a la ampliación del nodo [Moodle Users](#).

- Incorporación de anonimización de usuarios.
- Estimación de género de usuario.

- Memoria: actualización del apartado de Implementación.

Fechas: desde 06/09/2023 a 19/09/2023.

A.2.11. *Sprint 10*

Sprint dedicado a la preparación final del código y compilación de la solución final.

- Limpieza del código eliminando mensajes de depuración y añadiendo comentarios.
- Compilación final para generar el plugin que puede ser utilizado en cualquier *workflow* de KNIME.
- Memoria: actualización del apartado de Implementación.

Fechas: desde 20/09/2023 a 03/10/2023.

A.2.12. *Sprint 11*

Primer *sprint* dedicado al uso de los nodos implementados con datos reales.

- Importación de datos reales en Moodle local.
- Implementación de *workflows* básicos para probar el correcto acceso a datos reales.
- Memoria: actualización del apartado de *Workflows*.

Fechas: desde 04/10/2023 a 17/10/2023.

A.2.13. *Sprint 12*

Sprint dedicado a la implementación del *Workflow* con datos reales.

- Implementación del *workflow* con datos reales.
- Memoria: actualización del apartado de *Workflows*.

Fechas: desde 18/10/2023 a 31/10/2023.

A.2.14. *Sprint* 13

Sprint dedicado a la implementación del *Workflow* con datos reales (continuación).

- Implementación del *workflow* con datos reales.
- Memoria: actualización del apartado de *Workflows*.
- Memoria: revisión general.

Fechas: desde 02/11/2023 a 14/11/2023.

A.2.15. *Sprint* 14

Sprint dedicado a completar la memoria del proyecto.

- Memoria: actualización de apartados pendientes
- Memoria: corrección general según sugerencias reportadas en el *sprint* anterior.

Fechas: desde 15/11/2023 a 28/11/2023.

A.3. Estudio de viabilidad

A.3.1. Viabilidad económica

A.3.1.1. Costes de personal

Aunque el proyecto se ha estimado en 7 meses a tiempo parcial, para calcular el coste del proyecto, se considerará que se ha realizado por un desarrollador a tiempo completo en un período de 3,5 meses. Se considera el salario medio neto de un programador de 1.610 € mensuales [3]. (ver tabla A.1).

La cotización a la seguridad social se ha calculado como horas comunes, según el régimen general de 2023 (28,30 %) [2].

Concepto	Coste
Salario neto	1.610,00 €
Retención IRPF (17 %)	500,37 €
Seguridad social (28,30 %)	832,96 €
Salario bruto (mensual)	2.943,33 €
Total 3.5 meses	10.301,65

Tabla A.1: Costes de personal.

Además se sumará el sueldo de los dos tutores asignados al proyecto [9] durante 7 meses, ambos con la posición de Profesor Titular de Universidad. Se asignan 0,5 por tutor.

Sueldo mensual: 3.145,59 €. Imparte 24 créditos anuales.

$$\frac{3.145,59 \text{ €} \times 12 \text{ meses}}{24 \text{ créditos}} \times 0,5 \text{ créditos} \times 2 \text{ tutores} = 1.572,80 \text{ €}$$

El coste de los tutores corresponde a 1.572,80 €.

A.3.1.2. Costes de *hardware*

En esta sección se enumeran los costes del *hardware* usado durante el desarrollo.

Para el desarrollo se ha usado un equipo de sobremesa valorado en 1.200 €, con amortización en 4 años (ver tabla A.2).

$$\frac{1200 \text{ €}}{4 \text{ años} * 12 \text{ meses}} = 25$$

Concepto	Coste	Amortización
Ordenador sobremesa	1.200 €	25
Total 3,5 meses	87,5 €	

Tabla A.2: Costes de hardware.

A.3.1.3. Costes totales

En la tabla A.3 se agrupan todos los costes calculados del proyecto, dando el total de €.

Concepto	Coste
Personal	10.301,65 €
Tutores	1.572,80 €
<i>Hardware</i>	87,51 €
Total	11.961,96 €

Tabla A.3: Costes totales del proyecto.

A.3.2. Viabilidad legal

KNIME es una herramienta de software libre que puede ser descargada y utilizada gratuitamente bajo los términos de la Licencia Pública General de GNU versión 3 (GPLv3). Hay que tener en cuenta que, aunque la plataforma es gratuita, algunas extensiones de terceros pueden tener licencias específicas y restricciones de uso. En este proyecto se han utilizado extensiones adicionales de KNIME para definir los *workflows* de trabajo, pero no se han requerido estas extensiones dentro del código de la extensión y nodos desarrollados, por lo que solo se deben tener en cuenta los términos de la licencia GPLv3 del núcleo de KNIME.

Adicionalmente se ha incluido código de la aplicación externa UBUMonitor, distribuido bajo licencia MIT. Esta licencia permite, entre otras cosas, la modificación, distribución y uso comercial del código fuente, por lo que el código de UBUMonitor se puede incorporar a nuestro proyecto sin restricciones, pero a su vez sin garantías de ningún tipo.

Apéndice *B*

Especificación de requisitos

B.1. Introducción

En este apéndice se detallan sus requisitos, tanto funcionales como no funcionales.

B.2. Catálogo de requisitos

B.2.1. Requisitos funcionales

- **RF-1 Conexión:** la extensión debe poder conectarse con cualquier plataforma Moodle.
 - **RF-1.1 Conexión: Rol profesor:** el usuario debe tener el rol profesor en la plataforma.
 - **RF-1.2 Conexión: APP móvil:** la plataforma Moodle debe tener el acceso a la aplicación móvil activado.
- **RF-2 Consulta de cursos:** la debe ser capaz de extraer información de los cursos de la plataforma Moodle.
 - **RF-2.1 Filtro por identificador de curso:** la extensión debe poder extraer un curso específico, indicando su **id** dentro de la plataforma Moodle.
 - **RF-2.2 Filtro por identificador de categoría:** la extensión debe poder extraer todos los cursos de una categoría, indicando su **id** dentro de la plataforma Moodle.

- **RF-3 Consulta de usuarios:** la extensión debe de extraer información de los usuarios de la plataforma Moodle.
- - **RF-3.1 Filtro de usuarios por curso:** la extensión debe poder extraer los usuarios de un curso específico o de un listado de cursos.
 - **RF-3.2 Filtro de usuarios por rol:** la extensión debe poder filtrar solo por usuarios con el rol Estudiante.
 - **RF-3.3 Estimación del género del estudiante:** la extensión debe poder estimar el género del estudiante a partir de su nombre.
 - **RF-3.4 Anonimización de datos personales:** la extensión debe poder anonimizar los datos personales del usuario (nombre y apellidos).
- **RF-4 Consulta de logs:** la extensión debe ser capaz de extraer los logs de los cursos indicados.
- **RF-5 Consulta de calificaciones:** la extensión debe ser capaz de extraer las calificaciones de los usuarios registrados en los cursos indicados.
- **RF-6 Consulta de cuestionarios:** la extensión debe ser capaz de extraer información de los cuestionarios de los usuarios registrados en los cursos indicados.

B.2.2. Requisitos no funcionales

- **RNF-1 Usabilidad:** los nodos desarrollados deben ser intuitivos y fáciles de usar, siguiendo el diseño de otros nodos y extensiones de KNIME.
- **RNF-2 Mantenibilidad:** debe ser sencillo añadir funcionalidad adicional, siguiendo las recomendaciones de desarrollo de extensiones de KNIME.
- **RNF-3 Compatibilidad:** la extensión desarrollada debe ser compatible con la versión de KNIME 4.7.

Apéndice C

Especificación de diseño

C.1. Introducción

En este apéndice se describen los diseños de nodos que permiten cumplir con los objetivos generales del proyecto y los requisitos funcionales definidos previamente.

C.2. Diseño de la extensión KNIME Moodle Integration

Se ha diseñado la extensión KNIME Moodle Integration con los siguientes nodos individuales:

- **Moodle Connector**. Cubre los requisitos especificados en RF-1 (RF-1.1 y RF-1.2).
- **Moodle Courses**. Cubre los requisitos especificados en RF-2 (RF-2.1 y RF-2.2).
- **Moodle Users**. Cubre los requisitos especificados en RF-3 (RF-3.1, RF-3.2, RF-3.3 y RF-3.4).
- **Moodle Reports Logs**. Cubre los requisitos especificados en RF-4.
- **Moodle Reports Grades**. Cubre los requisitos especificados en RF-5.
- **Moodle Reports Quizzes**. Cubre los requisitos especificados en RF-6.

En el diseño se han tenido en cuenta también los requisitos no funcionales RNF-1, RNF-2 y RNF-3.

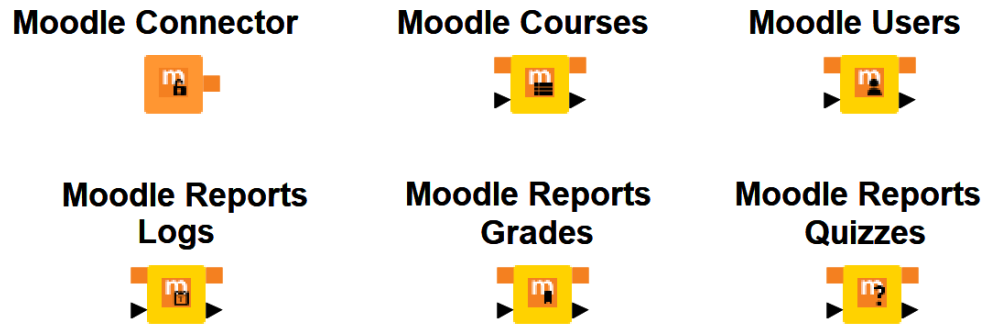


Figura C.2.1: Nodos implementados en la extensión [KNIME Moodle Integration](#).

Se describe a continuación el diseño de cada nodo implementado.

C.2.1. Diseño del nodo Moodle Connector

El nodo **Moodle Connector** establece la conexión con una plataforma Moodle. Requiere una cuenta con perfil de profesor (RF-1.1). Requiere que la plataforma Moodle tenga activado el acceso a la aplicación móvil (RF-1.1).

El nodo devuelve a través de un puerto de salida la información de sesión, de forma que, conectada a otros nodos de la extensión, estos pueden extraer información de Moodle sin necesidad de volver a conectarse.

El nodo tiene la siguiente **estructura**:

■ **Puertos de entrada:**

- Sin puertos de entrada.

■ **Puertos de salida:**

- **0. Moodle Courses.** Una conexión que se puede utilizar para acceder a la API de Moodle.

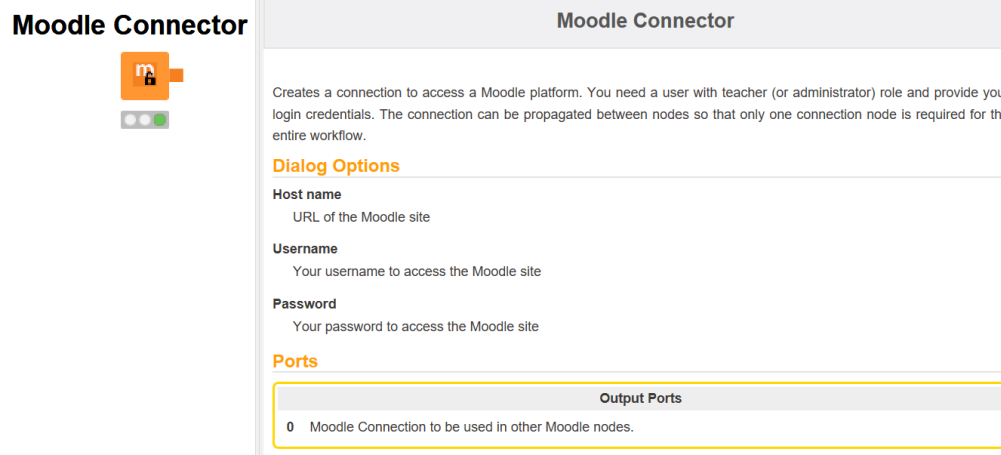


Figura C.2.2: Nodo **Moodle Connector**. Descripción.

El nodo permite la siguiente **configuración**:

- **Host name.** URL de la plataforma Moodle, empezando por http o https.

- **Username.** Nombre de usuario de acceso a Moodle.
- **Password.** Contraseña de acceso a Moodle.

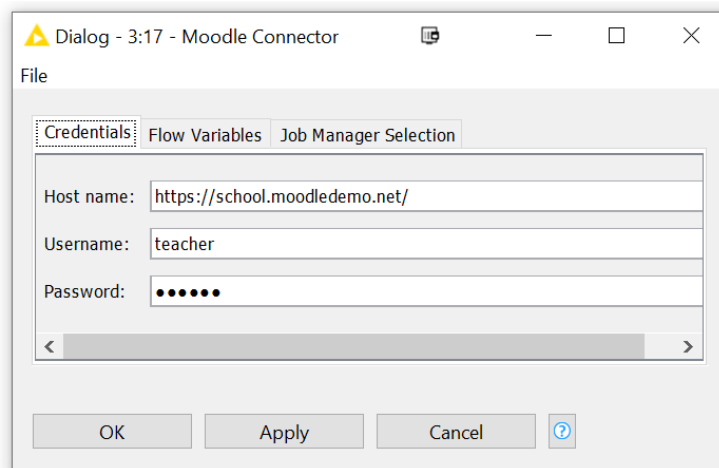


Figura C.2.3: Nodo Moodle Connector. Configuración.

C.2.2. Diseño del nodo Moodle Courses

El nodo **Moodle Courses** extrae información de cursos. Permite obtener uno o varios cursos según su identificador (RF-2.1) o todos los cursos dentro de las categorías especificadas (RF-2.2).

El nodo tiene la siguiente **estructura**:

■ **Puertos de entrada:**

- **0. Moodle Connection.** Conexión obtenida desde el nodo **Moodle Connector**.
- **1. Input table.** Tabla de entrada con información de filtrado. Columna con identificador de cursos y columna con identificador de categorías. Aunque el filtrado no es obligatorio, sí se requiere añadir un nodo que defina la tabla de entrada.

■ **Puertos de salida:**

- **0. Moodle Connection.** Devuelve la conexión para facilitar su transferencia al siguiente nodo del flujo de trabajo.
- **1. Output table.** Tabla de salida con la información de los cursos.

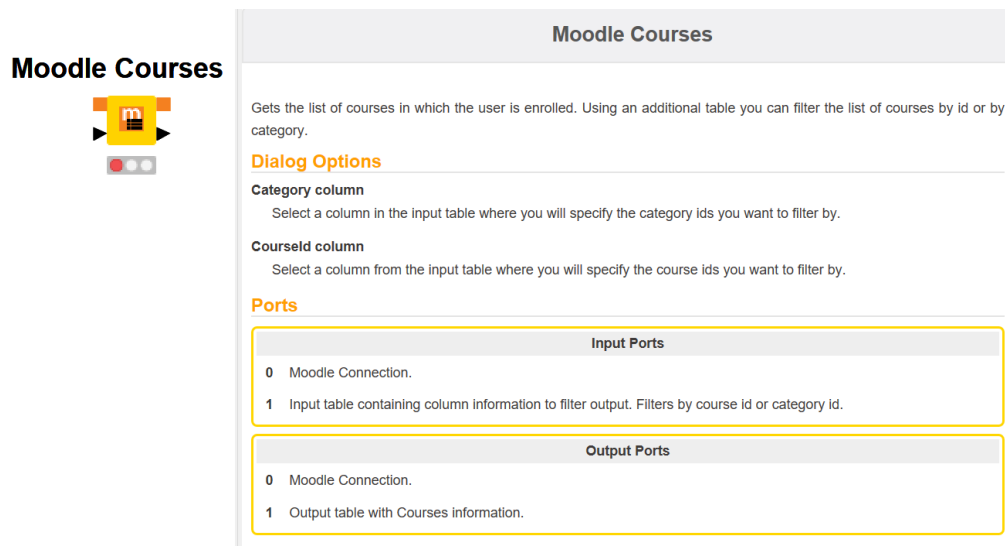


Figura C.2.4: Nodo **Moodle Courses**. Descripción.

El nodo permite la siguiente **configuración**:

- **Category column.** Permite seleccionar la columna en la que se encuentran las categorías por las que se desea filtrar. Es opcional.
- **CourseId column.** Permite seleccionar la columna en la que se encuentran los identificadores de cursos por los que se desea filtrar. Es opcional.

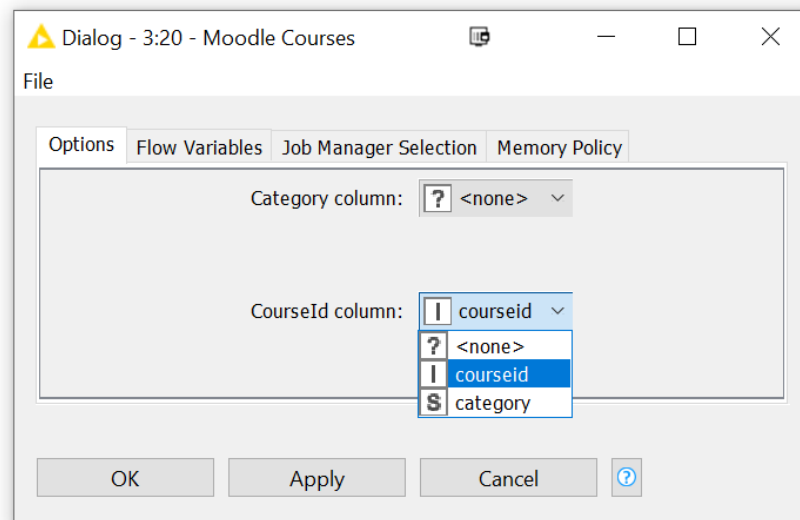


Figura C.2.5: Nodo Moodle Courses. Configuración.

En la tabla C.1 se muestra la estructura de los datos devueltos por cada curso encontrado en la plataforma Moodle:

Columna	Tipo	Descripción
courseid	integer	ID del curso
shortname	string	Descripción corta del curso
fullname	string	Nombre completo del curso
displayname	string	Nombre del curso que se muestra en el aula
enrolledusercount	string	Número de alumnos registrados en el curso
idnumber	string	ID interno del curso
visible	boolean	Indica si el curso es visible para los estudiantes (1)
summary	string	Resumen o descripción del curso
summaryformat	boolean	Indica si el resumen tiene formato de texto asociado (1)
format	string	Tipo de formato del resumen
category	string	ID de categoría
categoryname	string	Nombre de la categoría

Tabla C.1: Tabla de salida del nodo Moodle Courses.

Table-Out - 320 - Moodle Courses											
File Edit Hilitte Navigation View											
Table "default" - Rows: 18 Spec - Columns: 11 Properties Flow Variables											
Row ID	I courseid	S shortna...	S fullname	S displayname	I enrole...	S idnumber	I visible	S summary	I summaryformat	S format	I category
Row0	62	Psych Cine	Psychology in Cinema	Psychology in Cinema	13		1	<p>In this course we study... </p>	1	topics	2
Row1	66	Digital Liter...	Digital Literacy	Digital Literacy	11		1	<p>Introducing the concep...	1	topics	2
Row2	6	Cinema: Cla...	Class and Conflict i...	Class and Conflict i...	12		1	In this module we will anal...	1	weeks	2
Row3	63	Impressionists	The Impressionists	The Impressionists	10		1	Ages: 16+<p>An introduction to the ...<p><span class="label lab...	1	topics	2
Row4	68	Votes for W...	Votes for Women!	Votes for Women!	6		1		1	weeks	6
Row5	59	Celebrating ...	Celebrating Cultures	Celebrating Cultures	22		1	<p>Mount Orange is proud...	1	topics	6
Row6	56	History: Rus...	History: Russia in R...	History: Russia in R...	7		1	<p>This course is designe...	1	topics	6
Row7	35	World of Wa...	World of Water	World of Water	5		1	<p>We all need water! Let...	1	topics	6
Row8	49	Pilgrimage	Pilgrimage	Pilgrimage	1		1	<p>This course in Religio...	1	topics	6
Row9	70	LitFr 1100 - ...	Literature of France...	Literature of France...	12		1	<p dir="ltr" style="text-all...<p dir="ltr" style="text-all...	1	weeks	3

Figura C.2.6: Nodo Moodle Courses. Ejemplo de salida.

C.2.3. Diseño del nodo Moodle Users

El nodo **Moodle Users** extrae información de usuarios según los cursos especificados (RF-3.1). También es posible devolver solo los usuarios con rol estudiante (RF-3.2), estimar el género de cada usuario a partir de su nombre (RF-3.3) y anonimizar los datos personales (RF-3.4).

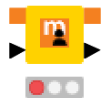
El nodo tiene la siguiente **estructura**:

■ **Puertos de entrada:**

- **0. Moodle Connection.** Conexión obtenida desde el nodo **Moodle Connector**.
- **1. Input table.** Tabla de entrada con un listado de identificadores de cursos.

■ **Puertos de salida:**

- **0. Moodle Connection.** Devuelve la conexión para facilitar su transferencia al siguiente nodo del flujo de trabajo.
- **1. Output table.** Tabla de salida con la información de los usuarios.

Moodle Users

Moodle Users

Returns the list of users registered in the course or courses indicated in the input table (output from Moodle Courses node).

Dialog Options

Only Students
Returns only users with the 'student' role.

Estimate gender
Obtains the user's gender from their first name.

Anonymize
Anonymizes user names using fake data.

Random Seed
Numerical value of the random seed. Use 0 if you do not want to apply a random seed. With a value other than 0, the same random results will always be generated for the user name.

Ports

Input Ports	
0	Moodle Connection.
1	Input table containing a list of valid Courses (output from Moodle Courses node)

Output Ports	
0	Moodle Connection.
1	Output table with Users information.

Figura C.2.7: Nodo **Moodle Users**. Descripción.

El nodo permite la siguiente **configuración**:

- **Only students.** Si se activa, devuelve únicamente los usuarios con rol estudiante.
- **Estimate gender.** Si se activa, devuelve una estimación del género del usuario (*male* o *female*).
- **Anonymize.** Si se activa, devuelve el nombre y apellidos del usuario anonimizados.
- **Random Seed.** Establece una semilla aleatoria para obtener un valor de nombre/apellidos anonimizado. Si se deja el valor 0, los nombres/apellidos de cada usuario cambiarán con cada ejecución del nodo.

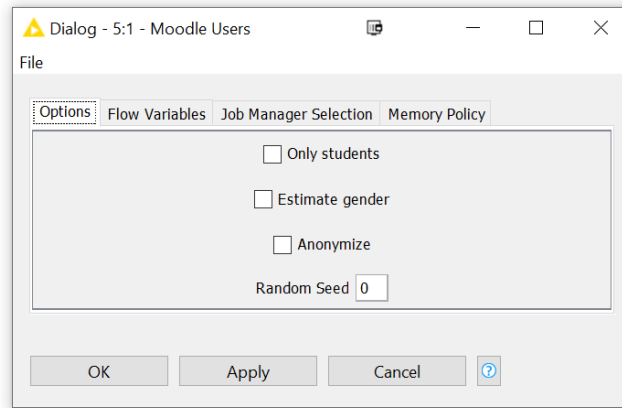


Figura C.2.8: Nodo Moodle Users. Configuración.

En la tabla C.2 se muestra la estructura de los datos devueltos por cada usuario:

Columna	Tipo	Descripción
courseid	integer	ID del curso en que el usuario está registrado
userid	integer	ID del usuario
fullname	string	Nombre completo del usuario. Puede ser el nombre real o el nombre Anonimizado, si se especifica en la configuración del nodo.
firstaccess	timestamp	Fecha de primer acceso al curso
lastcourseaccess	timestamp	Fecha de último acceso al curso
lastaccess	timestamp	Fecha de último acceso a la plataforma
roles	string	Listado de roles del usuario, separados por coma
country	string	Código del país del usuario
city	string	Ciudad del usuario
gender	string	Género estimado (<i>male</i> , <i>female</i>). Esta columna es opcional, en función de la configuración del nodo.

Tabla C.2: Tabla de salida del nodo Moodle Users.

Table-Out - 3:33 - Moodle Users

File Edit Hilite Navigation View

Table "default" - Rows: 20 Spec - Columns: 10 Properties Flow Variables

Row ID	I courseid	I userid	S fullname	I firstacc...	I lastcour...	I lastacc...	S roles	S country	S city	S gender
Row0	62	44	Marcos Crist	1387633312	1411659500	1543224188	student	AU	Orange City	male
Row1	62	45	Darcie Schoen	1285214320	1407934717	1407934713	student	AU	Orange City	female
Row2	62	46	Loreen Hessel	1276142535	1448277442	1543852453	student	AU	Orange City	female
Row3	62	47	Brandon Dooley	1387631743	1670083745	1670146669	student	AU	Orange City	male
Row4	62	48	Youlanda Collier	1276142128	1415619673	1621353194	student	AU	Orange City	female
Row5	62	49	Scot Reichert	1283320136	1432113767	1670068136	student	AU	Orange City	male
Row6	62	50	Ronny Medhurst	1405930757	1411659692	1411659692	student	AU	Orange City	male
Row7	62	51	Bridgette Kutch	1405932741	1408275842	1408275839	student	AU	Orange City	female
Row8	62	52	Bennie Feil	1387634640	1408351858	1543224055	student	AU	Orange City	male
Row9	62	53	Leo Rolfson	1405932102	1408276508	1487335105	student	AU	Orange City	male
Row10	62	56	Marilyn Will	1276140717	1675937516	1675937849	student	AU	Orange City	female
Row11	63	44	John Hodkiewicz	1387633312	0	1543224188	student	AU	Orange City	male
Row12	63	45	Jane Dicki	1285214320	0	1407934713	student	AU	Orange City	female
Row13	63	46	Jane Corkery	1276142535	0	1543852453	student	AU	Orange City	female
Row14	63	47	Ambrose Corwin	1387631743	1411635442	1670146669	student	AU	Orange City	male
Row15	63	48	Esther Tremblay	1276142128	0	1621353194	student	AU	Orange City	female
Row16	63	49	Darnell Haag	1283320136	0	1670068136	student	AU	Orange City	male
Row17	63	51	Jane Haag	1405932741	0	1408275839	student	AU	Orange City	female
Row18	63	53	John Cummerata	1405932102	0	1487335105	student	AU	Orange City	male
Row19	63	56	Jane Gutmann	1276140717	1487580568	1675937849	student	AU	Orange City	female

Figura C.2.9: Nodo Moodle Users. Salida de ejemplo.

C.2.4. Diseño del nodo Moodle Reports Logs

El nodo **Moodle Reports Logs** extrae información de *logs* de la plataforma Moodle (RF-4). Este nodo está diseñado para poder extraer logs de forma homogénea aunque provengan de diferentes fuentes dentro de Moodle (*Logs, Live logs, Activity Reports, Overview Statistics, Course Participation, Activity completion, Statistics, etc.*)

El nodo tiene la siguiente **estructura**:

■ **Puertos de entrada:**

- **0. Moodle Connection.** Conexión obtenida desde el nodo **Moodle Connector**.
- **1. Input table.** Tabla de entrada con un listado de identificadores de cursos.

■ **Puertos de salida:**

- **0. Moodle Connection.** Devuelve la conexión para facilitar su transferencia al siguiente nodo del flujo de trabajo.
- **1. Output table.** Tabla de salida con la información de los *logs* o eventos.

Moodle Reports Logs



Moodle Reports Logs

Returns the event logs registered by Moodle. The node receives as input the ids of the courses for which the event logs are to be obtained. The output table returns all logs, identifying the course id, the log date, the log identifier and other values such as the id of the user, item or module affected by the event.

Ports

Input Ports	
0	Moodle Connection.
1	Input table containing a list of valid Courses (output from Moodle Courses node)

Output Ports	
0	Moodle Connection.
1	Output table with event logs.

Figura C.2.10: Nodo **Moodle Reports Logs**. Descripción.

El nodo no tiene configuración específica.

En la tabla C.3 se muestra la estructura de los datos devueltos por cada evento del *log*:

Columna	Tipo	Descripción
courseid	integer	ID del curso
time	string	Fecha de registro del log en formato DD/M-M/YYYY - HH:MM:SS +TimeZone.
component	string	Nombre del componente o módulo de Moodle que registra el evento
eventName	string	Nombre de sistema del evento
origin	string	Origen (WEB, CLI, APP, etc.)
IPAddress	string	Dirección IP del usuario
course	string	ID del curso extraído del mensaje de log
user	string	ID del usuario extraído del mensaje de log
targetUser	string	ID del usuario objetivo extraído del mensaje de log
targetCourse	string	ID del curso objetivo extraído del mensaje de log
module	string	ID del módulo extraído del mensaje de log
section	string	ID de la sección extraído del mensaje de log
item	string	ID del ítem extraído del mensaje de log
description	string	Mensaje completo guardado en el log
descriptionMapped	boolean	Indica si el mensaje ha sido mapeado (1) o no (0)
label	string	Nombre del módulo (página, cuestionario, tarea, etc.)

Tabla C.3: Tabla de salida del nodo **Moodle Reports Logs**.

I	courseid	S	time	S	component	S	eventName	S	origin	S	IPAddress	T	course	I	user	T	targetUser	T	targetCourse	T	module	T	section	T	item	S	description
63	09/02/2023	...	logs	log_report_viewed	WEB	51.37.215.236	63	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	The user with id '13' viewed the log report for the course with id '63'.	
63	09/02/2023	...	logs	log_report_viewed	WEB	51.37.215.236	63	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	The user with id '13' viewed the log report for the course with id '63'.	
63	09/02/2023	...	system	course_viewed	WEB	79.226.183.22	63	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	The user with id '13' viewed the course with id '63'.	
63	09/02/2023	...	system	course_viewed	WEB	79.226.183.22	63	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	The user with id '13' viewed the course with id '63'.	
63	09/02/2023	...	logs	log_report_viewed	WEB	51.37.215.236	63	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	The user with id '13' viewed the log report for the course with id '63'.	
63	09/02/2023	...	logs	log_report_viewed	WEB	51.37.215.236	63	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	The user with id '13' viewed the log report for the course with id '63'.	
63	09/02/2023	...	system	course_viewed	WEB	84.54.120.82	63	56	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	The user with id '56' viewed the course with id '63'.	
63	09/02/2023	...	logs	log_report_viewed	WEB	51.37.215.236	63	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	The user with id '13' viewed the log report for the course with id '63'.	
63	09/02/2023	...	quiz	quiz_attempt_viewed	WEB	84.54.120.82	0	56	0	0	0	64	0	1	0	0	0	0	0	0	0	0	0	0	0	The user with id '56' has viewed page '1' of the attempt with id '64' b...	
63	09/02/2023	...	quiz	quiz_attempt_started	WEB	84.54.120.82	0	56	0	0	0	730	0	64	0	0	0	0	0	0	0	0	0	0	0	The user with id '56' has started the attempt with id '64' for the quiz ...	
63	09/02/2023	...	quiz	course_module_viewed	WEB	84.54.120.82	0	56	0	0	0	730	0	0	0	0	0	0	0	0	0	0	0	0	0	The user with id '56' viewed the 'quiz' activity with course module id '...	
63	09/02/2023	...	logs	log_report_viewed	WEB	5.254.23.36	63	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	The user with id '13' viewed the log report for the course with id '63'.	
63	09/02/2023	...	quiz	course_module_viewed	WEB	84.54.120.82	0	1	0	0	0	730	0	0	0	0	0	0	0	0	0	0	0	0	0	The user with id '1' viewed the 'quiz' activity with course module id '7'...	
63	09/02/2023	...	quiz	course_module_viewed	WEB	84.54.120.82	0	1	0	0	0	730	0	0	0	0	0	0	0	0	0	0	0	0	0	The user with id '1' viewed the 'quiz' activity with course module id '7'...	
63	09/02/2023	...	system	course_viewed	WEB	84.54.120.82	63	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	The user with id '1' viewed the course with id '63'.	
63	27/09/2021	...	system	course_viewed	WEB	2a00:23:b:a92:...	63	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	The user with id '13' viewed the course with id '63'.

Figura C.2.11: Nodo **Moodle Reports Logs**. Salida.

C.2.5. Diseño del nodo Moodle Reports Grades

El nodo **Moodle Reports Grades** extrae información de calificaciones agrupadas por curso, estudiante y actividad (RF-5). Adicionalmente, devuelve una columna con la calificación final obtenida.

El nodo tiene la siguiente **estructura**:

■ **Puertos de entrada:**

- **0. Moodle Connection.** Conexión obtenida desde el nodo **Moodle Connector**.
- **1. Input table.** Tabla de entrada con un listado de identificadores de cursos.

■ **Puertos de salida:**

- **0. Moodle Connection.** Devuelve la conexión para facilitar su transferencia al siguiente nodo del flujo de trabajo.
- **1. Output table.** Tabla de salida con la información de calificaciones.

Moodle Reports Grades



Moodle Reports Grades

Returns a table with grades by course, student and activity. It also returns an entry with the final grade obtained in the course.

Ports

Input Ports	
0	Moodle Connection.
1	Input table containing a list of valid Courses (output from Moodle Courses node)

Output Ports	
0	Moodle Connection.
1	Output table with grades.

Figura C.2.12: Nodo **Moodle Reports Grades**. Descripción.

El nodo no tiene configuración específica.

En la tabla C.4 se muestra la estructura de los datos devueltos por cada calificación, clasificada por curso/actividad/usuario:

Columna	Tipo	Descripción
courseid	integer	ID del curso en que está registrado el usuario
userid	integer	ID del usuario
gradeid	integer	ID de la calificación
gradename	string	Nombre de la actividad de calificación
gradetype	string	Tipo de calificación (mod, category, manual, course)
grademodule	string	Módulo de la actividad de calificación (assign, forum, quiz, etc.)
grademin	double	Valor mínimo de la calificación
grademax	double	Valor máximo de la calificación
graderaw	double	Calificación obtenida, sin formatear/procesar
gradeformatted	string	Calificación formateada
gradecategoryid	integer	ID de categoría de calificación

Tabla C.4: Tabla de salida del nodo **Moodle Reports Grades**.

I	courseid	I	userid	I	gradeid	S	gradename	S	gradetype	S	gradem...	D	grademin	D	grademax	D	graderaw	S	gradefo...	I	gradeca...
62	48	378					From Concept to Reality: Trauma and F...	mod	assign	0		100	0			0		-		86	
62	48	377					Course discussion	mod	forum	1		3	0			0		-		86	
62	48	379						category		0		100	0			0		-		0	
62	48	371					Group Project	mod	assign	0		100	0			0		-		87	
62	48	380						category		0		100	0			0		-		0	
62	48	376					Factual recall test	mod	quiz	0		10	6					<i class="ic...		88	
62	48	372					Dissertation: Fight club	mod	assign	0		100	0			0		-		88	
62	48	373					Dissertation: A Beautiful Mind	mod	assign	0		100	0			0		-		88	
62	48	374					Dissertation: Spider	mod	assign	0		100	0			0		-		88	
62	48	382					F2F Session	manual		1		4	3					Good		88	
62	48	393					Class presentation	manual		0		100	75					75.00		88	
62	48	394					Reflective Journal	mod	assign	0		10	8					8.00		88	
62	48	396					Commitment	mod	assign	1		3	2					Partial		88	
62	48	395					Participation	mod	assign	1		3	3					Full		88	
62	48	381						category		0		120	89					89.00		0	
62	48	390					Screening 1	manual		1		2	1					Absent		92	
62	48	391					Screening 2	manual		1		2	2					Present		92	
62	48	392					Screening 3	manual		1		2	2					Present		92	
62	48	399					Grammar help with your essays	mod	quiz	0		10	0					-		85	
62	48	370					Course Final	course		0		120	89					89.00		-1	

Figura C.2.13: Nodo **Moodle Reports Grades**. Salida.

C.2.6. Diseño del nodo Moodle Reports Quizzes

El nodo **Moodle Reports Quizzes** extrae información sobre los intentos realizados por los estudiantes en los cuestionarios (RF-6).

El nodo tiene la siguiente **estructura**:

- **Puertos de entrada:**
 - **0. Moodle Connection.** Conexión obtenida desde el nodo **Moodle Connector**.
 - **1. Input table.** Tabla de entrada con un listado de identificadores de cursos.
- **Puertos de salida:**
 - **0. Moodle Connection.** Devuelve la conexión para facilitar su transferencia al siguiente nodo del flujo de trabajo.
 - **1. Output table.** Tabla de salida con la información sobre los intentos realizados en cuestionarios.

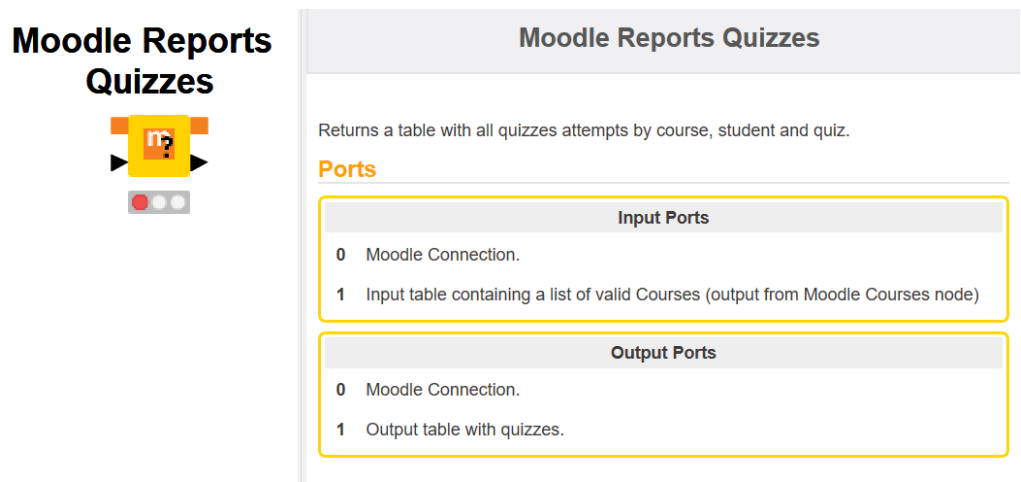


Figura C.2.14: Nodo **Moodle Reports Quizzes**. Descripción.

El nodo no tiene configuración específica.

En la tabla **C.5** se muestra la estructura de los datos devueltos por cada intento de realización de cuestionario:

Columna	Tipo	Descripción
courseid	integer	ID del curso en que está registrado el usuario
userid	integer	ID del usuario
coursemoduleid	integer	ID del módulo
quizid	integer	ID del cuestionario
quizname	string	Nombre del cuestionario
attemptid	integer	ID del intento de resolución del cuestionario
attemptnumber	integer	Número del intento de resolución del cuestionario
attemptduration	integer	Duración del intento en segundos
attemptgrade	double	Calificación obtenida en este intento

Tabla C.5: Tabla de salida del nodo [Moodle Reports Quizzes](#).

I	courseid	I	userid	I	cou...	I	quizid	S ▲	quizname	I	attemptid	I	attemptnumber	I	attemptduration	D	attemptgrade
680		10178		9143		277		?	Cuestionario para el vídeo «A Priori Algorithm»		11125		2		510		10
680		10189		9143		277		?	Cuestionario para el vídeo «A Priori Algorithm»		11126		1		1148		9.5
680		10193		9143		277		?	Cuestionario para el vídeo «A Priori Algorithm»		11127		1		953		8.64
680		10197		9143		277		?	Cuestionario para el vídeo «A Priori Algorithm»		11128		1		371		2.33
680		10197		9143		277		?	Cuestionario para el vídeo «A Priori Algorithm»		11129		2		163		2.67
680		10197		9143		277		?	Cuestionario para el vídeo «A Priori Algorithm»		11130		3		93		3.17
680		10197		9143		277		?	Cuestionario para el vídeo «A Priori Algorithm»		11131		4		317		9.5
680		10199		9143		277		?	Cuestionario para el vídeo «A Priori Algorithm»		11132		1		1433		10
680		10200		9143		277		?	Cuestionario para el vídeo «A Priori Algorithm»		11133		1		1825		8.4
680		10200		9143		277		?	Cuestionario para el vídeo «A Priori Algorithm»		11134		2		232		9.67
680		10200		9143		277		?	Cuestionario para el vídeo «A Priori Algorithm»		11135		3		148		10
680		10202		9143		277		?	Cuestionario para el vídeo «A Priori Algorithm»		11136		1		1002		10

Figura C.2.15: Nodo [Moodle Reports Quizzes](#). Salida.

Apéndice D

Documentación técnica de programación

D.1. Introducción

En este apéndice se describen los aspectos relevantes a nivel de programación para cualquier desarrollador interesado en continuar el proyecto o en desarrollar nuevas extensiones para la plataforma KNIME.

D.2. Estructura de directorios

A continuación se presenta la estructura principal de archivos y carpetas de la extensión knime-moodle-integration desarrollada. No se han incluido las carpetas genéricas de definición del proyecto en Eclipse, que se pueden consultar directamente en el repositorio del proyecto.

```
\libraries -- Librerías externas (jar)
\src.org.knime.moodle.
  .internal -- Librerías de uso interno
    .connection -- Librería de conexión utilizada por el nodo Moodle Connector
      MoodleConnection.java
      MoodleConnectionPortObject.java
      MoodleConnectionPortObjectSpec.java
      MoodleLogin.Java
    .logs -- Librería de Logs utilizada por el nodo Moodle Reports Logs
      Component.Java
```

```
Component.java
ComponentEvent.java
Event.java
LogDescription.java
LogLine.java
LogParameters.java
Logs.java
MoodleLogCreator.java
.nodes -- Nodos de la extensión
.connector -- Implementación del nodo Moodle Connector
    MoodleConnectorConfiguration.java
    MoodleConnectorNodeDialog.java
    MoodleConnectorNodeFactory.java
    MoodleConnectorNodeFactory.xml
    MoodleConnectorNodeModel.java
    MoodleConnectorNodePlugin.java
    MoodleConnectorNodeSettingsModel.java
    MoodleConnectorNodeView.java
    default.png
    moodle-connector.png
    package.html
.courses -- Implementación del nodo Moodle Courses
    MoodleCoursesNodeDialog.java
    MoodleCoursesNodeFactory.java
    MoodleCoursesNodeFactory.xml
    MoodleCoursesNodeModel.java
    MoodleCoursesNodeView.java
    default.png
    moodle-courses.png
    package.html
.reports
    .grades -- Implementación del nodo Moodle Reports Grades
        MoodleReportsGradesNodeDialog.java
        MoodleReportsGradesNodeFactory.java
        MoodleReportsGradesNodeFactory.xml
        MoodleReportsGradesNodeModel.java
        MoodleReportsGradesNodeView.java
        moodle-reports-grades.png
        package.html
    .logs -- Implementación del nodo Moodle Reports Logs
        MoodleReportsLogsNodeDialog.java
```



```
MoodleReportsLogsNodeFactory.java
MoodleReportsLogsNodeFactory.xml
MoodleReportsLogsNodeModel.java
MoodleReportsLogsNodeView.java
moodle-reports-logs.png
package.html
.users -- Implementación del nodo Moodle Users
MoodleUsersNodeDialog.java
MoodleUsersNodeFactory.java
MoodleUsersNodeFactory.xml
MoodleUsersNodeModel.java
MoodleUsersNodeView.java
moodle-users.png
package.html
```

D.2.1. Entorno de desarrollo con Eclipse y KNIME SDK

Los nodos de KNIME se desarrollan en Java utilizando el IDE de programación Eclipse. Aunque desde la versión 4.6 también se permite la implementación de nodos totalmente programados en Python, en este proyecto los nodos se han implementado en Java para la versión KNIME 4.7.x.

La documentación oficial de KNIME nos detalla paso a paso cómo preparar el entorno de trabajo necesario para desarrollar nuevos nodos en Java. Como primer paso debemos instalar Eclipse como IDE de desarrollo y KNIME SDK que nos proporciona las herramientas necesarias de desarrollo para compilar nuestra extensión y ejecutarla en KNIME. Se pueden consultar las instrucciones detalladas en este enlace <https://github.com/knime/knime-sdk-setup>.

Los pasos que tendremos que seguir son:

1. Instalación de Java 17 (OpenJDK 17). Esta versión puede variar para futuras versiones de KNIME.
2. Instalar Eclipse. Debemos instalar la versión «Eclipse for RCP and RAP Developers 2022-06», que es la que se corresponde a KNIME 4.7.x. Esta versión varía en función de la versión de KNIME en la que vayamos a trabajar.
3. Instalar Git.

4. Descargar KNIME SDK y configurar eclipse. Siguiendo las instrucciones correspondientes, indicaremos cuál es la plataforma objetivo activa, de forma que Eclipse pueda ejecutar KNIME y desplegar la extensión desarrollada.
5. Ejecutar KNIME desde KNIME Analytics Platform.launch ▷ Run as ▷ KNIME Analytics Platform.

D.2.2. Instalación y compilación de la extensión KNIME Moodle Integration

Por último, necesitamos incorporar la extensión [knime-moodle-integration](#) importando el proyecto desde el repositorio.

D.3. Desarrollo de nuevos nodos

Las instrucciones previas nos sirven para montar un entorno de desarrollo para trabajar con la extensión desarrollada en este proyecto. Si se desea implementar una nueva extensión, se recomienda seguir las instrucciones que KNIME nos facilita mediante la [implementación de un nodo de ejemplo](#)¹.

¹https://docs.knime.com/latest/analytics_platform_new_node_quickstart_guide/index.html#_introduction

Apéndice *E*

Documentación de usuario

E.1. Introducción

En la sección anterior se ha descrito cómo instalar y configurar el entorno de desarrollo con Eclipse y KNIME SDK. Si solo se desea utilizar la extensión desarrollada en un flujo de trabajo, sin ánimos de modificarla a nivel de código, podemos seguir las siguientes instrucciones de instalación.

E.2. Instalación KNIME

Como la extensión KNIME Moodle Integration se ha implementado en KNIME 4.7.7, instalaremos esta versión desde la sección [Previous Versions of KNIME](#)². KNIME está disponible para Windows, Linux y macOS.

E.3. Instalación de la extensión Moodle KNIME Integration

La extensión lista para ser utiliza en cualquier instalación de KNIME está disponible como pre-release en el repositorio del proyecto:

<https://github.com/frankgil/knime-moodle/releases/tag/extension>

Para utilizar esta extensión, debemos ejecutar KNIME y crear un Sitio de actualización local (Local Update Site). Seguiremos estos pasos:

²<https://www.knime.com/download-previous-versions>

En primer lugar descargaremos y descomprimiremos en una carpeta local la extensión.

Desde KNIME ▷ *Preferences* ▷ *Install/Update* ▷ *Available Software Sites* ▷ *Add*, añadiremos una ubicación local seleccionando la carpeta «org.knime.moodle.update», tal y como se muestra en la Figura E.3.1:

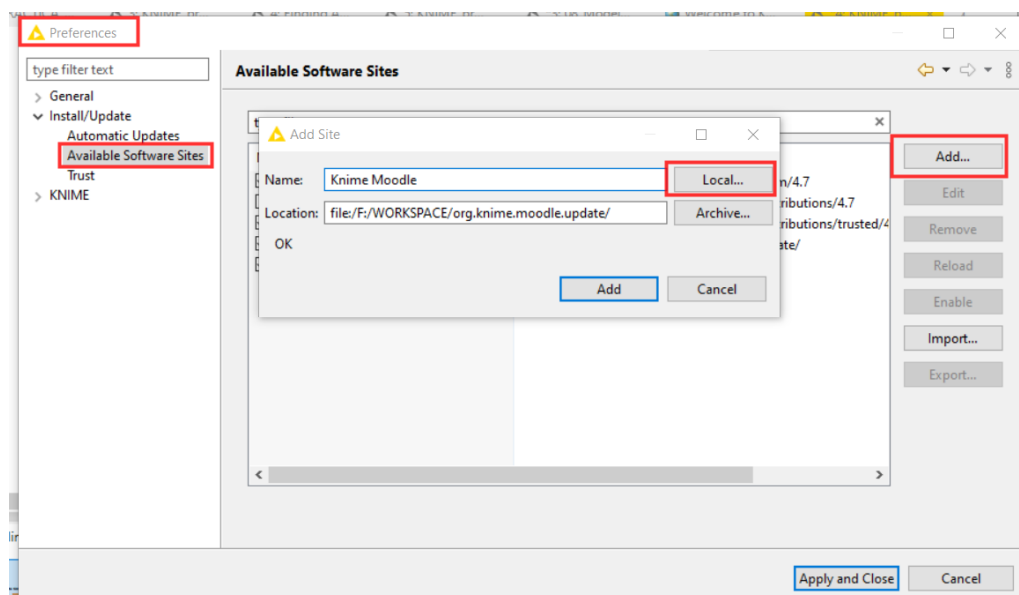


Figura E.3.1: Instalación de la extensión Moodle en KNIME 1.

A continuación seleccionamos el sitio añadido E.3.2,

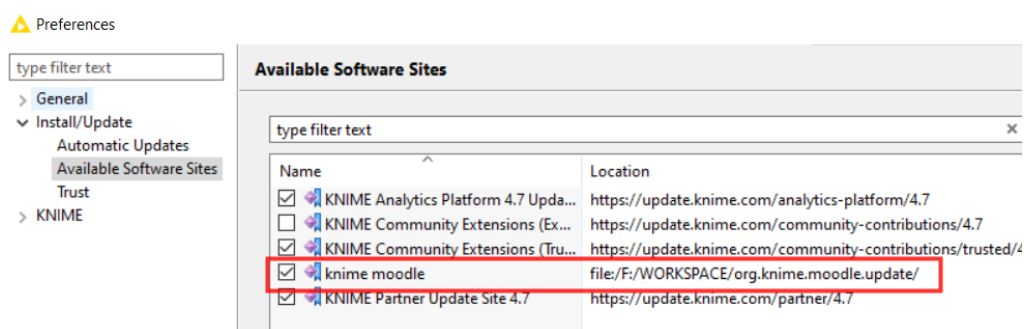


Figura E.3.2: Instalación de la extensión Moodle en KNIME 2.

y seleccionamos la instalación Moodle que queremos instalar E.3.3.

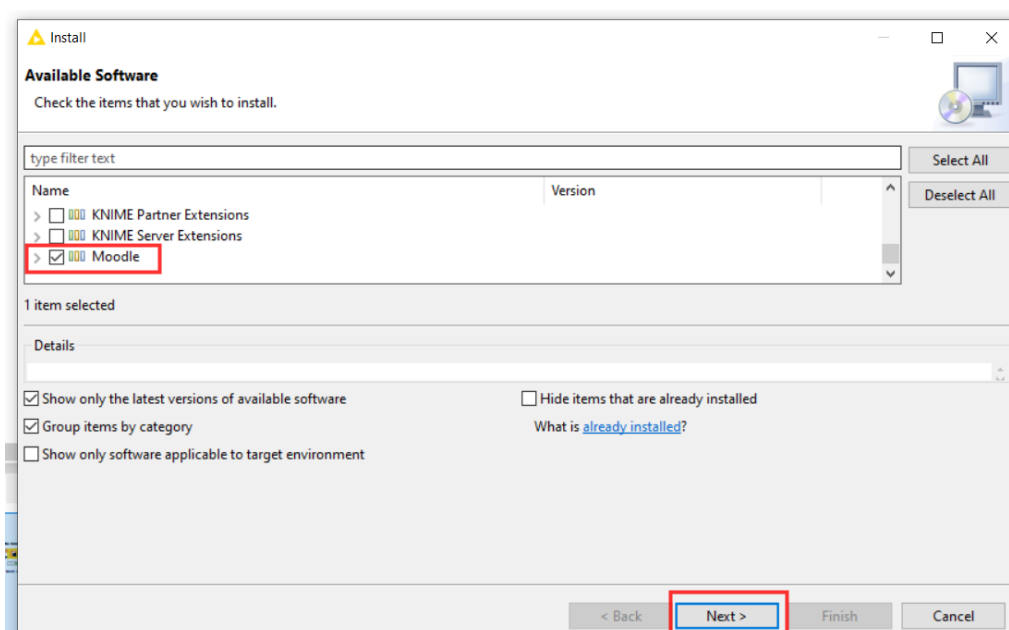


Figura E.3.3: Instalación de la extensión Moodle en KNIME 3.

Nos mostrará detalles de la extensión a instalar antes de completar la instalación E.3.4.

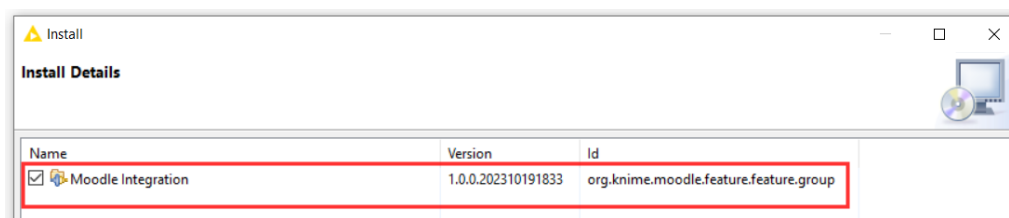


Figura E.3.4: Instalación de la extensión Moodle en KNIME 4.

Los nodos de la extensión Moodle se verán ahora en el repositorio de nodos E.3.5.

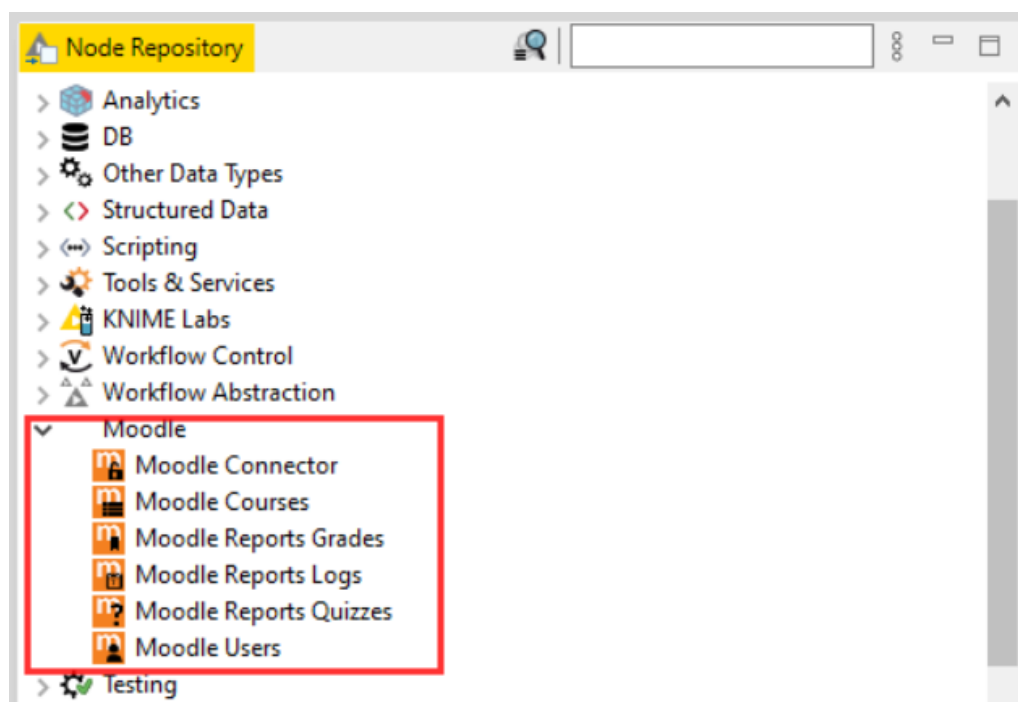


Figura E.3.5: Instalación de la extensión [Moodle](#) en KNIME 5.

E.4. Crear un *workflow* y usar la extensión

Ahora ya podremos crear *workflows* de KNIME incorporando los nodos de integración con Moodle y combinándolos con otros nodos de KNIME. Crea un nuevo Workflow desde *File* ▷ *New* ▷ *New KNIME Workflow* y sigue los siguientes pasos:

Localiza en el repositorio de nodos el nodo **Moodle Connector** y añádelo al *Workflow* como se muestra en la Figura E.4.1.

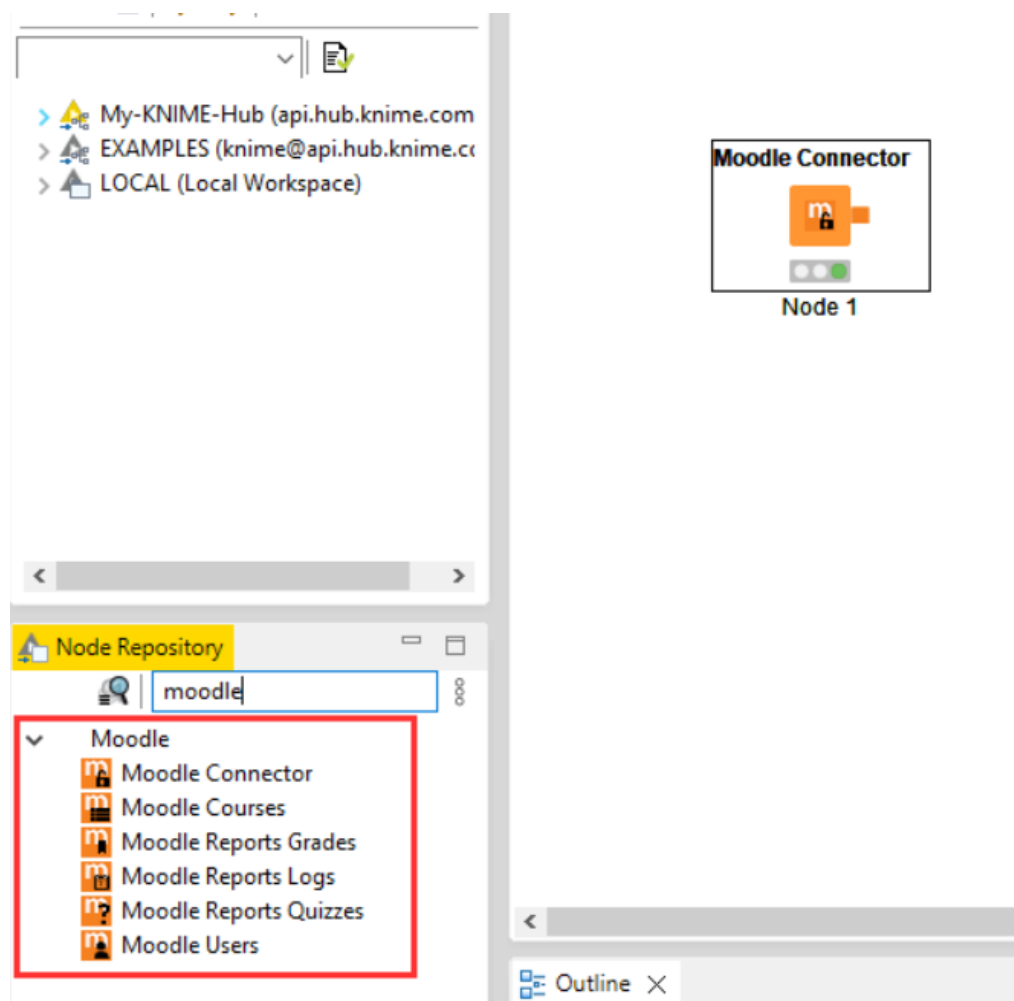


Figura E.4.1: Usando la extensión **Moodle**. Repositorio de nodos.

Configura el nodo para conectar con una plataforma Moodle a la que tengas acceso E.4.2.

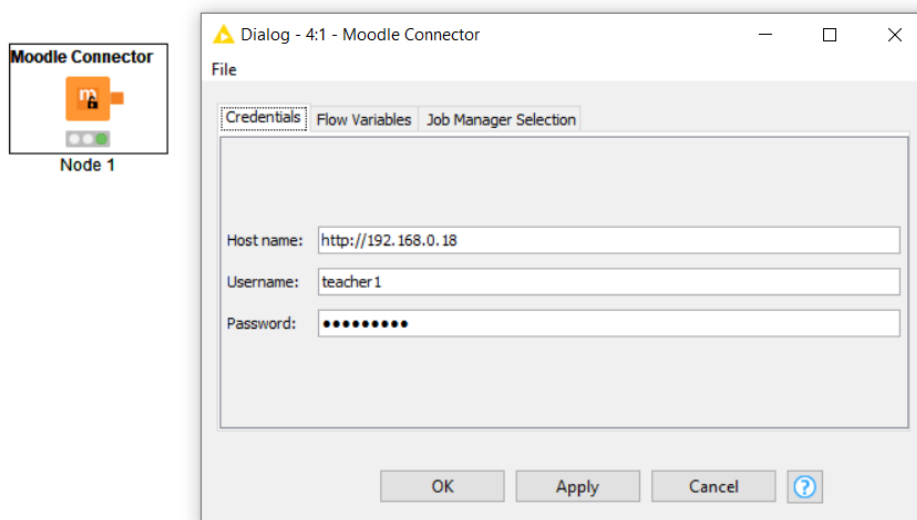


Figura E.4.2: Usando la extensión Moodle 1.

Ejecuta el *workflow* y comprueba que se conecta correctamente a Moodle. En la información de depuración por Consola de KNIME E.4.3 verás las entradas MoodleConnector con información relativa al token de conexión, id del usuario y token del webservice. Si no recibes esta información, revisa que los datos de conexión en el nodo sean correctos.

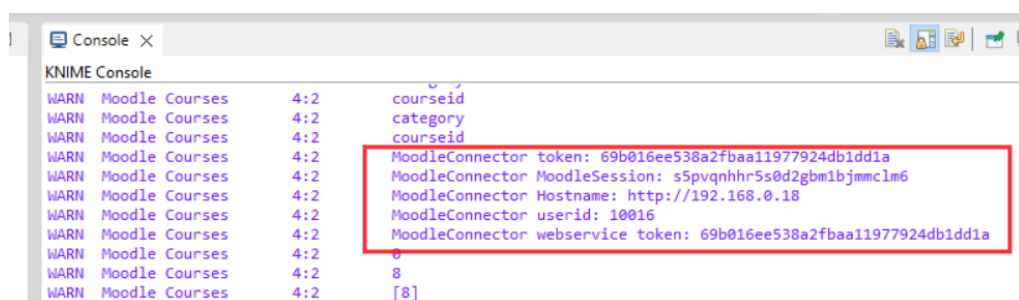


Figura E.4.3: Usando la extensión Moodle 2.

Añade a continuación un nodo de tipo Moodle Courses. El nodo necesita como entrada la salida de un nodo de tipo Table Creator. Une los nodos y configura la tabla con dos columnas, category y courseid E.4.4. En estas columnas podrás indicar los identificados de categorías o cursos de tu plataforma Moodle. Aunque el contenido de la tabla es optativo, debe añadirse y enlazarse el nodo correspondiente.

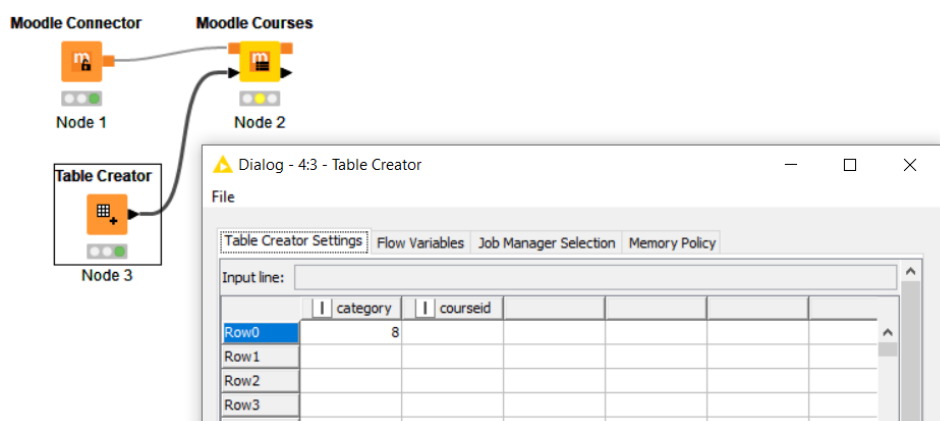


Figura E.4.4: Usando la extensión Moodle 3.

En la configuración del nodo **Moodle Courses**, selecciona la columna correspondiente a la categoría y la correspondiente a los identificadores de cursos [E.4.5](#).

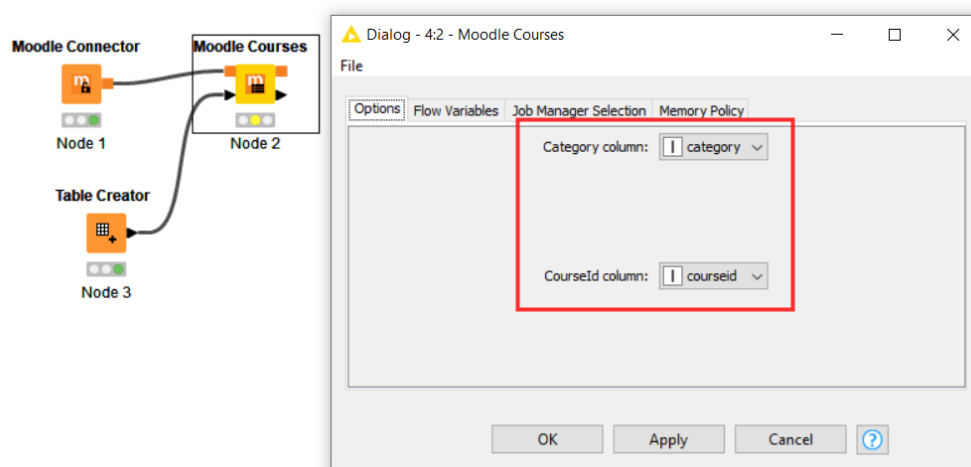


Figura E.4.5: Usando la extensión Moodle 4.

Por último, ejecuta el *workflow* y comprueba la tabla de salida de **Moodle Courses**. Debes obtener una salida como la mostrada en la Figura [E.4.6](#), con información de los cursos.

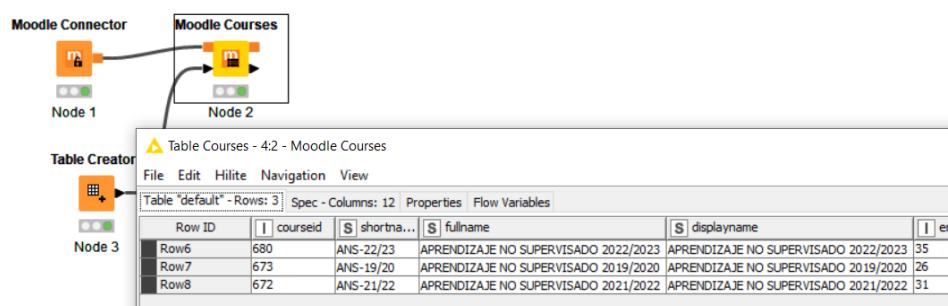


Figura E.4.6: Usando la extensión Moodle 5.

A partir de aquí puedes añadir el resto de nodos de la extensión y combinarlos con otros nodos de KNIME. Consulta el Apéndice C para saber más sobre el diseño y funcionalidad de los nodos disponibles en la extensión [KNIME Moodle Integration](#).

Bibliografía

- [1] Extract transform load (etl). <https://www.databricks.com/glossary/extract-transform-load>.
- [2] Seguridad Social: Bases y tipos de cotización 2023. <https://www.seg-social.es/wps/portal/wss/internet/Trabajadores/CotizacionRecaudacionTrabajadores/36537?changeLanguage=es>.
- [3] Sueldo del programador en españa. <https://www.jobted.es/salario/programador>.
- [4] Supervised learning workflow and algorithms. <https://uk.mathworks.com/help/stats/supervised-learning-machine-learning-workflow-and-algorithms.html>.
- [5] UBUMonitor. Manual de usuario. <https://ubumonitordocs.readthedocs.io/es/latest/>.
- [6] Michael R. Berthold, Nicolas Cebron, Fabian Dill, Thomas R. Gabriel, Tobias Kötter, Thorsten Meinl, Peter Ohl, Christoph Sieb, Kilian Thiel, and Bernd Wiswedel. KNIME: The Konstanz Information Miner. In *Studies in Classification, Data Analysis, and Knowledge Organization (GfKL 2007)*. Springer, 2007.
- [7] N. V. et al. Chawla. Smote: Synthetic minority over-sampling technique. <https://doi.org/10.1613/jair.953>, 2002.
- [8] Abdul Bujang S. D., Selamat A., Ibrahim R., Krejcar O., Viedma E. H., and Fujita H. Multi-class prediction model for student grade prediction using machine learning. <https://ieeexplore.ieee.org/document/9468629>, Junio 2021.

- [9] Universidad de Burgos. Retribuciones funcionarios año 2023. https://www.ubu.es/sites/default/files/portal_page/files/pdi_funcionario_ano_2023.pdf, Marzo 2023.
- [10] Berthold M., Cebron N., Dill F., Di Fatta G., Gabriel T.R., Georg F., Meinel T., Ohl P., Sieb C., and Wiswedel B. Kni-me: The konstanz information miner. version 2.0 and beyond. <https://kops.uni-konstanz.de/server/api/core/bitstreams/e707d437-0ce8-4ebd-bd8a-184b918312b4/content>, Noviembre 2009.
- [11] Raúl Marticorena-Sánchez, Carlos López-Nozal, Yi Peng Ji, Carlos Pardo-Aguilar, and Álgvar Arnaiz-González. Ubumonitor: an open-source desktop application for visual e-learning analysis with moodle. *Electronics*, 11(6):954, 2022.