

FACULTAD DE INGENIERÍA
UNIVERSIDAD DE LA REPÚBLICA

PROYECTO DE GRADO

Extracción y procesamiento de datos para modelado de trayectorias académicas en cursos universitarios

Matías Heredia, Marcelo Rydel, Mario Saul, Giuliano Severi



Tutora
Libertad Tansini

Montevideo, Uruguay - Mayo 2019

Resumen

En Uruguay, existen varios estudios sobre los altos niveles de deserción en la Universidad de la República. Uno de estos afirma que entre 1997 y 2007 el nivel de titulación fue de un 28 %, mientras que un 38 % de los desertores abandonó la carrera antes de completar el primer año de estudio. Un reporte realizado en marzo de 2017 indica que casi el 50 % de la población activa de la Facultad de Ingeniería aún no supera el primer año. Por este motivo resulta apremiante y oportuno desarrollar herramientas que permitan el monitoreo de la actividad estudiantil en cursos de la Universidad para asistir a docentes y autoridades, que mediante el procesamiento de los datos obtenidos posibiliten diagnosticar la situación de los estudiantes, permitiendo la formulación de prácticas educativas que garanticen el aprendizaje de los alumnos y acompañamiento de las etapas de aprendizaje de la trayectoria personal, identificando los éxitos y las dificultades en el proceso educativo.

El crecimiento y generalización de la tecnología educativa, la formación virtual y el uso de Internet como vehículo de aprendizaje ha dado lugar a la aparición de registros digitales que permiten saber cómo se relacionan los estudiantes con los entornos de aprendizaje, con qué frecuencia y en qué condiciones. Esto a su vez ofrece la posibilidad de contar con datos medibles y analizables, permitiendo comprender y optimizar el aprendizaje de los estudiantes y los entornos en los que se producen.

Esta recolección y análisis de datos educativos está estrechamente vinculado con el campo de Learning Analytics o analítica del aprendizaje, una disciplina reciente con potencial transformador, relacionado con el aprendizaje personalizado y adaptativo, y con incidencia en todas las disciplinas educativas.

El objetivo de este proyecto consiste en desarrollar una plataforma que facilite el acceso a datos encontrados en la plataforma Moodle con información del entorno virtual de aprendizaje de un curso de la Facultad de Ingeniería, la visualización y análisis de los datos recabados, así como también la generación de modelos predictivos para identificar situaciones de riesgo en determinados estudiantes, y la posibilidad de proveer feedback para los docentes sobre los motivos de dichas situaciones.

La plataforma desarrollada se denomina Fing Analytics. La misma cuenta con una diversidad de funcionalidades, entre las cuales se destaca la posibilidad de predecir el resultado académico de los estudiantes que están cursando actualmente una asignatura en base a la actividad de estudiantes de años anteriores, aplicando tecnologías de aprendizaje automático y técnicas relacionadas a Learning Analytics.

Glosario

- **API** Conjunto de funciones y/o procedimientos que pueden ser utilizados por otro programa o software. Sus siglas provienen del inglés y significa Interfaz de Programación de Aplicación.
- **Back-end** Parte del software encargados de procesos de negocios y accesos a datos, ejecutado del lado del servidor.
- **Cookie** Información enviada por un sitio web y almacenada en el navegador del usuario, de manera que el sitio web puede consultar la actividad previa del navegador.
- **CSS** Son las siglas de Cascading Style Sheets. Es un lenguaje que describe la presentación de los documentos estructurados en hojas de estilo para diferentes métodos de interpretación. Es muy usado para establecer el diseño visual de los documentos web.
- **CSV** Son las siglas de Comma-Separated Values. Es un formato archivo de texto que almacena los datos en forma de columnas, separadas por coma y las filas se distinguen por saltos de línea. Es una forma muy sencilla de representar la información, normalmente utilizado para importar o exportar de bases de datos de unas aplicaciones.
- **Excel** Programa informático desarrollado por la empresa Microsoft, que facilita el manejo de planillas y hojas de cálculo.
- **Framework** Conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular que sirve como referencia, para enfrentar y resolver nuevos problemas de índole similar.
- **Front-end** Parte del software que interactúa con los usuarios, ejecutado en el navegador web.
- **Facebook** Compañía estadounidense que ofrece servicios de redes sociales y medios sociales en línea.
- **Google** Compañía estadounidense especializada en Internet, software, dispositivos electrónicos y otras tecnologías.
- **HTML** Lenguaje de marcado que se utiliza para el desarrollo de páginas de Internet. Son las siglas de HyperText Markup Language.
- **HTTP** Es el protocolo de transmisión de información de la World Wide Web. Viene del inglés HyperText Transfer Protocol.

- **JavaScript** Lenguaje de programación interpretado, utilizado principalmente en su forma del lado del cliente, implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas.
- **jQuery** Biblioteca de JavaScript que permite simplificar la manera de interactuar con los documentos HTML.
- **JSON** Es un formato de texto liviano para el intercambio de datos. Es ampliamente utilizado para enviar y recibir datos entre distintos sistemas.
- **Login** Proceso mediante el cual se controla el acceso individual a un sistema informático mediante la identificación del usuario utilizando credenciales.
- **Minería de datos** Proceso de detección de información procesable a partir de grandes conjuntos de datos. Se aplica un análisis matemático para extraer información “oculta” de los datos, principalmente tendencias, relaciones y patrones que existen entre ellos que no resultan posibles de detectar de manera manual.
- **Node.js** Entorno JavaScript del lado del servidor, diseñado para generar aplicaciones web de forma altamente optimizada.
- **Python** Es un lenguaje de programación de fácil aprendizaje y cuya forma de escritura favorece la lectura y entendimiento del código.
- **React.js** Biblioteca escrita en JavaScript, que facilita la creación de componentes interactivos y reutilizables para interfaces de usuario.
- **REST** Estilo de arquitectura que permite la intercomunicación entre distintos sistemas sobre la red de Internet, respetando el protocolo HTTP.
- **Script** Archivo de texto, usualmente de tamaño pequeño, que contiene una serie de instrucciones que pueden ser ejecutadas en una línea de comandos.
- **SQL** Lenguaje de programación, diseñado para administrar, y recuperar información de sistemas de gestión de bases de datos relacionales. Son las siglas de Structured Query Language.
- **Staging** Ambiente utilizado para la ejecución de pruebas.
- **Token** Cadena de texto firmada digitalmente, utilizada en sistemas de autenticación de usuarios en sistemas informáticos.
- **TypeScript** Lenguaje de programación libre y de código abierto desarrollado y mantenido por Microsoft. Es un superconjunto de JavaScript, que esencialmente añade tipado estático y objetos basados en clases.

- **URL** Secuencia de caracteres que sigue un estándar y que permite denominar recursos dentro del entorno de Internet para que puedan ser localizados. Son las siglas de Uniform Resource Locator.

Índice general

1. Introducción	9
2. Contexto y marco teórico	11
2.1. Contexto	11
2.2. Learning Analytics	12
2.3. Aprendizaje Automático	15
2.4. Árbol de decisión	16
2.5. ETL - Extract, Transform & Load	20
3. Metodología de trabajo	23
3.1. Relevamiento de requerimientos	23
3.2. Planificación	23
3.3. Reuniones de avance	24
3.4. Herramientas	25
3.5. Proceso de desarrollo	25
4. Diseño	29
4.1. Requerimientos	29
4.2. Arquitectura	30
4.3. Modelo relacional	32
5. Plataforma Administrativa	37
5.1. Arquitectura	37
5.2. Funcionalidades	38
5.3. Tecnologías	39
6. Extracción, transformación y carga de datos	41
6.1. Encuesta demográfica	42
6.2. Documentos de años anteriores	46
6.3. Ambiente de pruebas	48
6.4. Servicios web de Moodle	48

6.5. Registros del curso en Moodle	50
7. Módulo de Aprendizaje Automático	53
7.1. Arquitectura	53
7.2. Funcionalidades	54
7.3. Tecnologías	56
7.4. Experimentación	57
8. Conclusiones y trabajo a futuro	73
8.1. Conclusiones generales	73
8.2. Trabajo a futuro	74
Referencias	75
Anexos	84
A. Manual de usuario	85
A.1. Login	85
A.2. Roles	86
A.3. Pantalla de Resumen	86
A.4. Pantalla de Detalles de Estudiante	88
A.5. Ajustes	90
A.6. Carga de Logs	94
A.7. Usuarios	95
B. Manual de instalación	97
B.1. Base de datos	97
B.2. Plataforma administrativa	98
B.3. Módulo de aprendizaje automático	99
C. Documentación de verificación del sistema	101
C.1. Login	101
C.2. Gestión de un curso	104
C.3. Gestión de usuarios	113
D. Servicios web de Moodle	115
D.1. moodle_mobile_app	115
D.2. core_enrol_get_enrolled_users	116
D.3. gradereport_user_get_grade_items	117
D.4. mod_feedback_get_feedbacks_by_courses	119
D.5. mod_feedback_get_responses_analysis	121

Capítulo 1

Introducción

La Facultad de Ingeniería actualmente utiliza un entorno virtual de aprendizaje denominado EVA, el cual está desarrollado en base al sistema de software libre Moodle. Moodle es una plataforma diseñada para proveer a educadores, administradores y alumnos con un sistema personalizado de enseñanza [1]. El sistema de Moodle es capaz de almacenar información de profesores y estudiantes de Facultad, como su cédula de identidad, notas de alumnos, y registros de actividad en los foros. Se puede acceder a estos datos a través de los servicios web que provee la plataforma [2]. Parte de la motivación por la cual se realiza el proyecto es la de utilizar la mayor cantidad de información que Moodle almacena de manera útil. Los datos que generan los participantes de los foros y cursos, por ejemplo, pueden ser usados para realizar análisis.

La investigación en temas de uso y estudio de datos académicos frecuentemente está relacionada con el término Learning Analytics, el cual refiere al análisis, minería y utilización de información académica con el objetivo de entender mejor al alumnado y sus comportamientos de forma de mejorar el diseño de los entornos en los que aprenden. El campo está estrechamente relacionado a la minería de datos educativa, pero también considera los datos de hábitos o comportamientos particulares de un estudiante [3].

Dado el contexto mencionado anteriormente, se propone como proyecto de grado el diseño y prototipado de una plataforma que permita gestionar, visualizar y analizar datos de los estudiantes del curso de Programación 2 de la Facultad de Ingeniería, buscando aplicar los conceptos antes mencionados de Learning Analytics. La herramienta es diseñada con el fin de ser utilizada por los docentes de la Facultad. En particular se realizó una reunión con integrantes de la Unidad de Enseñanza de la Facultad de Ingeniería, demostrando interés en el proyecto y el potencial de la información que se puede generar para incorporar a los estudios que la unidad realiza, como el reporte de marzo de 2017 que indica que casi el 50 % de la población activa de la Facultad de Ingeniería aún no supera el primer

año [4].

Otro estudio reporta que el entre los años 1997 y 2007, la titulación promedio fue del 28 % mientras que el 72 % restante se encuentra rezagado o desertó de la carrera. [5]

A partir de las necesidades identificadas y el problema planteado, se proponen los siguientes objetivos a realizar como parte del proyecto de grado:

- Relevar los datos disponibles en Moodle para ser utilizados en la herramienta.
- Diseñar e implementar un prototipo para el análisis de los datos recabados aplicando algoritmos de aprendizaje automático.
- Diseñar e implementar una herramienta para la gestión y visualización de los datos, y de los resultados obtenidos de su análisis.

De acuerdo con los objetivos establecidos, se esperan los siguientes resultados:

- Estudio de indicadores a considerar al aplicar técnicas de Learning Analytics.
- Prototipo del módulo de análisis de datos.
- Prototipo del módulo de visualización de datos.

En cuanto a la estructura de este documento, se divide en capítulos a través de los cuales se describe la situación actual y la solución propuesta.

En el capítulo 2 se ofrece una descripción del contexto y marco teórico sobre el cual se realiza el proyecto, describiendo los aspectos fundamentales para la comprensión del mismo. El capítulo 3 describe la metodología de trabajo aplicada para la ejecución del proyecto.

En el capítulo 4 se realiza una descripción de los componentes de la plataforma construida, en cuanto a su diseño y modelo de datos, detallando la estructura de la solución y las decisiones tomadas.

En los capítulos 5, 6 y 7 se describen con mayor detalle las arquitecturas, tecnologías utilizadas y funcionalidades ofrecidas por los dos módulos que componen la implementación de la herramienta construida, así como el proceso de relevamiento de los datos académicos a ser utilizados y como se procedió para obtenerlos.

Por último, en el capítulo 8 se describen las conclusiones y el trabajo futuro propuesto.

También se incluye una sección de anexos con los manuales de usuario y de instalación, así como un informe de las pruebas realizadas y los resultados obtenidos.

Capítulo 2

Contexto y marco teórico

En este capítulo se describe el contexto y situación actual en cuanto a la analítica del aprendizaje. Además, se provee un marco teórico con los conceptos necesarios para la comprensión del proyecto, incluyendo aprendizaje automático, árboles de decisión, procesos de entrenamiento y evaluación de modelos de predicción, y procesos de extracción, transformación y carga de datos.

2.1. Contexto

Un entorno virtual de aprendizaje (EVA) es una plataforma educativa que ofrece servicios en la web. Es un conjunto de herramientas informáticas que posibilitan la interacción didáctica de manera que el alumno pueda llevar a cabo las labores propias de la docencia como son dialogar, leer documentos, realizar ejercicios, formular preguntas al docente, trabajar en equipo, entre otras. Todo ello de forma simulada sin que medie una interacción física entre docentes y alumnos [6].

Los EVA propician que la cantidad de estudiantes de un cierto curso pueda llegar a ser considerada masiva, ya que puede llegar a miles de personas que se conectan a través de internet; como por ejemplo en los denominados cursos en línea masivos y abiertos, también llamados MOOC por sus siglas en inglés (Massive Open Online Courses) [7].

El Entorno Virtual de Aprendizaje utilizado en la Udelar, basado en un sistema de gestión de cursos denominado Moodle, es una herramienta que ha sido adoptada por los actores de la universidad en poco tiempo, cuando se lo contrasta con la larga historia de la institución. La plataforma fue inaugurada el 21 de agosto de 2008 como parte de la formación del proyecto “Generalización del uso educativo de las tecnologías de información y comunicación en la Udelar”, y es prueba de que la utilización de los entornos virtuales de aprendizaje universitarios es cada vez más común [8].

Pueden encontrarse diversas ventajas y desventajas en el uso de los EVA. Por un lado permite que la educación llegue a personas que no tendrían acceso a ella de otra manera; pero al mismo tiempo provoca que el contacto personal entre el docente y el estudiante disminuya. Se refleja el paradigma moderno de que la tecnología acerca a los que están lejos, y aleja a los que están cerca. Una de las mayores ventajas del uso de los EVA es la aparición de datos digitales. Estos datos pueden ser registrados con el fin de identificar y comprender las relaciones entre los distintos partícipes en el proceso de aprendizaje. Ejemplos de estos datos son la frecuencia con la cual un estudiante interactúa con el entorno de aprendizaje, en qué condiciones lo hace, las tareas que realiza, sus resultados académicos obtenidos, etc.

En resumen, son datos cuyo análisis permite identificar patrones en el comportamiento de los estudiantes, y potencialmente señalar los puntos débiles de los mismos en relación con el dictado del curso. El presente proyecto busca tomar ventaja de este hecho con el fin de ofrecer una nueva tecnología que ayude a los profesores de la Facultad a identificar estos comportamientos, monitorearlos, y accionar frente a ellos.

2.2. Learning Analytics

2.2.1. Definición

Las instituciones educativas por lo general hacen poco uso de los datos que se generan debido a la actividad estudiantil. Este tipo de datos ha crecido exponencialmente desde la aparición de las plataformas virtuales de aprendizaje. Learning Analytics busca sacar provecho de estos datos utilizando técnicas de la minería de datos educacional [9].

Learning Analytics, como un nuevo campo de estudio, es considerado un campo interdisciplinario que combina elementos de tecnologías de la educación, pedagogía, aprendizaje automático, inteligencia empresarial y estadística. Learning Analytics busca utilizar los datos educacionales para mejorar los procesos de aprendizaje y enseñanza [10]. Una de las definiciones más populares del campo fue presentada por SOLAR (Society for Learning Analytics Research) en la primera conferencia LAK (Learning Analytics and Knowledge Conference), realizada por primera vez en 2011 y cada año hasta el día de hoy: Learning Analytics es “la medida, recolección, análisis y reporte de datos sobre alumnos y sus contextos, con el propósito de comprender y optimizar el aprendizaje y el entorno en el cual ocurre” [11].

Algunos autores consideran a las técnicas de Learning Analytics como la tercer ola de desarrollo en tecnologías de la educación. La primer ola comienza con el surgimiento de los sistemas de gestión de aprendizaje (LMS por sus siglas en inglés;

Learning Management Systems) en 1991, y la aparición de las redes sociales y su influencia en la educación se considera la segunda ola [12].

2.2.2. Finalidad y estudios recientes

Expertos en el área de la educación superior predicen que en un futuro, Learning Analytics podría ser ampliamente utilizado en la educación en línea para identificar patrones de comportamiento de los estudiantes, con el fin de mejorar los niveles de retención estudiantil en niveles de educación superior [13].

Estudios sobre Learning Analytics concluyen que las ventajas de su uso son diversas, y pueden agruparse según cada parte interesada que se ve beneficiada por su uso. En primer lugar, el uso de Learning Analytics fomenta el acercamiento de estudiantes con la institución educativa, resultando en un mayor auto-conocimiento del proceso personal de aprendizaje del individuo, y potencialmente mejoras en los resultados académicos alcanzados. Los profesores, por otro lado, no sólo cuentan con un monitoreo en tiempo real del comportamiento de los estudiantes, sino que también pueden predecir resultados futuros, en base a los cuales es posible realizar intervenciones eficientes en las metodologías de enseñanza empleadas. Por último están los beneficios para las instituciones que adoptan las técnicas de Learning Analytics, entre los cuales se encuentran el aumento de la retención de los estudiantes, la mejora de la toma de decisiones a nivel institucional, la evaluación de los docentes, la identificación de estudiantes en riesgo, el monitoreo de la actividad estudiantil, la medición del impacto en nuevas políticas o metodologías de enseñanza empleadas, entre otros [10].

Actualmente existen universidades que aplican técnicas de Learning Analytics en sus programas de educación, por ejemplo la Universidad Purdue de Indiana, Estados Unidos, la cual implementa desde 2007 un sistema de resultados estudiantiles que permite proveer a los propios estudiantes con feedback de su participación en los cursos, empleando modelos predictivos y técnicas de aprendizaje automático. Un artículo publicado por docentes de la Universidad demuestra el impacto positivo que la herramienta tiene tanto en los grados de desempeño académico de los estudiantes, como en los niveles de retención en las asignaturas que hacen uso de la misma [14].

Uno de los casos de estudio del uso de Learning Analytics más reconocido en la literatura existente es la de los cursos en línea masivos y abiertos. Debido a su naturaleza, se tratan de cursos generalmente con una alta cantidad de estudiantes registrados, dando lugar a la acumulación de enormes volúmenes de datos, prácticamente imposibles de analizar si no es con técnicas de aprendizaje automático. Aunque los MOOCs han revolucionado la educación en línea, éstos han presentado altos niveles de deserción por parte de los estudiantes registrados, llegando a alcanzar niveles de 95 %. Esto es justificado por los problemas que surgen debido a

la naturaleza de los cursos masivos, como el desafío de mantener a los estudiantes motivados al no contar con un trato directo con los docentes, la posibilidad de los estudiantes de autorregular su aprendizaje, entre otros. Estos factores han dado lugar a abundancia de estudios en los últimos años con respecto a los patrones de compromiso y categorización de estudiantes [15].

2.2.3. Desafíos

Los desafíos del uso de Learning Analytics pueden abordarse según dos aspectos, desafíos técnicos y desafíos educacionales. Algunos ejemplos de los desafíos técnicos incluyen como se rastrean los datos, como son recolectados y como los mismos son evaluados. A continuación se explican con mayor detalle algunos de los principales desafíos educacionales. [10]

- *Ética y privacidad de datos:*
Se trata de uno de los desafíos más importantes que se encuentran al utilizar Learning Analytics. Se plantea la interrogante de si los análisis realizados deberían tener acceso a datos personales de los individuos. En caso afirmativo, es importante determinar quién tiene acceso a dichos datos, y hasta qué punto la información es tratada de modo seguro.
- *Alcance y calidad de datos:*
Conocer el alcance y la calidad de los datos utilizados es fundamental en un proceso de Learning Analytics. El alcance hace referencia a cuanta información debería ser recolectada, y a la variedad de la misma. La calidad, por otro lado, se refiere a que se deben recolectar datos de valor en el proceso de aprendizaje y que permitan hacer predicciones confiables. Los modelos de datos utilizados no deben ser genéricos ni recolectados a partir de datos genéricos, sino que deben estar concentrados en un alcance específico de datos de alta calidad que permita obtener verdaderos conocimientos sobre el aprendizaje y la enseñanza.
- *Fundamentos teóricos y educativos:*
El último desafío clave que puede apreciarse es la brecha que existe entre la teoría y la práctica al aplicar metodologías de Learning Analytics, en lo que se refiere a estudios de enseñanza y aprendizaje. El Learning Analytics debería basarse en suposiciones pedagógicas y epistemológicas, que determinarían el tipo de análisis realizado, con respecto al método a utilizar. Diversos estudios sugieren que muchas de las herramientas de Learning Analytics existentes no se han desarrollado basándose en estrategias educacionales [16].

2.3. Aprendizaje Automático

Uno de los componentes fundamentales de la plataforma cuenta con un procesamiento de datos utilizando algoritmos de aprendizaje automático. El aprendizaje automático, también conocido como Machine Learning, es una disciplina científica del ámbito de la Inteligencia Artificial, que utiliza datos existentes para crear sistemas que aprendan automáticamente. En este contexto, aprender se refiere a la habilidad de identificar patrones complejos en grandes cantidades de datos, y lograr predecir comportamientos. Utilizando aprendizaje automático, las computadoras aprenden e incluso pueden tomar decisiones en base al comportamiento predictivo, con poca intervención humana [17]. Machine Learning se divide en dos áreas principales: aprendizaje supervisado y aprendizaje no supervisado.

En el marco del proyecto se busca a partir de los datos obtenidos de años anteriores predecir el comportamiento de los estudiantes en las diferentes etapas del curso de Programación 2, para esto se utilizó la herramienta de aprendizaje automático *Scikit-Learn*, por lo cual se describen los conceptos necesarios para la comprensión de su funcionamiento, así como los conceptos más estrechamente vinculados con el campo de Learning Analytics. Cabe mencionar que varios métodos de aprendizaje automático fueron investigados, incluyendo métodos de redes neuronales, clasificador Bayesiano, etc, pero se decidió junto a la tutora del proyecto utilizar el método de árboles de decisión, debido a la facilidad que presenta tanto a la hora de visualizar los modelos construidos, como para comprender por qué una predicción tiene un resultado determinado.

2.3.1. Aprendizaje Supervisado

El aprendizaje supervisado permite buscar patrones en datos históricos relacionando todas las variables de entrada con una variable de salida, denominada campo objetivo. Uno de sus usos más extendidos consiste en hacer predicciones a futuro basadas en comportamientos o características presentes en el histórico de datos ya almacenados. Se aplica mediante un algoritmo al cual se le indica cual es el resultado esperado para determinado valor. Tras mostrarle una cantidad suficiente de ejemplos con las entradas y salida correspondiente, si se dan las condiciones necesarias el algoritmo será capaz de dar un resultado correcto incluso para variables de entrada para las cuales no haya visto antes la salida esperada.

Los métodos de aprendizaje supervisado pueden en general categorizarse en métodos de clasificación o de regresión. Un sistema de clasificación predice una categoría, mientras que una regresión predice un número. La predicción del resultado final de una materia cursada por un estudiante es un ejemplo de clasificación (las clasificaciones serían: Exonera, Examen y Recursa). Por otro lado, un ejemplo de regresión es predecir el precio de un artículo, o cuántas reservas se harán en

una fecha determinada en un hotel. [18]

2.3.2. Aprendizaje No Supervisado

Por otro lado, el aprendizaje no supervisado usa datos que no están etiquetados. El fin es explorarlos para encontrar alguna estructura o forma de organizarlos. El aprendizaje no supervisado generalmente trata los datos de objetos de entrada como un conjunto de variables aleatorias, construyendo así un modelo de densidad para el conjunto de datos. A continuación se describen algunas de las técnicas más conocidas y utilizadas de este método. [19]

- *Algoritmos de clustering o agrupación de datos:*
Se agrupan los datos de entrada en grupos de manera tal que los elementos dentro de un mismo grupo (denominado cluster) sean más similares entre sí que aquellos elementos que se encuentran en un grupo distinto. El análisis de los elementos en sí mismo no es un algoritmo específico, sino la tarea general a resolver.
- *Análisis de componentes principales (Principal Component Analysis o PCA):*
Mediante esta técnica se describe un grupo de datos en términos de nuevas variables no correlacionadas. Estos componentes se ordenan por la cantidad de varianza original que describen, por lo que la técnica resulta de mucha utilidad para reducir la dimensionalidad de un conjunto de datos. El PCA se emplea sobre todo en análisis explorativo de datos y para construir modelos predictivos.
- *Descomposición en valores singulares (Singular Value Decomposition o SVD):*
La SVD de una matriz real o compleja es una factorización de la misma con muchas aplicaciones en los campos de la estadística, el procesamiento de señales y otras disciplinas.

2.4. Árbol de decisión

Un árbol de decisión es un modelo de predicción, con una estructura de árbol similar a un diagrama de flujo donde un nodo interno representa una característica (o atributo), la rama representa una regla de decisión y cada nodo hoja representa el resultado. El nodo superior en un árbol de decisión se conoce como el nodo raíz. Aprende a particionar sobre la base del valor del atributo. Se particiona el árbol de manera recursiva. Esta estructura similar a un diagrama de flujo lo ayuda a tomar decisiones. Al tener una visualización como un diagrama de flujo, que imita fácilmente el pensamiento a nivel humano, son fáciles de entender e interpretar

esta fue la principal razón por la que se decidió utilizar árboles de decisión en la implementación del presente proyecto.

2.4.1. Algoritmos para implementar árboles de decisión

En esta sección se describen los distintos algoritmos para implementar árboles de decisión.

ID3 (Iterative Dichotomiser 3) fue desarrollado en 1986 por Ross Quinlan. El algoritmo crea un árbol multi-camino, buscando para cada nodo la característica categórica que producirá la mayor ganancia de información para los objetivos por categoría. Los árboles crecen hasta su tamaño máximo y luego, por lo general, se aplica un paso de poda para mejorar la capacidad del árbol para generalizar datos no vistos aún. [20]

C4.5 es el sucesor de ID3 y eliminó la restricción de que las características deben ser por categorías mediante la definición dinámica de un atributo discreto (basado en variables numéricas) que divide el valor del atributo continuo en un conjunto discreto de intervalos. C4.5 convierte los árboles entrenados (es decir, la salida del algoritmo ID3) en conjuntos de reglas if-then. La precisión de cada regla se evalúa para determinar el orden en el que deben aplicarse. La poda se realiza eliminando la condición previa de una regla si la precisión de la regla mejora sin ella. [20]

C5.0 es la última versión de Ross Quinlan bajo una licencia propietaria. Utiliza menos memoria y crea conjuntos de reglas más pequeños que C4.5 mientras que es más preciso. [20]

CART (Classification and Regression Trees) es muy similar a C4.5, pero se diferencia en que admite variables objetivo numéricas (regresión) y no calcula conjuntos de reglas. CART construye árboles binarios utilizando la característica y el umbral que producen la mayor ganancia de información en cada nodo. [20]

Scikit-Learn, que es usado en este proyecto, únicamente ofrece una versión optimizada del algoritmo CART para la generación de árboles de decisión.

Proceso de creación

El proceso de creación del árbol consiste en:

- Seleccionar el mejor atributo usando medidas de selección de atributos para separar los registros
- Hacer ese atributo un nodo de decisión y dividir el conjunto de datos en pequeños subconjuntos
- Construir el árbol repitiendo los pasos anteriores recursivamente, hasta que se cumpla una de estas condiciones:

- Todas las tuplas pertenecen al mismo atributo
- No hay mas atributos
- No hay mas instancias

Medida de selección de atributos

La medida de selección de atributos (Attribute Selection Measures o ASM) es una heurística para seleccionar el criterio de separación de datos para que estos se dividan de la mejor manera posible. También se conoce como reglas de división porque determina los puntos de quiebre de las tuplas dado un nodo. ASM provee un ranking para cada *feature* o atributo, cuanto mayor puntuación tenga un atributo este va a ser seleccionado como atributo de separación. Las medidas de selección más usadas son Information Gain, Gain Ratio, e índice Gini.

- **Information gain:** En física y matemática, la entropía es un concepto que permite medir la impureza del conjunto de entrada, y está referida a qué tan aleatorio es un sistema o que tan impuro. En la teoría de la información, se refiere a la impureza en un grupo de ejemplos. Information gain es el descenso de entropía, se calcula como la diferencia entre la entropía antes de dividir y el promedio de entropía después de dividir dado un atributo.
- **Índice Gini:** Otro algoritmo de decisión de árboles es CART (Classification and Regression Tree) este usa el método de Gini para crear puntos de división.

$$Gini(D) = 1 - \sum_{i=1}^m P_i^2$$

Donde, P_i es la probabilidad de que una tupla en D pertenezca a la clase C_i . El índice Gini considera una separación binaria por cada atributo. Se puede calcular una suma ponderada de impureza por cada partición. Si se considera una separación binaria con el atributo A del conjunto D en D_1 y D_2 el índice Gini es:

$$Gini_A(D) = \frac{|D_1|}{|D|} Gini(D_1) + \frac{|D_2|}{|D|} Gini(D_2)$$

En el caso de un atributo de valores discretos, el subconjunto que de el mínimo índice de Gini es seleccionado como atributo de separación, en el caso de atributos de valores continuos, la estrategia es seleccionar cada par

de valores adyacentes como un punto de separación y el punto con menor índice de Gini es seleccionado.

El atributo con el mínimo índice de Gini es seleccionado como atributo para realizar la separación.

2.4.2. Entrenamiento y evaluación de resultados

Una vez construido un modelo predictivo, es fundamental su posterior estudio y evaluación para lograr determinar la correctitud de su funcionamiento. Un método para evaluar los resultados es mediante el uso de la denominada *matriz de confusión*, una herramienta que permite la visualización del desempeño de un algoritmo que se emplea en un proceso de aprendizaje supervisado.

Para poder utilizar este método de evaluación de un modelo de clasificación se deben separar los datos de entrenamiento en dos conjuntos de datos o datasets; uno de entrenamiento (generalmente se toman entre el 60 % y 80 % de los datos) y otro de test. El proceso consiste en entrenar el modelo utilizando las observaciones que se encuentran en el dataset de entrenamiento, y luego evaluar el modelo utilizando las observaciones del dataset de test. Esto permite medir como se comporta el modelo cuando se lo aplica a datos nuevos. [21]

A partir de los resultados obtenidos se construye una matriz donde se comparan los valores correctamente predichos por el modelo, con las predicciones erróneas. De la construcción de dicha matriz se desprende el cálculo de diversos indicadores que permiten determinar la eficiencia del modelo construido. [22]

Tomando como ejemplo un modelo binario cuya salida toma los valores de *Positivo* o *Negativo*, se tienen los siguientes parámetros:

- **Positivo verdadero:** TP por sus siglas en inglés (*True Positive*), se refiere a los valores correctamente predichos como verdaderos.
- **Negativo verdadero:** TN por sus siglas en inglés (*True Negative*), se refiere a los valores correctamente predichos como negativos.
- **Falso positivo:** FP por sus siglas en inglés (*False Positive*), se refiere a los valores predichos como positivos, pero cuyo valor real es negativo.
- **Falso negativo:** FN por sus siglas en inglés (*False negative*), se refiere a los valores predichos como negativos, pero cuyo valor real es positivo.

Una vez determinado estos parámetros, es posible calcular los indicadores de eficiencia: *Accuracy*, *Precision*, *Recall* y *F1-score*.

- **Accuracy:** Es el ratio entre predicciones correctamente predichas y el total de predicciones realizadas. Es una medida útil siempre y cuando se trabaje con conjuntos de datos simétricos donde las cantidades de falsos positivos y falsos negativos sean similares.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

- **Precision:** Es la razón entre predicciones correctamente predichas como positivas, y predicciones positivas realizadas. Valores altos de precisión indican una baja cantidad de falsos positivos.

$$Precision = \frac{TP}{TP + FP}$$

- **Recall (Sensitivity):** Es la razón entre predicciones correctamente predichas como positivas, y valores reales positivos. Valores altos de recall indican una baja cantidad de falsos negativos.

$$Recall = \frac{TP}{TP + FN}$$

- **F1-score:** Es el promedio ponderado de la precisión y el recall. Toma en cuenta tanto la cantidad de falsos positivos como de falsos negativos.

$$F_1score = 2 \frac{Recall * Precision}{Recall + Precision}$$

2.5. ETL - Extract, Transform & Load

Para la realización del proyecto se notó desde un principio que se necesitaba recabar información de varias fuentes totalmente distintas. Al investigar brevemente dichas fuentes, se pudo concluir que se debe utilizar una variedad de tecnologías para acceder a la información; que los datos se encuentran estructurados y modelados de distintas maneras; y que es necesario un método que ayude a procesar estos datos para que sea posible ingresarlos en la base de datos del sistema.

Aquí es donde entran los denominados procesos de extracción, transformación y carga de datos, o como se los conoce por sus siglas en inglés, procesos ETL. Este tipo de procesos son de gran ayuda cuando se intenta extraer datos de distintas fuentes y transformarlos para lograr una estructura o un modelo común que ayude a la organización que lo implementa a lograr sus objetivos. Típicamente los datos transformados son cargados a una base de datos. El objetivo detrás de un proceso ETL es lograr manejar datos uniformes en un solo lugar (o conjunto acotado de lugares) de forma de facilitar el cálculo, cómputo o análisis de los mismos. [23]

Las operaciones básicas de un proceso ETL son las siguientes:

- Identificación de las fuentes de datos
- Extracción de datos de las fuentes
- Filtrado de datos (selección y proyección)
- Integración de varias estructuras fuente en una única estructura destino
- Transformación de datos de entrada en datos de salida

2.5.1. Extracción

Involucra técnicas para la extracción de información de las fuentes identificadas. Se pueden utilizar tanto herramientas de ETL ya existentes o construir nuevas que se adapten a las necesidades de cada fuente de datos.

Típicamente se asocia a cada fuente de datos un componente de extracción (o también denominado *wrapper*). Estos componentes tienen las funciones básicas de proveer una descripción de los datos almacenados en la fuente con un modelo común, y otras funciones más complejas como puede ser la detección y extracción de cambios en la información, y propagar dichos cambios a los datos ya obtenidos.

El proceso de extracción se puede clasificar de dos maneras. En primer lugar, existen los métodos de extracción lógica, que se dividen en extracción completa y extracción incremental. El primero refiere a cuando los datos extraídos componen la totalidad del sistema fuente, por lo que no es necesario llevar registro de los cambios que se hacen a los datos cada vez que se realiza la extracción. Un ejemplo puede ser la exportación de datos de una tabla de una base de datos. El segundo, refiere a cuando los procesos de extracción necesitan identificar los datos que cambiaron desde la última extracción exitosa. Es muy común que los sistemas fuente no cuenten con lógica adicional de control de cambios, por lo que sería necesario agregar tal lógica del lado del componente de extracción de dicha fuente. Mecanismos como *timestamps* de actualización y cálculo de los datos son algunos de los más comunes para tratar con este tipo de extracción.

En segundo lugar, se mencionan los métodos de extracción física. Estos dependen de los métodos elegidos de extracción lógica y de las capacidades y restricciones de las fuentes de datos. Los datos pueden ser extraídos tanto de forma en línea (*online*) como fuera de línea (*offline*). Cuando ocurre la extracción *online* el componente de extracción se conecta directamente con la fuente o con un sistema intermedio que almacene la información de manera preconfigurada. Por otro lado, en la extracción *offline* los datos son llevados fuera del sistema fuente primero, para luego ser trabajados. [24]

2.5.2. Transformación

Es constituida por varios procesos que moldean los datos con el objetivo de llegar a un modelo uniforme. Algunos de los pasos generalmente presentes en el proceso de transformación son los de limpieza, integración de valores y estructura, resumen y agrupamiento, y cambios de estructuras en la representación de datos. Típicamente luego del proceso de extracción los datos son almacenados en una base de staging desde la cual se aplican los procesos de transformación.

El proceso de limpieza de los datos es uno de los más importantes, ya que está presente en la mayoría de los procesos de migración y su objetivo es mejorar la calidad de los datos. La limpieza incluye estandarizar datos libres, como por ejemplo cédulas de identidad, nombres y fechas. También se encarga de descartar (o completar en caso de que sea posible) datos incompletos; todo con ayuda de las funciones de conversión y normalización. Las funciones de conversión se encargan de convertir ciertos tipos y representaciones de datos a otros con mayor utilidad a la hora de procesar o analizar; por ejemplo convertir los tipos *string* que valen por números a datos del tipo *integer*. Las funciones de normalización sirven con el objetivo de formatear datos de un mismo tipo a un formato común, con el fin de permitir la fácil comparación entre los valores.

Dentro de la transformación se pueden también encontrar algoritmos de correspondencia entre campos equivalentes de fuentes diferentes. [25]

2.5.3. Carga

Es el proceso que tiene como objetivo cargar los datos transformados al medio de almacenamiento final accedido por usuarios finales o por sistemas aplicativos, usualmente una base de datos. Puede haber procesos de carga inicial o de actualización [25]. Los de carga inicial están enfocados únicamente en poblar con los datos una sola vez, mientras que los de actualización se los provee con mecanismos que sirven para modificar datos ya existentes. Hay ocasiones en las que los procesos de carga inicial y actualización coinciden.

Capítulo 3

Metodología de trabajo

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

3.1. Relevamiento de requerimientos

Las primeras sesiones entre los integrantes de este grupo de proyecto de grado y la tutora fueron utilizadas para relevar los requerimientos del prototipo a construir. La problemática a atacar fue planteada y partir de la misma se definieron los primeros requerimientos funcionales, y algunos de los no funcionales. Si bien éstos fueron definidos en una etapa temprana del proyecto, fue necesaria una investigación sobre los distintos conceptos a utilizar (como aprendizaje automático, por ejemplo) previa al desarrollo.

3.2. Planificación

Al comenzar el proyecto se planificaron las actividades previstas así como su duración como se muestra en el diagrama de Gantt de la Figura 3.1. Las actividades previstas fueron las siguientes:

1. *Planificación.*
2. *Investigación.*
3. *Desarrollo.*
4. *Prueba.*
5. *Informe.*

En principio, se planificó terminar el proyecto en el mes de Diciembre del 2018 como se muestra en la Figura 3.1.

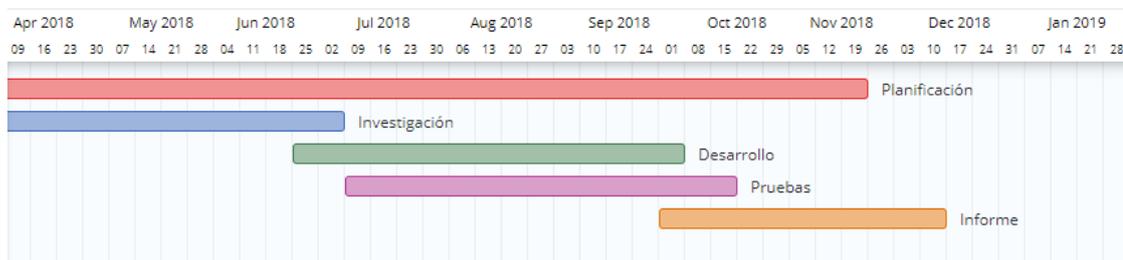


Figura 3.1: Planificación original del proyecto

Sin embargo, esto no ocurrió tal cual planificado. La principal razón fue la gran incertidumbre que hubo en las etapas más tempranas del proyecto con respecto a las posibilidades disponibles para la extracción de datos y la posterior dificultad para procesarlos. Esto sumado a que durante los meses de Enero y Febrero los integrantes de este grupo de proyecto de grado gozaron de sus respectivas licencias, resultó que el proyecto efectivamente concluya en el mes de Mayo del 2019 como se muestra en la Figura 3.2.

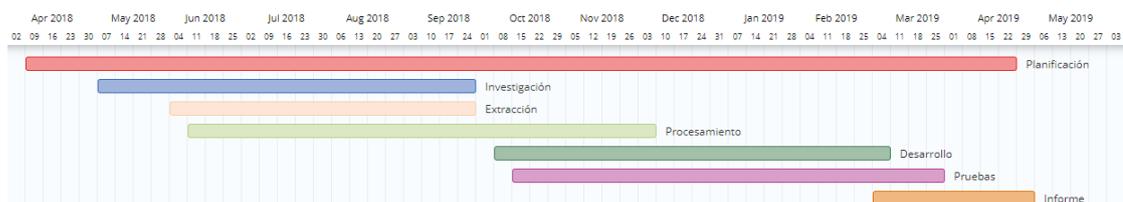


Figura 3.2: Progreso real del proyecto

3.3. Reuniones de avance

A lo largo del proyecto hubo varias instancias de reporte en las cuales se mostró el avance obtenido. Estas reuniones fueron mensuales hasta muy avanzado el proyecto, donde se continuaron haciendo cada dos semanas. Todas las reuniones tuvieron una duración de aproximadamente una hora.

Los objetivos de estas reuniones fueron los siguientes.

1. *Informar sobre posibles bloqueos externos al proyecto.*

Este tipo de bloqueos incluyó la gestión de permisos de profesor para el curso de Programación 2 en EVA, la creación de una copia del curso viejo de

Programación 2, la creación de una copia de la base de datos de Moodle, y finalmente la facilitación de las notas de cursos anteriores en formato Excel.

2. *Presentaciones orales sobre la bibliografía y antecedentes.*

Las presentaciones fueron realizadas con el fin de organizar los conceptos revisados de Learning Analytics, Aprendizaje Automático, procesos ETL y antecedentes de proyectos similares. De esta manera toda la información revisada individualmente se comunica al resto del grupo y a la tutora del proyecto.

3. *Demostraciones del prototipo en construcción.*

Éstas fueron realizadas con el fin de mostrar las funcionalidades desarrolladas del proyecto, y cotejar que se esté construyendo el sistema correcto. Fueron en especial útiles para tomar en cuenta sugerencias y ajustes que la plataforma necesitara.

4. *Demostración final.*

Una vez construido el prototipo de la plataforma, se realizó además una reunión de demostración con integrantes de la Unidad de Enseñanza de la Facultad de Ingeniería. Dicha reunión tuvo un resultado positivo, quedando validado el potencial de la herramienta para futuros años.

3.4. Herramientas

Para el proyecto se utilizaron varias herramientas indispensables. Para el control de versiones se utilizó Git [26], con dos repositorios de código alojados en GitHub [27], uno para cada módulo implementado.

Para la organización de tareas a realizar a lo largo del proyecto se utilizaron dos tableros de Trello [28]. El primero para las tareas asociadas al sub-proyecto de la plataforma administrativa, y el segundo para tareas del equipo; tareas de investigación y documentación son algunos ejemplos.

3.5. Proceso de desarrollo

Una vez que se contó con la base teórica acorde a las necesidades del proyecto, se comenzó con una fase de experimentación con distintas herramientas de aprendizaje automático utilizando los datos de cursos anteriores de Programación 2, y posteriormente con los datos de la encuesta demográfica realizada. En un principio, se investigaron distintas herramientas de minería de datos y aprendizaje automático. Algunas de ellas fueron:

- *RapidMiner*. [29]
Es una herramienta de procesamiento de datos muy poderosa. Tiene la gran desventaja de ser una aplicación paga, por lo que esta herramienta fue descartada.
- *KNIME*. [30]
Es una plataforma de código abierto basada en auxiliar soluciones dirigidas por datos. Tiene la ventaja de ser gratuita y de uso libre.
- *Orange*. [31]
Es una herramienta para minería de datos y visualización. Es de código abierto y de uso libre.

Luego de probar las herramientas se llegó a la conclusión de que cualquiera de ellas requeriría recorrer una curva de aprendizaje de alto nivel, interfiriendo con el objetivo de que la plataforma pueda ser utilizada por docentes de la manera más simple posible. Por otro lado, la posibilidad de integración con ellas es nula o muy limitada, dificultando la independencia entre módulos y herramientas externas. Finalmente se tomó la decisión de utilizar bibliotecas de aprendizaje automático en vez de herramientas ya construidas.

En un principio se quiso construir un back-end utilizando únicamente javascript con *node.js* [32]. Esto no fue posible debido a que las librerías disponibles de aprendizaje automático para tal lenguaje son muy limitadas en cuanto a funcionalidades. Por lo tanto, se procedió a investigar sobre el uso de Python [33] y la biblioteca *Scikit-Learn* [34], ya que ésta es muy popular en la comunidad del aprendizaje automático y cuenta con un gran conjunto de funcionalidades que resultaron útiles a la hora de implementar la solución.

La solución cuenta con dos módulos bien definidos e independientes, el módulo de aprendizaje automático y la plataforma administrativa. Los mismos son subproyectos de naturalezas considerablemente diferentes debido a los tipos de rutinas que requieren implementar y la forma de manejar los datos que poseen. Por lo que ambos fueron construidos siguiendo distintos procesos de desarrollo.

El primer módulo a construir fue el de la plataforma administrativa. Se utilizó un proceso de desarrollo Kanban [35], cuyo funcionamiento consiste en dividir el desarrollo en tareas bien definidas e indicar su estado de progreso en una tabla. Estos estados pueden ser:

1. “*Para hacer*”. La tarea esta lista para ser tomada.
2. “*En progreso*”. La tarea esta siendo realizada.
3. “*Terminada*”. La tarea fue terminada según la *definición de terminado*.

La *definición de terminado* utilizada para la plataforma administrativa consistía en tres puntos.

1. Se cumple el comportamiento esperado definido en la tarea.
2. En caso de ser una funcionalidad a ser ejecutada desde un cliente, se debe verificar que el sistema se comporta bien ante entradas de datos malformados, erróneos o inconsistentes.
3. En caso de haber cambios en código, la rama de Git utilizada es correctamente unida a la rama de desarrollo.

En cuanto al control de versiones, se optó por utilizar el proceso *Gitflow* [36] propuesto por Atlassian [37], ya que es el estándar utilizado en la industria. A modo de explicación se puede decir que el proceso consiste en mantener una rama de producción (*master*) y una rama de desarrollo (*develop*). Las nuevas funcionalidades parten de la rama de desarrollo, y una vez terminadas son unidas a la misma a través de un *Pull Request*. Este *Pull Request* es después revisado por un integrante del equipo de desarrollo distinto a quien lo creó, con el fin de compartir los cambios realizados y verificar que no se hayan cometido errores al momento de implementar la funcionalidad. La rama de desarrollo es utilizada para realizar pruebas de integración entre las funcionalidades. Las ramas de producción y desarrollo son unidas cuando que se quiera liberar el producto.

Por otra parte, para el módulo de aprendizaje automático se tomó una iniciativa distinta. Al ser un módulo con funcionalidades muy distintas a las de la plataforma administrativa, se tomó como enfoque un proceso de desarrollo incremental el cual tuvo varias iteraciones. Al ser el entrenamiento de los modelos y la predicción las funcionalidades clave del módulo se prestó especial atención a que los mismos dieran resultados coherentes y de calidad durante el proceso de desarrollo. Estos resultados son detallados en el capítulo 7.

Capítulo 4

Diseño

En este capítulo se describen los requerimientos tenidos en cuenta para el diseño de la solución, la arquitectura de la misma, y el modelo relacional propuesto para los datos.

4.1. Requerimientos

El planteo inicial del presente proyecto consistió en la implementación de un prototipo que sea capaz de obtener información de los estudiantes del curso de Programación 2 con el fin de identificar aquellos estudiantes en riesgo de perder la asignatura. Esta idea se iteró durante varias sesiones de definición de requerimientos con la tutora encargada hasta delimitar completamente el alcance. Tanto requerimientos funcionales como no funcionales fueron tratados en las reuniones.

Un punto importante a destacar es que algunas de las funcionalidades brindadas por la plataforma no fueron creadas debido a requerimientos pedidos, si no como una forma de mejorar la solución.

En las siguientes sub-secciones se detallan los requerimientos funcionales y no funcionales.

4.1.1. Requerimientos funcionales

- Identificar cuales son los estudiantes en riesgo de perder el curso utilizando algoritmos de aprendizaje automático.
- Identificar de una manera fácilmente interpretable por los docentes del curso las razones por las que un estudiante está en peligro de perder el curso.
- Tener acceso a la información de un curso de Programación 2 de forma unificada. Esta información consiste en:

- Respuestas de encuestas
 - Resultados de laboratorio
 - Notas de parciales
 - Notas finales
- Extraer información de la plataforma Moodle, o en su defecto, que sea posible importar información al sistema a través de un archivo CSV.

4.1.2. Requerimientos no funcionales

- Generar una solución que cuente con los siguientes atributos de calidad.
 - *Modularidad.* La solución debe dividirse en componentes independientes bien definidos que puedan ser intercambiados.
 - *Usabilidad.* La interfaz de usuario debe ser intuitiva y clara.
 - *Portabilidad.* El sistema debe poder ser portado a distintos sistemas operativos sin que sea necesario hacer cambios a nivel de código fuente.
 - *Escalabilidad.* El crecimiento en la cantidad de datos almacenados no debe impactar de manera excesiva la performance del sistema.
 - *Mantenibilidad.* El esfuerzo requerido para actividades de mantenimiento debe ser bajo.
 - *Adaptabilidad.* El esfuerzo requerido para utilizar la plataforma para otros cursos debe ser bajo.

4.2. Arquitectura

Al momento de idear la arquitectura de la solución se tomaron en cuenta los requerimientos no funcionales, de manera de cumplir con ellos lo mejor posible. Para atender el requerimiento de la *modularidad* la solución es dividida en tres partes con el fin de que cada parte funcione como un módulo independiente de los otros. De esta manera cada una puede ser intercambiada sin que las demás necesiten ser ajustadas, siempre y cuando se implementen correctamente las interfaces expuestas. Estas partes son: la base de datos, el módulo de la plataforma administrativa y el módulo de aprendizaje automático.

El módulo de aprendizaje automático es un sistema escrito en *Python* y expone servicios web REST. A su vez se comunica con la base de datos directamente utilizando SQL.

La plataforma administrativa contiene dos sub-sistemas. El back-end y el front-end. El primero es un sistema escrito en *TypeScript* [38], y corre en *node.js*. Es el encargado de consumir de las interfaces del módulo de aprendizaje automático y Moodle, se conecta a la base de datos mediante SQL, y expone servicios web REST que son utilizados desde el front-end. Este último es la interfaz que utilizarán los usuarios para hacer uso de la mayoría de las funcionalidades de la plataforma; se trata de un sitio web estático que utiliza *React.js* [39] como *framework* de desarrollo. Al momento de desarrollar el front-end se tuvo especial cuidado en que la *usabilidad* es un requerimiento. Al no ser un requerimiento que pueda medirse con exactitud se pidió feedback acerca de la usabilidad de la plataforma en cada instancia de demostración.

Cabe destacar que los lenguajes de programación y tecnologías utilizadas para la construcción de la solución son transportables a cualquier sistema operativo teniendo los entornos de ejecución correctamente instalados (*node.js* y *Python*). De esta manera la solución satisface el requerimiento de *portabilidad*.

Ambos módulos se conectan a la base de datos directamente ya que se tomó en cuenta el requerimiento de *escalabilidad*. Resulta mucho más rápido acceder a la fuente de información directamente que utilizando una capa adicional que abstraiga el acceso a datos. Es importante tener esto en cuenta ya que al momento de entrenar los algoritmos de aprendizaje automático se consulta una cantidad relativamente grande de información. En principio, con la cantidad de datos con los que se cuenta actualmente la diferencia no es notable, pero a medida que el volumen de datos crezca tener acceso a la base directamente puede hacer la diferencia entre que el entrenamiento demore segundos o minutos.

Las interconexiones entre las distintas partes se pueden ver gráficamente en la Figura 4.1.

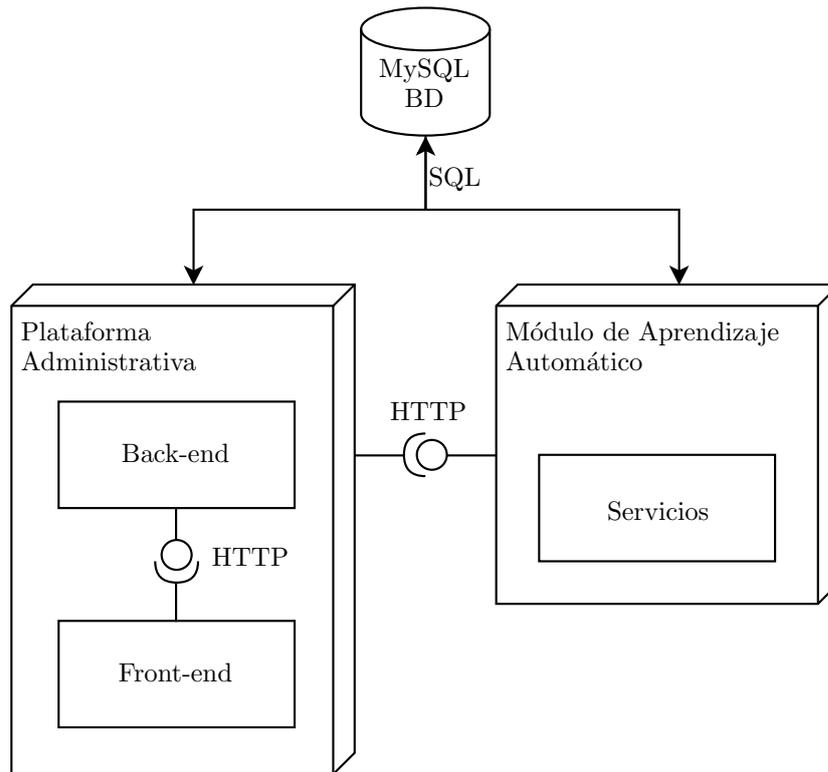


Figura 4.1: Arquitectura de la solución

Al tratarse de un sistema de porte medio, se justifica la *mantenibilidad* del mismo, no hay enormes cantidades de lógica de negocio programadas, por lo tanto ajustes de pequeña a mediana complejidad pueden realizarse sin mucho esfuerzo. A su vez, los módulos internos están divididos en porciones de lógica separadas mediante criterios de SoC - Separation of Concerns [40], lo que facilita aspectos del código como la legibilidad y la verificabilidad.

Una de las grandes falencias a nivel de arquitectura que posee el prototipo es que la *robustez* como atributo de calidad no pudo ser conseguido, debido a que no se tiene control sobre los posibles cambios que pueda tener la API de Moodle o el sitio EVA. Esto último se explica en más detalle en las secciones 6.4 y 6.5.

4.3. Modelo relacional

En la Figura 4.2 se presenta el modelo relacional utilizado. En primer lugar se modelan los estudiantes con la tabla *students*, incluyendo los datos básicos de un estudiante como su CI, nombre completo (*Name*) e email. También se incluye información extraída desde Moodle. *ProfileImageUrl* es la URL de su imagen de

perfil en Moodle, y *EvaId* es el identificador del estudiante en la plataforma. Este último es de gran importancia ya que se necesita para extraer información de sus notas, resultados y para identificar su respuesta en la lista de respuestas de la encuesta del curso.

Por otro lado, se modelan los distintos cursos de la asignatura con la tabla *courses*. Se indica el año del curso (*Year*), su identificador en Moodle (*EvaId*) y el nombre que se le da a la encuesta (*SurveyName*).

Un curso tiene varios laboratorios (*assignments*), parciales (*tests*) y un conjunto de notas finales (*finals*). Cada una de estas entidades está ligada a un único curso y tiene un nombre para ser mostrado en la plataforma administrativa (*Name*), y el nombre con el cual la actividad está representada en Moodle (*EvaName*). Se agrega este último para que la correspondencia entre los registros de la solución y las actividades de Moodle sea dinámica y configurable por los usuarios de la plataforma administrativa. Los laboratorios tienen la particularidad de que puede existir una actividad para la re-entrega, por lo que se agrega la columna *EvaRedoName* para corresponder una re-entrega con dicha actividad de Moodle.

Un estudiante puede tener varios detalles de cursos (*course_details*). Esta relación se asocia con todas las entidades referentes a un curso específico. Por tanto, un detalle de curso está asociado a un único curso. También se tienen predicciones (*predictions*), respuestas de la encuesta (*student_surveys*) y los registros de actividad de Moodle (*logs*).

De las predicciones se tiene el resultado (*Result*) y la marca de tiempo en la cual se creó el registro (*Timestamp*). El resultado puede tomar los valores “PASS” (Exonera), “FAIL” (Recurso) ó “EXAM” (Derecho a exámen).

De los registros de actividad se tiene la acción (*Action*) y la marca de tiempo en la cual se creó el registro. La acción puede tomar los valores “ACCESS”, “FILE_ACCESS”, “FORUM_ACTIVITY” ó “SURVEY_RESPONSE”.

Un detalle de curso está asociado con los resultados que tuvo un estudiante para el curso en cuestión, modelado como la entidad *activity_result*. Estos resultados están asociados únicamente con un laboratorio, parcial o nota final dependiendo de su tipo (*Type*). Se indica el resultado de un laboratorio cuando es de tipo “Assignment”, de un parcial cuando es de tipo “Test”, e indica una nota final cuando es de tipo “Final”.

Para el uso de la plataforma administrativa se tiene la tabla de usuarios (*users*), con su rol (*Role*). Al usuario se le asocia información extraída de Moodle, como su nombre de usuario (*Username*), URL de la imagen de perfil (*ProfilePictureUrl*) y nombre completo (*FullName*). El rol del usuario puede tomar los siguientes valores:

- *new*: Un usuario ingresa al sistema por primera vez. No tiene permisos para acceder a la información.
- *teacher*: Usuario docente, solo tiene permisos para consultar información.

- *admin*: Usuario administrador. Tiene permisos para consultar y editar ajustes de los cursos.

Restricciones no estructurales:

1. Cada instancia de la entidad *activity_results* está relacionado a solo una de las entidades (*assignment*, *tests*, *final*), dependiendo del valor del atributo *Type*.
2. La entidad *assignment*, *tests* o *finals* y la entidad *course_details* relacionadas a un *activity_result*, deben estar relacionadas al mismo *course*.

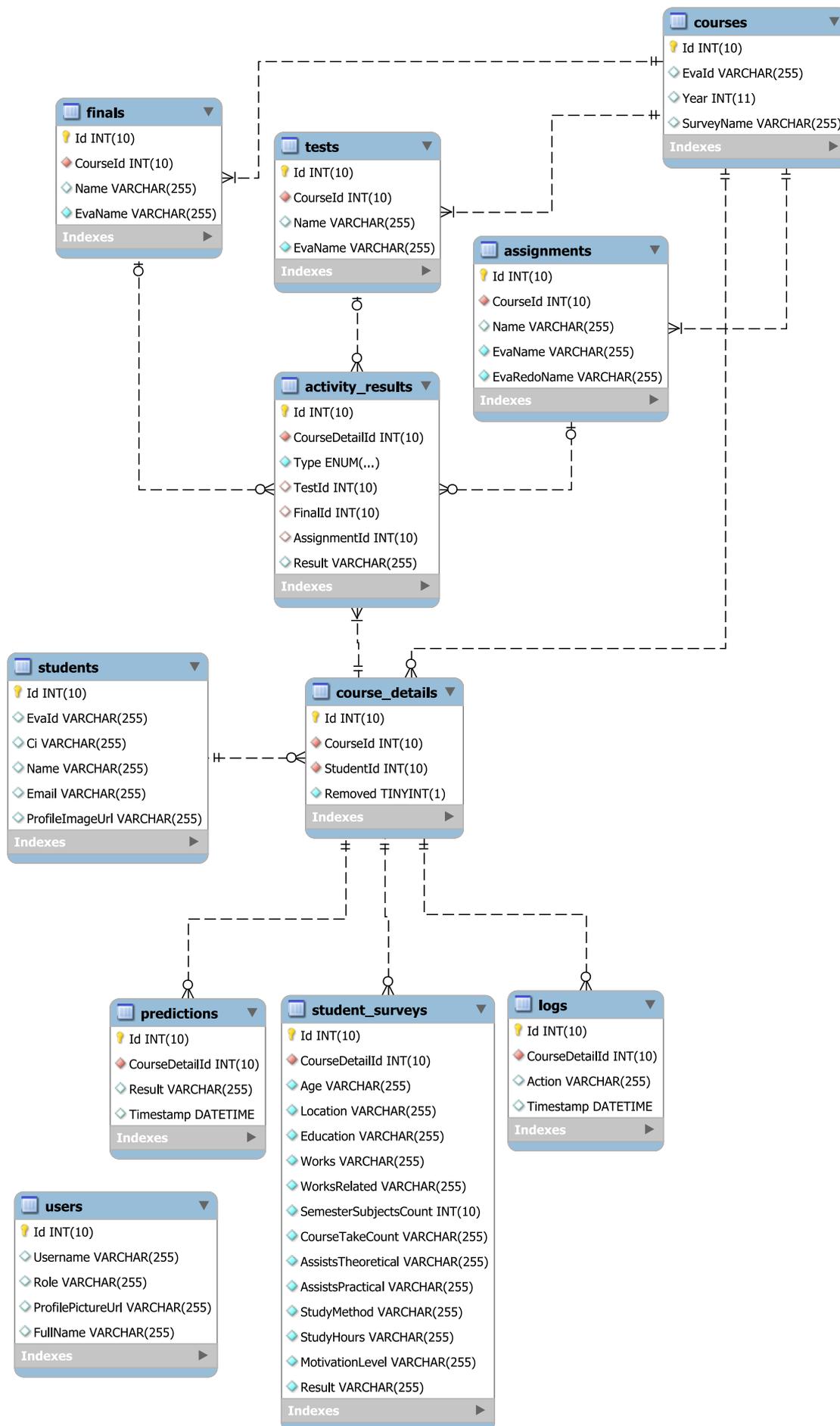


Figura 4.2: Modelo relacional de la solución

Capítulo 5

Plataforma Administrativa

En este capítulo se presenta una descripción detallada de la Plataforma Administrativa construida, indicando su arquitectura, tecnologías utilizadas y funcionalidades disponibles. Complementariamente, en el Anexo A se encuentra el manual de usuario de la Plataforma Administrativa donde se explican con mayor detalle las funcionalidades que la misma ofrece.

5.1. Arquitectura

En esta sección se detalla la arquitectura utilizada para el desarrollo del módulo administrativo, la cual se ilustra en la Figura 5.1.

La plataforma consiste de dos módulos desacoplados, un módulo de back-end desarrollado en *node.js* y uno de front-end desarrollado en *React*. Ambos módulos cuentan con una arquitectura basada en capas.

El módulo back-end tiene una capa de *acceso a datos*, donde se realiza la conexión con una base de datos MySQL a través del framework *Objection* de *javascript*. Esta capa es consumida por la capa de *lógica de negocio*, donde también se consumen los servicios web expuestos por la REST API de Moodle. Por último, la capa de *controladores* expone los servicios del módulo como servicios web REST.

La arquitectura del módulo front-end es similar. Inicialmente se tiene una capa que actúa como *cliente API* y consume los servicios expuestos por el módulo back-end. Las respuestas de dichos servicios son convertidas a entidades utilizadas por la capa de *presentación* en la capa *mapeador de objetos*. La capa de *presentación* se refiere a la interfaz de usuario.

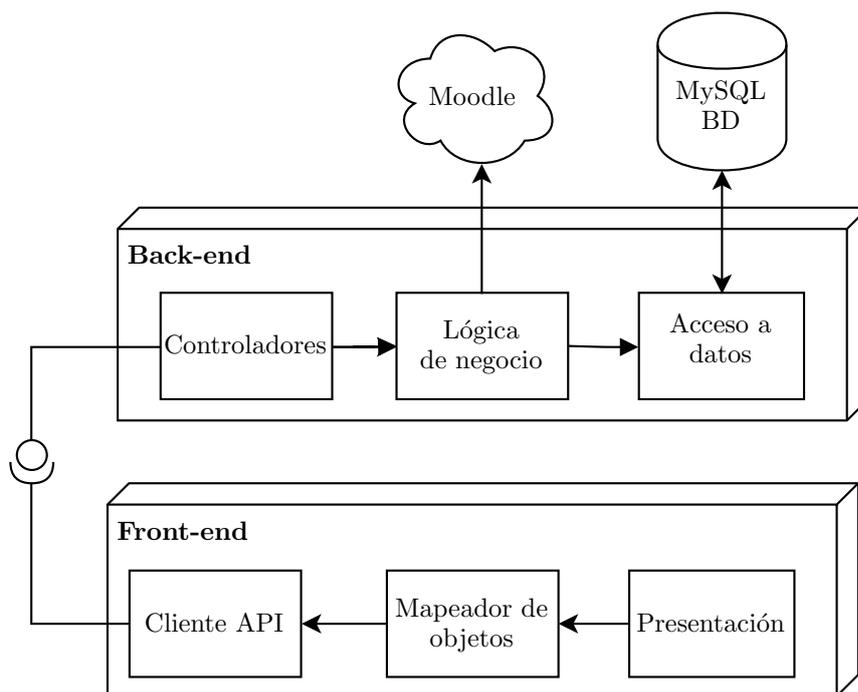


Figura 5.1: Arquitectura de la plataforma administrativa

5.2. Funcionalidades

Para el desarrollo de la plataforma se optó por utilizar tres roles para los usuarios: *nuevo* (*new*), *profesor* (*teacher*) y *administrador* (*admin*). Para los usuarios que acceden por primera vez, es necesario que un administrador active la cuenta con un rol que permita acceder al resto de las funcionalidades. Los usuarios con rol de profesor tienen acceso a todas las funcionalidades de visualización, mientras que los administradores también pueden acceder a funcionalidades de gestión.

Al ingresar a la plataforma el usuario tiene la posibilidad de acceder a las siguientes funcionalidades dependiendo de su rol:

- **Login:** El sistema cuenta con un mecanismo de autenticación integrado con Moodle, permitiendo a los usuarios ingresar con las mismas credenciales utilizadas en EVA.
- **Visualización de resumen:** Se listan los estudiantes que cursan o han cursado la materia para cada edición del curso, indicando para cada uno la nota final obtenida, o la última predicción realizada por el módulo de aprendizaje automático. También se incluyen gráficas que resumen la cantidad de pre-

dicciones realizadas, y la cantidad de accesos por parte de los estudiantes al EVA en la última semana.

- **Visualización de estudiante:** Se muestran los datos disponibles de un estudiante para una edición determinada del curso. Se detallan sus notas obtenidos, encuesta, últimos accesos al EVA e historial de predicciones realizadas. Para la última predicción, se muestra además el camino recorrido por el árbol de decisión correspondiente al modelo predictivo empleado.
- **Gestión de cursos:** La plataforma permite dar de alta nuevos cursos en el sistema, indicando para cada uno el año de edición, identificador interno utilizado en Moodle, y el nombre de la encuesta enviada a los estudiantes, en caso de haber una. También es posible configurar los nombres de las diferentes actividades realizadas en el curso (laboratorios, parciales y resultados finales) para sincronizar los datos desde Moodle.
- **Gestión de estudiantes:** Permite seleccionar estudiantes para ser ignorados en el listado de resumen del curso, de manera de poder eliminar aquellos que abandonen la materia luego de matricularse en la misma.
- **Sincronización de datos:** Los administradores del sistema pueden sincronizar los datos de un curso actualmente mostrados en el sistema, obteniendo los últimos valores desde Moodle. Los datos actualizados son: listado de estudiantes matriculados, notas obtenidas, y encuestas respondidas. También existe la posibilidad de cargar las notas de los estudiantes en formato de archivo CSV, si éstas no estuvieran disponibles desde Moodle.
- **Predicción de resultados:** Se invoca al módulo de aprendizaje automático a demanda, re-entrenando los modelos predictivos y almacenando nuevas predicciones para todos los estudiantes actualmente cursando la materia.
- **Extracción automática de registros de acceso:** El módulo de back-end corre un proceso recurrente cada 15 minutos que se encarga de extraer los registros de acceso de Moodle y guardarlos en la base de datos.

5.3. Tecnologías

El componente back-end de la solución está implementado con el lenguaje *TypeScript*, el cual no es un lenguaje ejecutable. Es necesario convertirlo primero a Javascript haciendo uso de la API de *TypeScript*. Una vez teniendo el archivo Javascript resultante, se corre el mismo en el entorno de ejecución de *node.js*.

Algunas de las bibliotecas o frameworks utilizados fueron las siguientes:

- *Express.js* [41]: Framework utilizado para construir la API del back-end. Levanta un servidor web que se encarga de escuchar los pedidos HTTP realizados a la plataforma.
- *Objection.js* [42]: Framework de mapeo entre entidades del negocio y objetos almacenados en la base de datos. Se encarga del almacenado y posterior obtención de las entidades del negocio en una base de datos de manera eficiente.
- *cheerio.js* [43]: Biblioteca que permite manejar data extraída de la web con facilidad utilizando sintaxis de *jQuery*. Utilizada en el proceso de logueo integrado con Moodle.
- *csvtojson* [44]: Biblioteca utilizada para el de archivos en formato CSV, utilizada para la carga de notas de los estudiantes y para la carga inicial de datos.

Por otro lado, el componente front-end está implementado en el lenguaje *TypeScript* (que luego es convertido a Javascript), CSS y HTML; y utiliza *React.js* como *framework*.

Algunas de las bibliotecas utilizadas en el módulo incluyen las siguientes:

- *chart.js* [45]: Biblioteca que facilita la creación y renderización de gráficos animados en el navegador.
- *randomcolor* [46]: Biblioteca que ayuda a generar códigos hexadecimales de color.

Capítulo 6

Extracción, transformación y carga de datos

Para comenzar el proceso de extracción, transformación y carga se tuvo que investigar sobre los indicadores que mayor utilidad brindan al objetivo del estudio y de donde se podían extraer. Identificar las fuentes de datos fue el primer paso del proceso. En primera instancia se contaba con planillas Excel creadas por los docentes del curso para los años 2016, 2017 y 2018. Luego se investigó Moodle, que cuenta con una API REST de la cual se puede extraer información.

La gran mayoría de los datos utilizados en estudios anteriores no estaban disponibles en los documentos de notas de cursos anteriores. Datos demográficos sobre los estudiantes como la edad; si asiste o no a clase; si trabaja o no; si se formó en educación pública o privada; etc, resultaban de difícil acceso al ser privados y no estar disponibles en ninguna base de datos o recursos a los que se tenga acceso públicamente.

Los datos a los que se tuvo acceso directamente fueron la CI, nombres e información relevantes a los cursos de las ediciones 2016, 2017 y 2018 del curso; como notas de parciales, resultados de laboratorios y re-entregas. Con estos datos básicos ya se podía empezar a trabajar en modelos de aprendizaje automático pero de todas maneras, dados los indicadores utilizados para los antecedentes estudiados se quiso intentar obtener información más detallada, con el fin de obtener mejores modelos de predicción. Es por esta razón que se creó una encuesta pública para que los estudiantes ingresen sus datos.

Para un proyecto como el presente en el cual es clave tener una buena calidad de datos es necesario definir un proceso ETL y una estructura que nos permita cumplir con los objetivos del mismo. Por lo tanto, a la hora de diseñar el proceso ETL se notó que contar con una etapa de staging facilitaría el proceso de carga a la base de datos del sistema. Es entonces que se crea una base de datos de staging a la que se cargan los datos extraídos desde las fuentes identificadas. Una vez están

almacenados todos los datos, se aplica otro proceso ETL donde la extracción se hace desde la base de datos de staging y la información se transforma para poder ser cargada en la base de datos del sistema con la estructura correcta. La Figura 6.1 refleja el flujo del proceso.

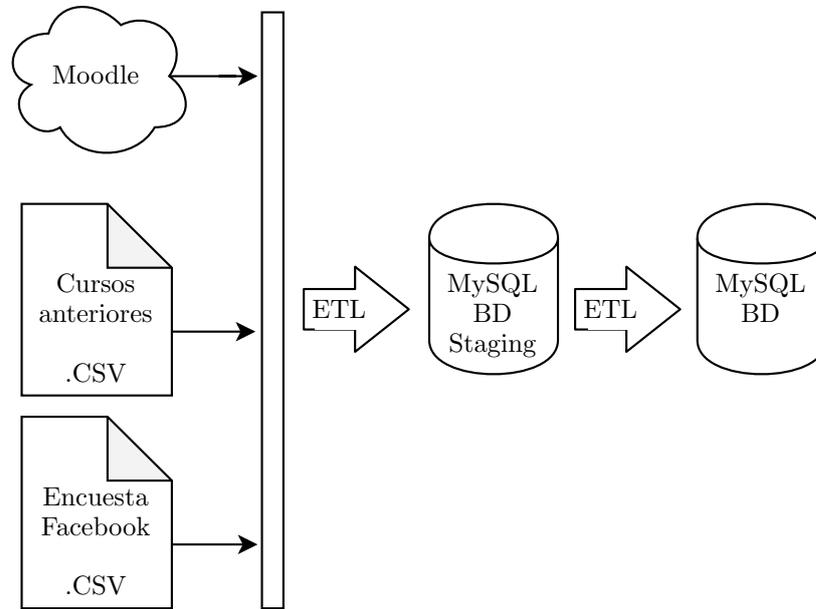


Figura 6.1: Diagrama del proceso ETL

6.1. Encuesta demográfica

Con el objetivo de obtener información demográfica de estudiantes del curso 2018 y anteriores, se realizó un encuesta en el foro de Facebook de facultad, que cuenta con un gran número de estudiantes. Se agregó al formulario un campo para agregar la cédula de identidad y el año en que se cursó la materia, con el fin de identificar al estudiante en los documentos de notas y asociarlo con sus resultados. Un estudiante podría contestar varias veces para distintos años.

Las respuestas de la encuesta son registradas por Google en un documento CSV, que luego es cargado en la base de datos de staging a través de un proceso de transformación. A las cédulas se les removió cualquier símbolo que no sea número y fueron truncadas en los siete caracteres con el fin de normalizar el formato de las cédulas ingresadas. Notar que al truncar en siete caracteres se pierde el dígito verificador, que no es necesario. El resto de los datos fueron cargados en la base de datos sin cambios.

Las preguntas de la encuesta fueron las siguientes:

- ¿Qué edad tenés?
 - 18-20 años
 - 21-24 años
 - 25 o más
- ¿Sos de Montevideo o del Interior?
 - Montevideo
 - Interior
- ¿En qué tipo de institución cursaste bachillerato?
 - Privada
 - Pública
 - U.T.U.
- ¿En qué año cursaste Prog. 2?
 - 2016
 - 2017
 - 2018
- ¿Trabajaste mientras cursaste Prog. 2?
 - No
 - Si, Part-Time
 - Si, Full-Time
- Si trabajaste, ¿Tu trabajo estaba relacionado con la carrera? (Opcional)
 - Sí
 - No
 - Vacío (no contesta)
- ¿Cuántas materias cursaste en dicho semestre? (Campo numérico libre)
- ¿Cuántas veces cursaste Prog. 2?
 - Una
 - Dos
 - Más de dos
- ¿Asististe a las clases teóricas?
 - Sí
 - No
 - A veces
- ¿Asististe a las clases prácticas?
 - Sí
 - No
 - A veces
- ¿Estudiaste solo o en grupo?
 - Solo
 - En grupo
 - Solo y En grupo
- ¿Cuántas horas aproximadamente le dedicaste a Prog. 2 por semana?
 - 3 o menos
 - Entre 3 y 6
 - 6 o más
- ¿Cómo definirías tu nivel de motivación con Prog. 2?
 - Alto
 - Medio
 - Bajo
- ¿Qué resultado obtuviste?
 - Recursé
 - Derecho a examen
 - Exoneré
- CI (Opcional) (Campo de texto libre)

Análisis de los datos obtenidos

Debido a que se trata con una fuente de información menos confiable al haber sido recopilada a través de Facebook, es necesario una verificación exhaustiva de los datos para evitar comprometer el análisis de los mismos en las etapas posteriores del proyecto.

Se cuenta con un total de 301 respuestas de las cuales 246 incluyeron la cédula de identidad. Los datos fueron ingresados en el marco de una encuesta con opciones de respuestas predefinidas, lo que limita la posibilidad de errores de correctitud. No obstante, se pidió explícitamente responder una única vez por la última vez que se cursó la materia, por lo cual es de interés encontrar casos de respuestas duplicadas.

Luego se procedió a definir criterios de calidad para evaluar la confianza de las respuestas. El objetivo principal es comparar las respuestas de la encuesta con los datos del curso para identificar inconsistencias entre las fuentes. Como medio para llevar a cabo el análisis, se utilizaron las tablas *student_surveys* que contiene los datos de las respuestas de las encuestas y *results* que contiene la información asociada a los resultados del curso de los estudiantes que cursaron la materia en los años contemplados en la encuesta, ambas obtenidas del proceso de ETL.

Respuestas duplicadas

Se quiere obtener la cantidad de respuestas con la misma cédula de identidad. Se observa que hubo cinco estudiantes que respondieron dos veces la encuesta. Se verificó manualmente y se llegó a que uno de ellos respondió respuestas totalmente distintas mientras que los cuatro restantes respondieron ambas veces de manera consistente.

Se resolvió descartar ambas respuestas inconsistentes así como eliminar las respuestas duplicadas de los cuatro restantes.

Respuestas con CI: 246 (total) - 2 (inconsistentes) - 4 (duplicadas) = 240
--

Cédulas inválidas

Se quiere obtener la cantidad de respuestas cuyas cédulas corresponden a una cédula válida en los cursos 2016/2017/2018 (teniendo en cuenta respuestas duplicadas y múltiples cursadas). Para esto se unificó la tabla de las encuestas y la tabla de estudiantes del curso por cédula.

Se observa que 216 respuestas corresponden a cédulas válidas registradas en el curso.

Respuestas con CI: 216

Respuestas inconsistentes

Se quiere obtener la cantidad de respuestas que no coincidan con los datos reales del curso. Para ello se compara la respuesta de resultado obtenido con el registro del resultado en el curso.

Se observa que en 29 respuestas no corresponden el resultado respondido con el obtenido.

216 (sin duplicados ni inválidos) - 29 (inconsistentes) = 187
--

Se concluyó que luego de descartar las respuestas que no cumplieron con los criterios de calidad definidos, se cuentan con 187 respuestas válidas para ser utilizadas al momento del análisis.

Cabe mencionar que este análisis se llevó a cabo únicamente para la encuesta hecha a los estudiantes que cursaron la materia en 2018 o antes. Para la edición del 2019, la misma encuesta fue realizada como una actividad opcional para los estudiantes directamente en el EVA, donde no es necesario el chequeo de respuestas duplicadas ni la validación de cédulas de identidad.

6.2. Documentos de años anteriores

Para el caso de los archivos de los cursos 2016, 2017 y 2018, se tenía que ninguno de los tres contaba con la misma estructura, nombre de columnas, e incluso la notación utilizada para los resultados de laboratorio variaban año a año. Esto dificultó el procesamiento automático de los archivos, por lo que la carga de esos datos al sistema fue de forma manual para luego poder transformar los datos de una forma automatizada. El proceso consistió en copiar y pegar las columnas de las planillas Excel que interesaban a una nueva planilla y se utilizó la función de exportar a CSV de *Google Spreadsheets*. El archivo luego es cargado a la base utilizando *MySQL Workbench* con la función de importar CSV, previamente creando una tabla auxiliar que cumpla con la forma del CSV. Por cada curso se crea una tabla donde se almacenan los resultados obtenidos por un estudiante de forma unificada.

Los datos relevados de los archivos fueron:

- | | |
|--|--|
| <ul style="list-style-type: none">■ Cédula de Identidad■ Nombres y Apellidos■ Notas del primer parcial■ Notas del segundo parcial■ Notas final del curso | <ul style="list-style-type: none">■ Resultados del laboratorio del primero al quinto■ Resultados de re-entregas de laboratorio del primero al quinto■ Nota final del laboratorio |
|--|--|

Para futuras ediciones del curso se definió un formato estándar para cargar la información, con el objetivo de que se pueda cargar al sistema por un docente a través de la aplicación web construida. Este formato se define en el Anexo A.

Los datos obtenidos pasaron por un proceso de conversiones y normalizaciones. A diferencia de las respuestas de la encuesta demográfica, las cédulas de identidad en estos documentos contaban con el formato correcto por lo que no fue necesario normalizarlas. Se utilizó este dato para identificar a los estudiantes y poder completar en una sola tabla con toda su información del curso en cuestión. Esta información estaba distribuida en varios archivos distintos por lo que fue necesario cargar datos a la base y realizar consultas SQL para unificar la información en una sola tabla.

Los nombres de los estudiantes fueron registrados de distintas formas en distintos documentos. En algunos, era “NOMBRES APELLIDOS”; en otros “APELLIDOS, NOMBRES”; y en ambos casos variaba si se usaba o no segundo apellido o segundo nombre, y también se variaba constantemente el uso de mayúscula sólo al principio, o en todos los caracteres. Con vistas a futuro de saber diferenciar donde terminan los nombres o los apellidos, se adoptó el formato “APELLIDOS, NOMBRES” de aquellos documentos donde se denotaba claramente que se utilizaban

todos los nombres y apellidos de los estudiantes. También se normaliza la forma en que se escriben, utilizando sólo mayúsculas.

En el caso de los resultados de las entregas y re-entregas del laboratorio hubo varias consideraciones. Para empezar, los resultados en crudo son de la forma de unos, ceros y otros caracteres cuyo significado cambia entre las distintas entregas. Por ejemplo en algunos casos se utiliza la letra “N” para indicar que no entregó, mientras que otros casos se utiliza un simple cero. De la misma manera, para indicar que no se pasó una prueba de stress en algunos casos se utiliza “C” y en otros “T”. Otro problema que se presenta es que año a año, si bien la cantidad de entregas no cambia (siempre son cinco), la cantidad de casos de prueba sí puede cambiar. Esto provoca que comparar las cadenas de unos y ceros año a año no aporte ninguna utilidad a la hora de comparar. Así como se convirtieron los datos de notas de parciales a porcentajes del total, se realiza lo mismo para los resultados de laboratorio, donde se toma la cantidad de unos sobre el total de caracteres. Notar que se está asumiendo que resultado del tipo “C”, “N” y “T” no son “1”, de manera que contarían como casos de prueba perdidos. De esta manera todos los resultados de entrega toman valores numéricos con un mismo significado semántico. El dato que se toma para cada instancia de laboratorio es el de la última entrega realizada por el estudiante o de la re-entrega en caso de existir una.

Análisis de los datos obtenidos

Se proporcionaron datos de los parciales y laboratorios de las ediciones de los años 2016 y 2017. Para el año 2018, se utilizaron los datos obtenidos directamente desde Moodle, con la limitación de que al momento de consultarlos había un conjunto de estudiantes ya desmatriculados del curso. La cantidad de datos para cada año se muestra en la tabla 6.2.

Cabe mencionar que no fue posible obtener registros sobre los accesos a los cursos por parte de los estudiantes de años anteriores al 2018, al ser datos que se almacenan en la base de datos de Moodle y se eliminan luego de transcurrido un año.

	2016	2017	2018
Recurso	114	142	52
Derecho a examen	99	54	61
Exonera	156	136	120
Total	369	332	233

Tabla 6.2: Datos de cursos anteriores

6.3. Ambiente de pruebas

La docente responsable realizó una copia del curso de 2018 para utilizar como ambiente de pruebas en la plataforma del EVA. Se dio acceso como profesor a los integrantes de este grupo de proyecto de grado con el objetivo de utilizar los datos del mismo. Sin embargo, esto no fue posible dado que en el curso copia no estaban matriculados los estudiantes por lo que faltaban los resultados de las instancias de parciales y laboratorios, y los resultados de las encuestas. Dado este inconveniente, se dio el mismo permiso de acceso a los integrantes del grupo para el curso real de Programación 2. El mismo contaba con la mayoría de la información que se buscaba, no obstante, algunos de los estudiantes que habían realizado el curso ya no se encontraban matriculados en el mismo, por lo que no fue posible obtener sus datos.

Por otro lado, se contó con una copia de la información del curso, realizado por la Unidad de Recursos Informáticos de la Facultad. Una vez obtenido el archivo, se procedió a instalar Moodle [47] en un entorno local, utilizando uno de los paquetes instalables provistos en su sitio web oficial, los cuales se pueden ejecutar utilizando el entorno de ejecución gratuito MAMP [48], con el propósito de importar la información y analizar su contenido.

Una vez importada la copia del curso en el servidor local, se contó con una réplica de la base de datos de la plataforma, lo cual permitió la posibilidad de realizar consultas SQL directamente desde ella. Sin embargo, este enfoque no resultó viable, dado que una vez analizado el esquema a tratar, se encontró que éste tenía una gran complejidad, contando aproximadamente con 400 tablas y escasa documentación pública de la cual poder estudiar.

6.4. Servicios web de Moodle

Como fuente de datos también se encontró que la plataforma Moodle cuenta con una API REST, cuyas funcionalidades se encuentran listadas en el sitio web de la herramienta [49]. Cabe mencionar que dicha API no cuenta con una gran documentación pública disponible, lo cual implicó un trabajo en base a prueba y error, con un fuerte apoyo en foros de la comunidad de Moodle.

Para poder ejecutar las funciones expuestas por la API fue necesario contar con permisos de “Profesor” para el curso de Programación 2, que fueron otorgados por la docente responsable del curso. Una vez obtenidos los permisos necesarios, se procedió utilizando el programa Postman [50] para obtener un token de acceso de Moodle y así poder acceder a las funcionalidades web provistas.

La API permitió obtener datos de valor relacionados al curso de manera estructurada. Las funciones utilizadas fueron:

- *core_enrol_get_enrolled_users*: devuelve el listado de estudiantes matriculados a un curso determinado. Resultó útil para realizar la unión de información de estudiantes del curso con los archivos de resultados y con los registros del curso, al contar con el nombre, apellido y cédula de identidad de cada estudiante de manera estructurada. Un ejemplo de respuesta de este servicio se muestra en el JSON D.1, en el Anexo D.

De estos datos disponibles se utilizan *firstname* (nombre), *lastname* (apellido), *email*, *idnumber* (CI del estudiante), *profileimageurl* (URL a la imagen del estudiante) y finalmente el *id* con el cual se identifica al estudiante dentro de Moodle. El resto de campos no son utilizados. Cabe destacar que este servicio también retorna información de los profesores anotados al curso. Fue necesario aplicar un filtro que tome solo aquellos usuarios que tengan rol de estudiante, es decir, que exista en *roles* un rol donde *shortname* sea *student* (como sucede en la Figura 6.2).

- *gradereport_user_get_grade_items*: provee el detalle de los resultados de los estudiantes matriculados a determinado curso, incluyendo ambos parciales, instancias de entrega y re-entrega de los laboratorios, el examen y el curso en general. Esta información resultó de utilidad para verificar la integridad de los datos obtenidos de las planillas Excel de los resultados antes mencionados. Cabe mencionar que al estar limitados a los resultados de los estudiantes matriculados al curso, no fue posible obtener la información de aquellos estudiantes ya desmatriculados al momento de realizar la carga de datos. De todos modos, la función puede ser utilizada para ediciones futuras del curso contando con la totalidad de estudiantes matriculados. Un ejemplo de respuesta de este servicio se muestra en el JSON D.2, en el Anexo D.
- *mod_feedback_get_feedbacks_by_courses*: devuelve todas las encuestas creadas para un curso en particular. Se utiliza este servicio porque, si bien los datos de encuesta que se utilizaron para determinar los modelos de predicción son aquellos obtenidos mediante la encuesta demográfica en Facebook, es de interés obtener dichos datos de una manera más formal. Por lo tanto, se creó una encuesta en el curso de Programación 2 para que los estudiantes contesten en los años futuros. Para el curso del 2019 ya se está utilizando la misma. Un ejemplo de respuesta de este servicio se muestra en el JSON D.3, en el Anexo D.
- *mod_feedback_get_responses_analysis*: devuelve las respuestas a una encuesta en particular. Un ejemplo de respuesta de este servicio se muestra en el JSON D.4, en el Anexo D.

```

1  [
2    [
3      [
4        "20/03/2019 21:30",
5        "Juan Pérez",
6        "-",
7        "Curso: Prog2 Programación 2",
8        "Sistema",
9        "Curso visto",
10       "The user with id '30377' viewed the section number '8' of the
11       ↪ course with id '132'.",
12       "web",
13       "127.0.0.1"
14     ],
15     [
16       "20/03/2019 23:39",
17       "Juan Pérez",
18       "-",
19       "Foro: Tarea 1",
20       "Foro",
21       "Módulo de curso visto",
22       "The user with id '30377' viewed the 'forum' activity with course
23       ↪ module id '85761'.",
24       "web",
25       "127.0.0.1"
26     ],
27     ...
28   ]
29 ]

```

Figura 6.2: Ejemplo de respuesta del servicio web de registros de actividad

6.5. Registros del curso en Moodle

Otra fuente de datos utilizada surge de la funcionalidad existente en Moodle para obtener registros estructurados de la actividad de los estudiantes en el curso. Estos registros detallan en qué instantes en el tiempo los estudiantes realizan determinadas acciones, como visualizar un mensaje de un foro, realizar una consulta, entre otras. Cabe mencionar que esta funcionalidad no es una de las que se encuentran expuestas en la API REST de Moodle, sino que fue encontrada directamente en el sitio EVA del curso, una vez obtenidos los permisos necesarios para acceder a la misma. Se puede observar un ejemplo de la respuesta que brinda Moodle en la Figura 6.2.

De esta información disponible se toma la fecha y hora del registro, la categoría

(en el ejemplo se trata de *Curso visto* y *Módulo de curso visto*), y también es necesario realizar una extracción del identificador del usuario en la descripción del registro. Por ejemplo, en el primer registro del ejemplo, de la descripción "*The user with id '30377' viewed the section number '8' of the course with id '132'.*" se extrae el identificador 30377. Para conseguir esto fue necesario crear una expresión regular, ya que la descripción varía para cada categoría de registros.

Al obtener registros de todo un semestre, se notó que hay varias decenas de categorías de registros distintas. Para que esta información pueda tener un uso práctico es necesario agrupar en categorías de registros similares, con el fin de bajar el nivel de heterogeneidad y que sea mas simple para un usuario del sistema comprender el resultado. A su vez algunas categorías de registros, para un semestre completo, tenían cantidades demasiado pequeñas como para ser consideradas relevantes (debajo de diez registros), y se optó por excluirlas del sistema.

En total se tomaron cuatro categorías distintas de registros de actividad:

1. "ACCESS". Acceso a una sección del curso, sea cual sea.
2. "FILE_ACCESS". Se descarga un directorio en formato zip desde el sitio.
3. "FORUM_ACTIVITY". Una actividad en el foro. Como puede ser contestar en un hilo y crear una nueva discusión.
4. "SURVEY_RESPONSE". Se contesta una encuesta del curso.

La interpretación planteada entre las categorías y los tipos de registro se muestra en la Tabla 6.3.

Se encontraron dos inconvenientes a la hora de consumir el servicio de registros del curso. En primer lugar, el token de acceso a las funcionalidades de Moodle no resultó útil para acceder a la funcionalidad. Al tratarse de un servicio web utilizado directamente desde el sitio web del EVA, el mismo requería de una cookie que permitiera identificar al usuario solicitante como un usuario que cuente con los permisos necesarios. Para resolver esto, se implementó una automatización del proceso de login en el sitio y posterior extracción de los contenidos HTML obtenidos, de manera de obtener una cookie de forma automática sin necesidad de acceder directamente a un navegador web. Una vez obtenida la cookie fue posible obtener los registros en formato de archivo CSV.

Por otro lado el alto volumen de datos disponibles para descargar implicó una complicación, dado que la funcionalidad necesita aplicar un filtro para descargar los registros por lotes. Dicho filtro podría ser uno de varios: obtener los registros de determinado estudiante o grupo de laboratorio, de determinada acción en el sitio, en determinada fecha, entre otros. Con el objetivo de automatizar la descarga de todos los registros disponibles, se optó por el filtro de la fecha de la acción, la cual

Correspondencia de categorías	
ACCESS	CURSO VISTO
	MODULO DE CURSO VISTO
	TEMA VISTO
	INFORME DE NOTAS DE USUARIO VISTO
	SE HA VISUALIZADO EL ESTADO DE LA ENTREGA
	FORMULARIO DE ENTREGA VISTO
	INSTANCIA DEL MÓDULO DEL CURSO VISUALIZADA
	INFORME USUARIO DEL CURSO VISTO
	PERFIL DE USUARIO VISTO
	LISTA DE USUARIOS VISTA
	INFORME DE RESUMEN DE NOTAS VISTO
FORUM_ACTIVITY	ALGÚN CONTENIDO HA SIDO PUBLICADO
	MENSAJE CREADO
	TEMA CREADO
FILE_ACCESS	DIRECTORIO DESCARGADO EN FORMATO ZIP
SURVEY_RESPONSE	RESPUESTA ENVIADA

Tabla 6.3: Tabla de correspondencia entre categorías de registros de actividad

se consideró la más sencilla de implementar de manera genérica, y porque permite la descarga de registros diarios para futuras ediciones del curso.

Capítulo 7

Módulo de Aprendizaje Automático

En este capítulo se presenta una descripción mas detallada del módulo de Aprendizaje Automático, sus funcionalidades, su interacción con la Plataforma Administrativa, y la experimentación realizada para su diseño e implementación. El módulo es el responsable del entrenamiento de los modelos utilizados para predecir los resultados de los estudiantes, así como la generación de las imágenes de los árboles de decisión construidas para cada predicción.

7.1. Arquitectura

En la Figura 7.1 se detalla la arquitectura interna del módulo de aprendizaje automático. El mismo es dividido en componentes con distintas responsabilidades.

- **Componente de Entrenamiento**
Es el encargado de entrenar todos los modelos y persistirlos en el sistema de archivos.
- **Componente de Predicción**
Es el encargado de obtener el modelo correspondiente a un estudiante en base a los datos que se tengan disponibles del mismo, para posteriormente predecir. Los modelos son cargados desde el sistema de archivos.
- **Generación de Imágenes**
Es el componente utilizado para generar y cargar las imágenes de los árboles de decisión.

- **Controladores**

Es un componente utilizado para exponer los servicios que provee el módulo de aprendizaje automático.

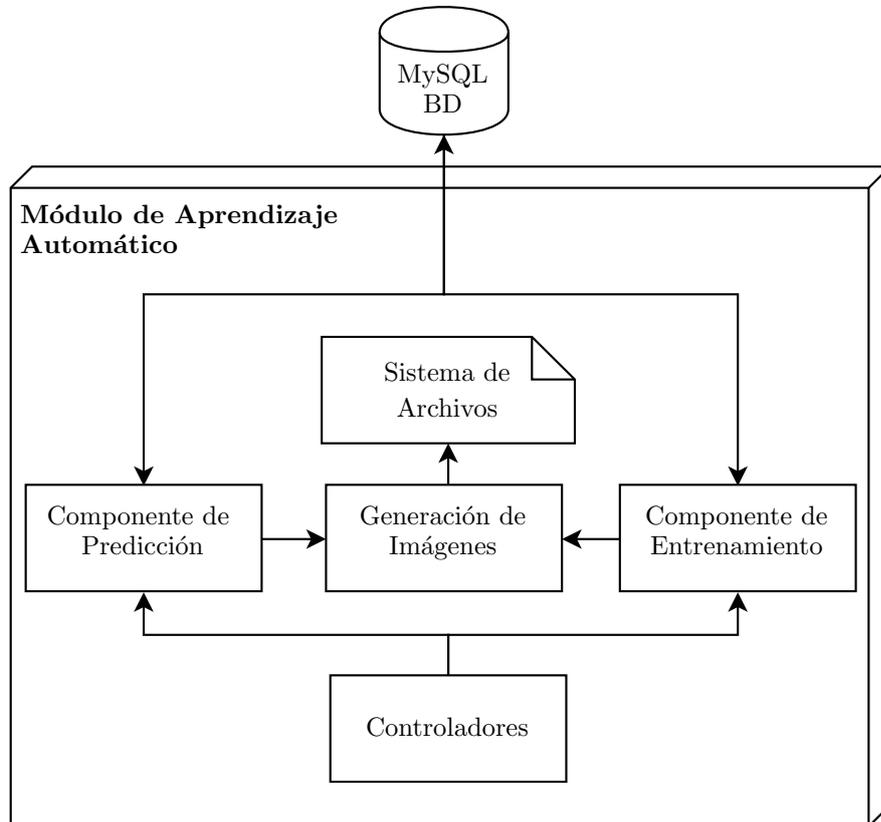


Figura 7.1: Arquitectura del Módulo de Aprendizaje Automático

7.2. Funcionalidades

El módulo cuenta con tres servicios. Cada uno de éstos es responsable de una funcionalidad que se expone para ser utilizada por la Plataforma Administrativa. En las siguientes secciones se detallan las funcionalidades: "Entrenamiento de modelos variables para los estudiantes", "Predicción de resultados", "Generación del árbol de decisión de un estudiante" del módulo.

7.2.1. Entrenamiento de modelos variables para los estudiantes

El objetivo de esta funcionalidad es entrenar los distintos modelos necesarios para ser posteriormente utilizados al momento de predecir los resultados de los estudiantes durante el transcurso del curso lectivo. Para esto, el primer paso es obtener de la base de datos todos los estudiantes de cursos anteriores, ya que de ellos se conocen todos sus datos incluyendo el resultado final obtenido.

Luego, se itera sobre todos estudiantes y se generan los distintos modelos para cada tipo predefinido en base a la información disponible de los mismos. Esto se debe a que para ciertos cursos no fue posible obtener los registros de acceso correspondientes y también al hecho de que no todos los estudiantes respondieron la encuesta.

Por último, se serializan los modelos generados en una secuencia de bytes para poder ser persistidos en el sistema de archivos del ambiente donde se encuentra instalado el sistema. Esto con el objetivo de ser obtenido y des-serializado en su posterior uso al momento de predecir.

Una aclaración importante con respecto a los datos utilizados para el entrenamiento de los modelos, es que para los registros de accesos a Moodle se toma un promedio de la cantidad de accesos semanales de cada estudiante. Esta decisión de implementación se debe a la diferencia en las cantidades de accesos que se veían en diferentes etapas de la asignatura por parte de los estudiantes, y a la necesidad de obtener un valor que pueda ser utilizado para entrenar modelos en cualquiera de estas etapas, de la manera más equitativa posible.

7.2.2. Predicción de resultados

El objetivo de esta funcionalidad es utilizar los modelos generados previamente para predecir el resultado de los estudiantes en el correr del curso. Por lo tanto para el correcto funcionamiento se tiene como pre-condición haber ejecutado la funcionalidad de entrenar modelos previamente.

Contrariamente a la funcionalidad de entrenamiento de modelos, se comienza obteniendo de la base de datos todos los estudiantes del curso actual, ya que son aquellos para los que no se conoce el resultado final.

Luego, para cada estudiante se elige el modelo correspondiente a la altura del curso en que se quiere predecir. Esta elección se hace en base a la información parcial con la que se cuenta así como también si respondió la encuesta o no. Finalmente se predicen los resultados de los estudiantes utilizando los modelos correspondientes y se persisten las predicciones en la base de datos.

La funcionalidad está diseñada para ser utilizada reiteradamente a lo largo del avance del curso, a medida que se cuenta con mayor información desde Moodle y

lograr predicciones con cada vez más datos de entrada.

7.2.3. Generación del árbol de decisión de un estudiante

El objetivo de esta funcionalidad es utilizar los modelos generados previamente para generar y devolver el árbol de decisión con el cual fueron predichos los resultados de los estudiantes. Por lo que en este caso también se tiene como pre-condición haber ejecutado la funcionalidad de entrenar modelos previamente.

Como parámetro de entrada se tiene la cédula del estudiante del cual se quiere obtener el árbol a modo de identificador. Con este se obtiene de la base de datos los datos del estudiantes así como el modelo con el cual fue predicho.

Finalmente se genera la imagen del árbol de decisión, se guarda en el sistema de archivos y se retorna en formato png. Esto es para que en caso de ser invocada la funcionalidad posteriormente para el mismo estudiante no se tenga que generar nuevamente y se pueda retornar la imagen ya previamente guardada.

7.3. Tecnologías

El módulo de aprendizaje automático fue implementado sobre Django, que es un framework de aplicaciones web gratuito y de código abierto escrito en Python [51]. La principal razón de esta decisión fue justamente por el lenguaje Python, el cual posee características particularmente atractivas para el desarrollo de sistemas de aprendizaje automático.

7.3.1. Lenguaje de programación

Una de las grandes ventajas que ofrece Python sobre otros lenguajes de programación es lo grande y prolifera que es la comunidad de desarrolladores que lo rodean. Comunidad que ha contribuido con una gran variedad de bibliotecas de primer nivel que extienden la funcionalidades del lenguaje en particular para el caso de aprendizaje automático.

7.3.2. Bibliotecas de Python utilizadas

Para implementar las diferentes funcionalidades previamente descritas, se hizo uso de diferentes bibliotecas. A continuación se listan y describen las más relevantes.

- *Scikit-Learn*

Scikit-Learn es la principal bibliotecas que existe para trabajar con aprendizaje automático. Incluye la implementación de un gran número de algoritmos de aprendizaje. Posee una API que es consistente en todos los modelos y se integra muy bien con el resto de los paquetes científicos que ofrece Python. Esta biblioteca también facilita las tareas de evaluación, diagnóstico y validaciones cruzadas ya que nos proporciona varios métodos de fábrica para poder realizar estas tareas en forma muy simple.

- *Graphviz*

Graphviz es un software de visualización de grafos de código abierto. Provee una manera de representar información estructurada como diagramas de grafos abstractos y redes. Fue de gran utilidad para generar las imágenes de los árboles y tiene la ventaja de contar con una interfaz fácilmente integrable con Scikit-Learn [52].

- *Pickle*

El módulo Pickle permite almacenar casi cualquier objeto Python directamente en un archivo. Lo que el módulo realiza es lo que se denomina serialización de objetos, es decir, convertir objetos a secuencias de byte para su almacenamiento o transmisión. El flujo de bytes puede escribirse en un archivo, y ser restaurado en un punto posterior. Esto resultó útil a la hora de persistir y retornar los modelos generados por el módulo de Scikit-Learn [53].

7.4. Experimentación

7.4.1. Composición del conjunto de datos para pruebas

El conjunto de datos está compuesto por los estudiantes de los cursos de Programación 2 (2016/2017/2018), los distintos datos con los cuales se puede contar en un momento dado del curso:

- Resultados de laboratorios: 5 entregas en total.
- Resultados de pruebas parciales: 2 parciales en total.
- Encuesta: en caso de que el estudiante haya respondido la encuesta realizada, se agregan sus respuestas al modelo.
- Registros de accesos a Moodle: Se agrega para las ediciones del curso donde se cuenta con esa información.

7.4.2. Definición de modelos

Teniendo en cuenta que las predicciones pueden ser realizadas en diferentes momentos del curso, y que para algunos estudiantes puede no existir una respuesta a la encuesta o que no se cuente con la información de registros de accesos a Moodle, se decidió crear un conjunto de modelos de manera tal que se utilicen distintos modelos para un estudiante y un momento del curso dado.

Para esto se dividió en 4 modelos para cada momento del curso relevante, los cuales se muestran en la Figura 7.2.

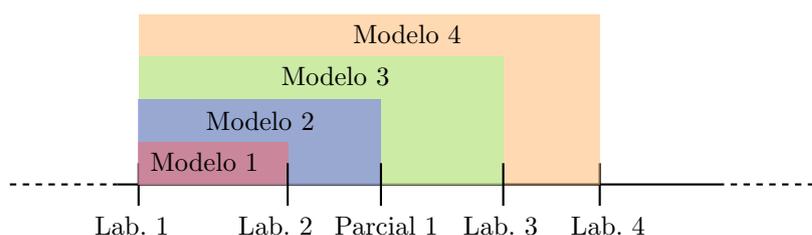


Figura 7.2: Línea de tiempo indicando el contenido de cada modelo

Notar que los modelos que contienen el laboratorio 5 y el segundo parcial no fueron tomados en cuenta, ya que para cuando se tiene los resultados de los mismos el curso ya ha finalizado. Tener una predicción a tal altura del curso no aporta valor.

Por otro lado, además de los cuatro modelos definidos anteriormente se tiene que para cada uno de ellos, el estudiante puede contar con respuestas de la encuesta y/o registros de accesos de la plataforma Moodle, dando un total de 16 modelos. En la Tabla 7.1 se observa la cantidad de estudiantes utilizada por cada modelo y además la cantidad por resultado final, se observa que para cada uno de los modelos las cantidades son iguales ya que los datos son completos (se cuenta con todas las entregas de estos estudiantes).

Composición de los distintos modelos				
Modelo	Total	Exoneran	Examen	Recursan
ModeloX	877	414	235	228
ModeloX + Logs	211	121	76	14
ModeloX + Encuesta	170	122	44	4
ModeloX + Logs + Encuesta	75	51	21	3

Tabla 7.1: Composición de los distintos modelos

7.4.3. Definición parámetros óptimos

La biblioteca *Scikit-Learn* provee distintos parámetros mediante los cuales se puede ajustar y mejorar los resultados obtenidos [54]. Los parámetros que se consideraron fueron:

- Criterio: función que permite medir la calidad de una separación.
- Separador: estrategia usada para separar cada nodo.
- Profundidad: camino máximo desde la raíz a la última hoja del árbol.

Con el objetivo de determinar los parámetros óptimos para la naturaleza del problema, se realizó un script que prueba todas las distintas combinaciones de los parámetros utilizados para entrenar con los distintos modelos. Se tomó como referencia el modelo 4 con registros de accesos y encuesta por ser el modelo con mayor cantidad de datos disponibles, utilizando un 60% de los datos para entrenar y el 40% restante para evaluar el modelo. A su vez se tomaron las profundidades del árbol entre 3 y 8, ya que menos de 3 el árbol se degenera y pierde mucha información, y con una profundidad mayor a 8 se dificulta mucho la lectura y manejo de los árboles. De esta forma se obtuvo la Tabla 7.2.

Modelo 4 con registros de accesos y encuesta						
Criterio	División atributos	Profundidad	F1-score	Accuracy	Recall	Precision
gini	best	3	0.489	0.6	0.5	0.5
gini	best	4	0.499	0.567	0.542	0.527
gini	best	5	0.444	0.533	0.458	0.472
gini	best	6	0.466	0.567	0.479	0.486
gini	best	7	0.422	0.567	0.417	0.437
gini	best	8	0.534	0.667	0.542	0.534
gini	random	3	0.603	0.667	0.667	0.611
gini	random	4	0.688	0.8	0.688	0.688
gini	random	5	0.712	0.867	0.667	0.929
gini	random	6	0.441	0.8	0.417	0.504
gini	random	7	0.656	0.767	0.667	0.649
gini	random	8	0.444	0.767	0.444	0.446
entropy	best	3	0.554	0.567	0.729	0.658
entropy	best	4	0.554	0.567	0.729	0.658
entropy	best	5	0.583	0.733	0.583	0.583
entropy	best	6	0.55	0.6	0.625	0.58
entropy	best	7	0.656	0.767	0.667	0.649
entropy	best	8	0.631	0.7	0.688	0.629
entropy	random	3	0.567	0.8	0.563	0.661
entropy	random	4	0.378	0.7	0.375	0.383
entropy	random	5	0.61	0.767	0.604	0.62
entropy	random	6	0.368	0.567	0.403	0.412
entropy	random	7	0.558	0.7	0.563	0.556
entropy	random	8	0.76	0.867	0.729	0.817

Tabla 7.2: Ejemplo de precisión con distintos parámetros para el modelo 4

En la tabla 7.2 se observa que las dos combinaciones con mejores resultados son (gini, random, 5) y (entropy, random, 8). Se decidió tomar (gini, random, 5) como los parámetros óptimos ya que cuenta con una profundidad menor, esto facilita la interpretación del árbol obtenido del modelo. En lo que refiere a tiempo de ejecución del algoritmo, al tener una cantidad de datos no muy grande, los tiempos de ejecución de los procesos de entrenamiento y predicción son menores a un segundo.

Utilizando los parámetros anteriormente descriptos se calculó la precisión de

cada modelo con el fin de ver que tan bien se desempeñan los modelos al tener más y menos datos para entrenar, en la tabla 7.3 se pueden observar los resultados.

Precisión de los distintos modelos				
Modelo	F1-score	Accuracy	Recall	Precisión
Modelo 1	0.504	0.647	0.574	0.457
Modelo 1 + Logs	0.312	0.588	0.337	0.331
Modelo 1 + Encuesta	0.472	0.735	0.496	0.489
Modelo 1 + Logs + Encuesta	0.567	0.8	0.563	0.661
Modelo 2	0.738	0.735	0.761	0.724
Modelo 2 + Logs	0.694	0.729	0.722	0.672
Modelo 2 + Encuesta	0.463	0.838	0.438	0.562
Modelo 2 + Logs + Encuesta	0.722	0.833	0.708	0.74
Modelo 3	0.722	0.707	0.744	0.713
Modelo 3 + Logs	0.656	0.788	0.665	0.657
Modelo 3 + Encuesta	0.62	0.765	0.61	0.64
Modelo 3 + Logs + Encuesta	0.603	0.667	0.667	0.611
Modelo 4	0.723	0.687	0.768	0.757
Modelo 4 + Logs	0.719	0.765	0.747	0.698
Modelo 4 + Encuesta	0.602	0.809	0.591	0.781
Modelo 4 + Logs + Encuesta	0.712	0.867	0.667	0.929

Tabla 7.3: Precisión de los distintos modelos

Se observa que el modelo 1 cuenta con los peores resultados, esto tiene sentido ya que se cuenta con menos información, los posteriores modelos cuentan con buenos valores de precisión mayores a un 50 %.

Luego de definir los parámetros adecuados para la naturaleza del problema, se enfrentó el problema de la visualización de los datos. Para esto se realizó una función encargada de leer y transformar los árboles obtenidos, esto permite que los árboles puedan ser interpretados de forma natural. Cada nodo cuenta con una regla (que se puede cumplir o no) y con las probabilidades de pertenecer a cada una de las distintas clases (Exonera, Examen o Recursa). Luego, cada nodo tiene un subárbol derecho y un subárbol izquierdo. Si se avanza por el derecho, se asume que la evaluación de la regla anterior dio como resultado «falso», y así sucesivamente hasta alcanzar una hoja.

A continuación se estudiarán los distintos árboles obtenidos a partir de los modelos anteriormente descritos. Para cada uno de ellos se mencionarán los resultados más relevantes. Para tener una mejor visualización se dividió algunos modelos en dos secciones, la izquierda y la derecha, ya que tienen un mayor tamaño.

Análisis Modelo 1 con encuesta

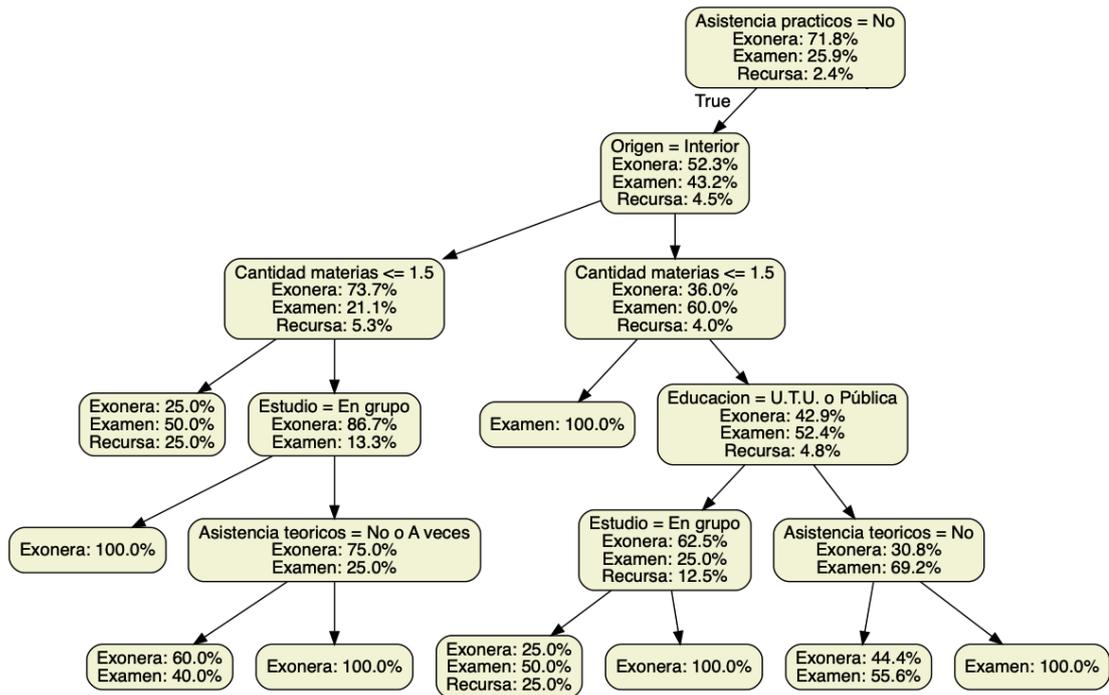


Figura 7.3: Modelo 1 con resultados de la encuesta, sección izquierda

En la Figura 7.3 se observa el árbol obtenido del modelo 1 con resultados de la encuesta sección izquierda. Para las conclusiones de este subárbol se asume que los estudiantes no concurren a los prácticos, las conclusiones más relevantes obtenidas son:

- Provenir del interior y cursar una materia o menos implica examen con un 50 %.
- Ser de Montevideo, cursar 2 o más materias y haber estudiado en la educación pública, sí no se estudia en grupo se exonera, en cambio sí se estudia en grupo las probabilidades de exoneración bajan a un 50 %.

- Ser de Montevideo, cursar 2 o más materias y haber estudiado en la educación privada, sí se asiste a teóricos se va a examen con un 55%, en cambio si se concurre a los teóricos se va a examen con un 100%.

Se puede concluir que estudiar en grupo, y concurrir a los teóricos para algún tipo de estudiante puede ser perjudicial para las posibilidades de exoneración.

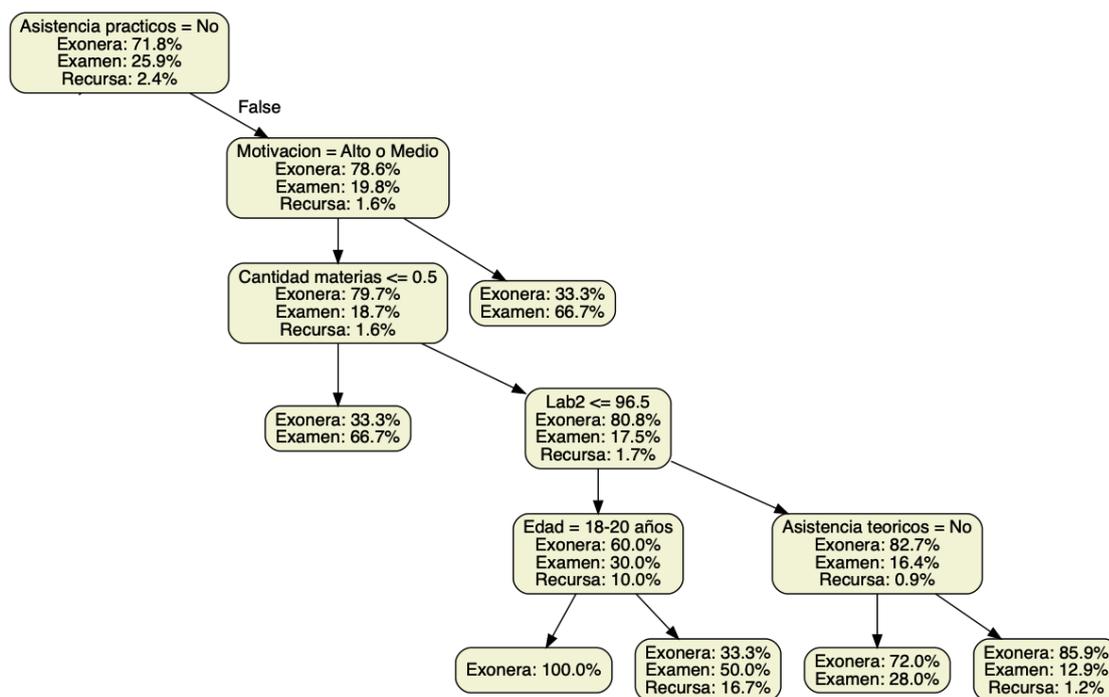


Figura 7.4: Modelo 1 con resultados de la encuesta sección derecha

En la Figura 7.4 se observa el árbol obtenido del modelo 1 con resultados de la encuesta, sección derecha. Para las conclusiones de este subárbol se asume que los estudiantes concurren a los prácticos, las conclusiones más relevantes obtenidas son:

- Si no se tiene un nivel de motivación alto o medio se va a examen con un 67%.
- Tener un nivel de motivación alta o medio y obtener menos de 96 en el lab 2, sí se tiene entre 18-20 años implica exonerar de lo contrario de va a examen con un 50%.
- Asistir a teóricos aumenta las posibilidades de exoneración de un 72% a un 86%.

En este modelo se presenta que la motivación es un factor determinante a la hora de obtener buenos resultados en el curso, también se observa que la edad es un factor importante, ya que tener una mayor edad disminuya las probabilidades de exoneración y finalmente se muestra que asistir a teórico aumenta levemente las posibilidades de exoneración.

7.4.4. Análisis Modelo 3 y encuesta

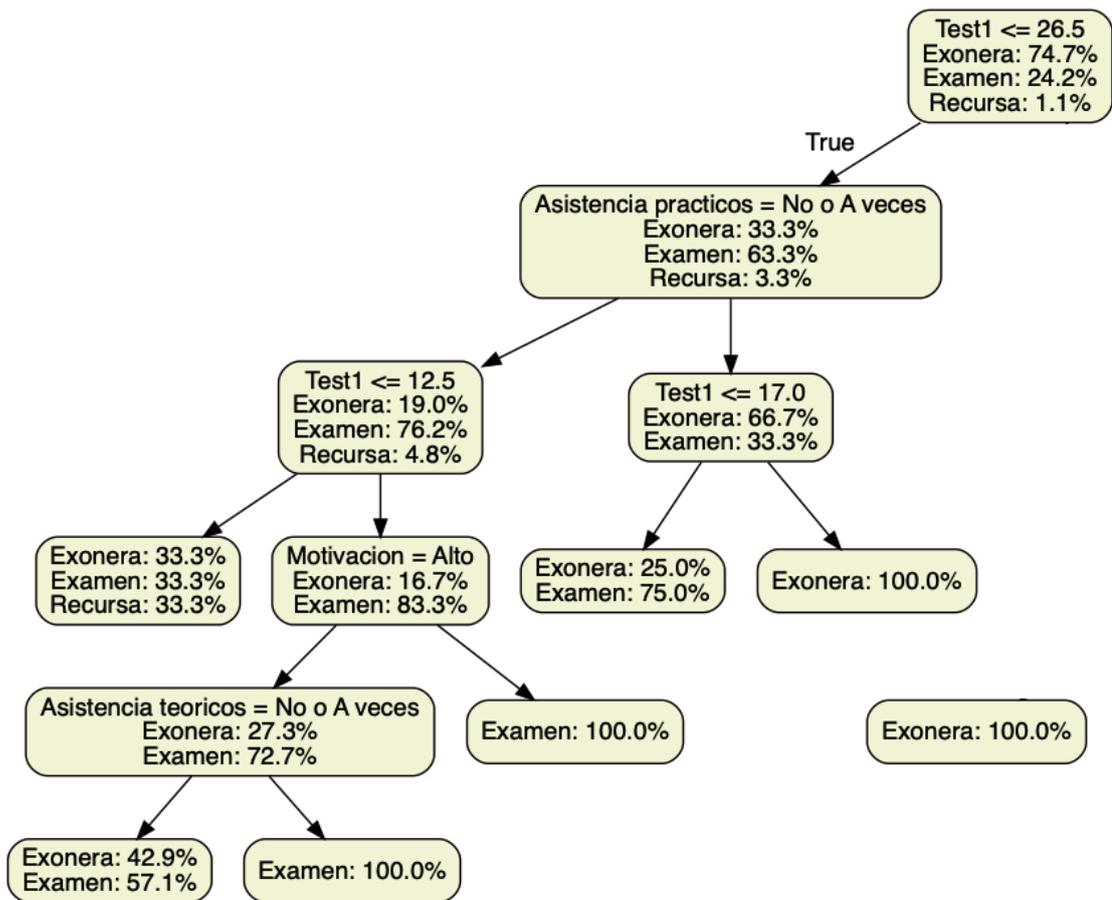


Figura 7.5: Modelo 3 con encuestas sección izquierda

En la Figura 7.5 se observa la sección izquierda del árbol obtenido del modelo 3 con resultados de la encuesta, para todos estos casos se considera que el estudiante obtuvo menos de 26 puntos en el parcial 1, las conclusiones más relevantes obtenidas son:

- Asistir a prácticos y obtener 17 puntos o mas en el parcial 1 favorece la exoneración.
- Obtener entre 13 y 26 puntos en el parcial 1 y no tener una motivación alta favorece irse a examen.

De este modelo la conclusión más importante que se puede extraer es que la asistencia a prácticos y la alta motivación tienen una influencia positiva en cuanto a los resultados del estudiante.

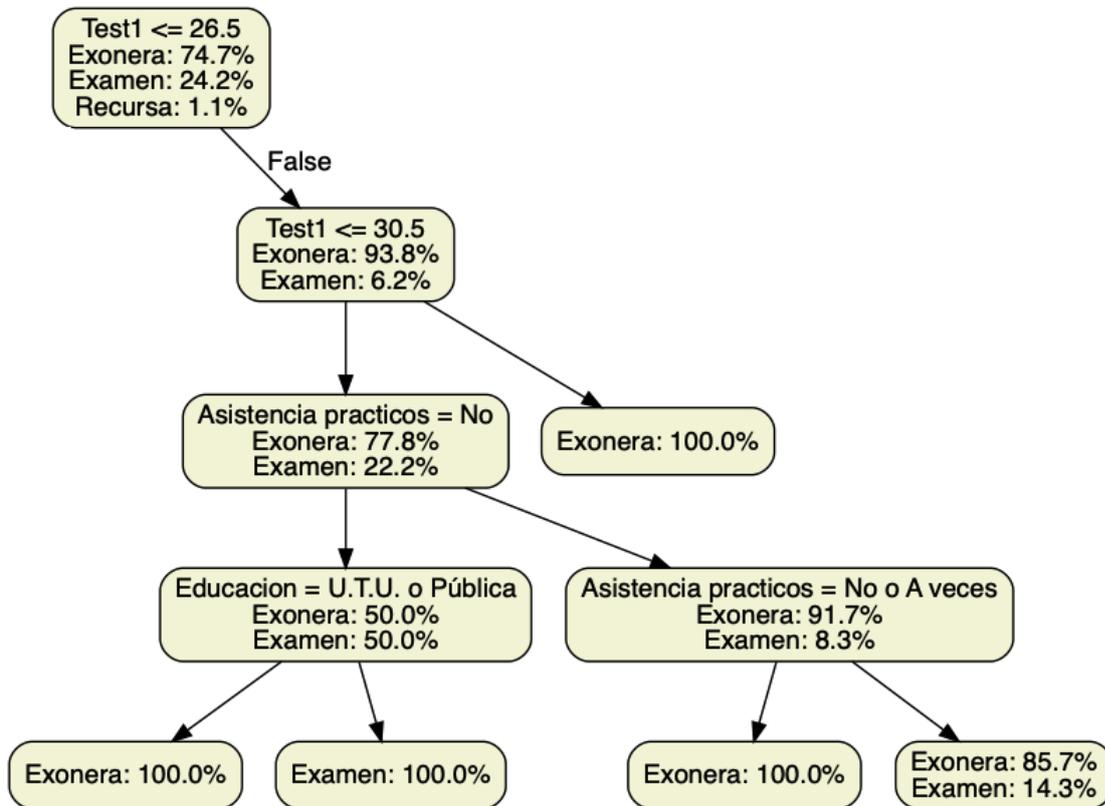


Figura 7.6: Modelo 3 con encuestas sección derecha

La sección derecha de este modelo es idéntica a la 7.9.

7.4.5. Análisis Modelo 4, registros accesos y encuesta

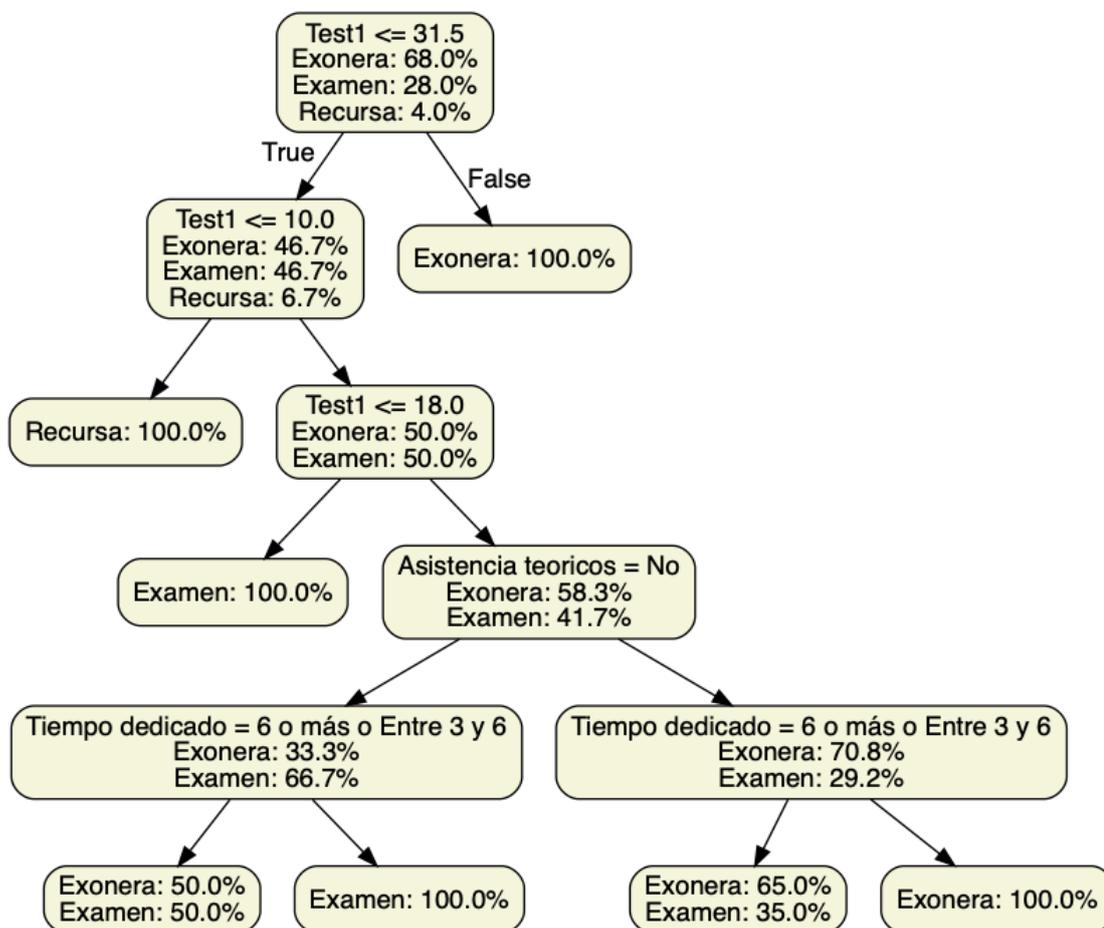


Figura 7.7: Modelo 4 con registros de acceso y resultados de la encuesta

En la Figura 7.7 se observa el árbol obtenido del modelo 4 con datos de registros de acceso y resultados de la encuesta. Las conclusiones más relevantes obtenidas son:

- Obtener 31 puntos o más en el parcial 1 favorece las posibilidades de exoneración.
- Obtener 10 puntos o menos en el parcial 1 favorece las posibilidades de recurrir.
- Obtener entre 18 y 10 puntos en el parcial 1 favorece las posibilidades de examen.

- Obtener más de 18 puntos en el parcial 1, asistir a los teóricos y dedicar más de 3 horas semanales favorece las posibilidades de exonerar.
- Obtener más de 18 puntos en el parcial 1, no asistir a los teóricos y dedicar menos de 3 horas semanales favorece las posibilidades de examen.

Se observa que los resultados obtenidos por este modelo son consistentes con la que se podría esperar. Los atributos que tuvieron mas peso fueron Test 1, Asistencia a teórico y tiempo dedicado, notar que este modelo también incluía registros de acceso pero estos fueron descartados por el algoritmo al no tener suficiente influencia en los resultados.

7.4.6. Análisis Modelo 4 y encuesta

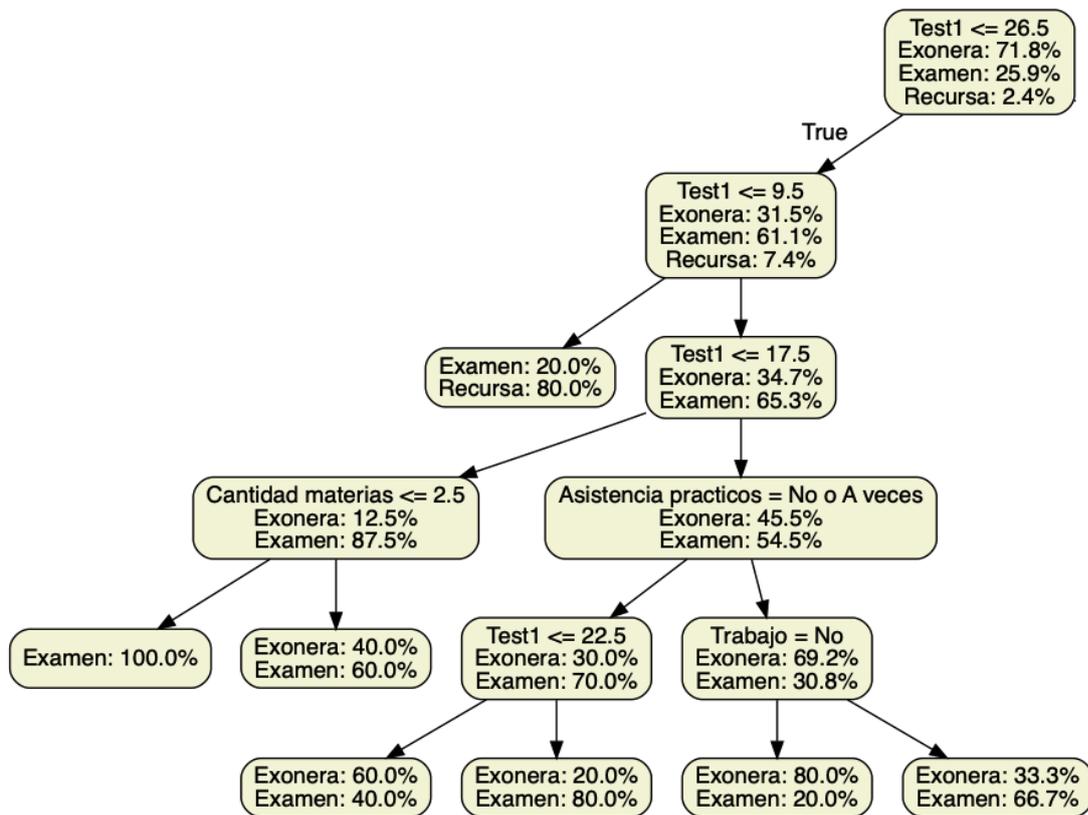


Figura 7.8: Modelo 4 con encuestas sección izquierda

En la Figura 7.8 se observa la sección izquierda del árbol obtenido del modelo 4 con resultados de la encuesta. Las conclusiones mas relevantes obtenidas son:

- Obtener menos de 10 puntos en el parcial 1 favorece las posibilidades de recurrar con un 80 %.
- Asistir siempre a prácticos y no trabajar implica un 80 % de posibilidades de exonerar, en cambio si se trabaja implica un 33.3 %.
- Asistir a practico mejora las posibilidades de exoneración de un 30 % a un 70 %

Se puede concluir de este modelo la importancia de asistir a prácticos, la asistencia mejora las posibilidades de exoneración, también se observa que trabajar es negativo para lograr la exoneración. Por otro lado obtener menos de 10 puntos en el parcial 1 prácticamente implica la perdida del curso.

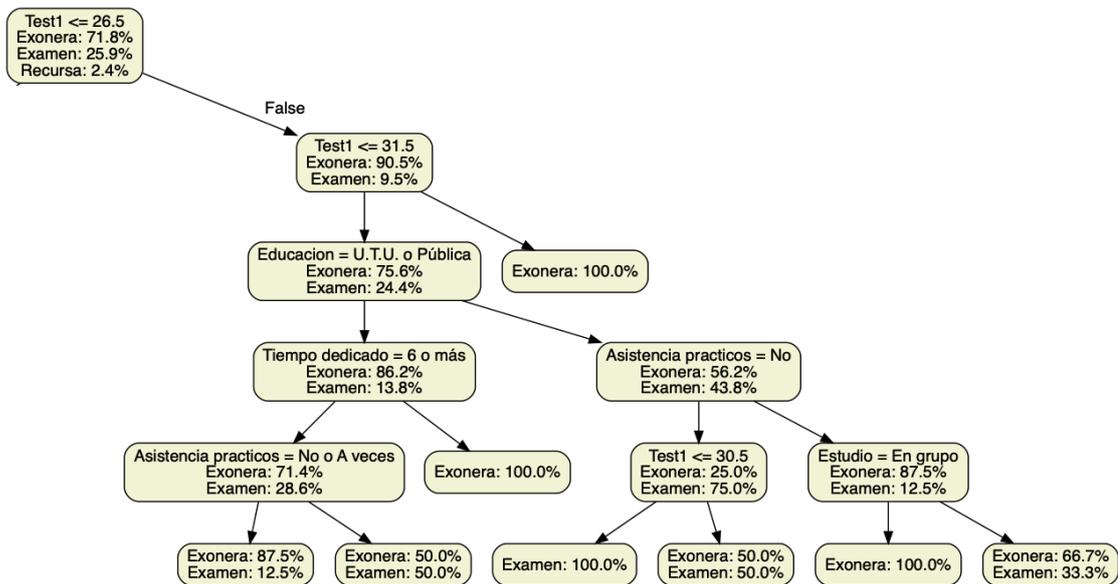


Figura 7.9: Modelo 4 con encuesta sección derecha

En la Figura 7.9 se observa la sección derecha del árbol obtenido del modelo 4 con resultados de la encuesta. Las conclusiones mas relevantes obtenidas son:

- Obtener más de 31 puntos en el parcial 1 favorece las posibilidades de exonerar
- Asistir a la educación privada, concurrir a las clases practicas y estudiar en grupo favorece las posibilidades de exonerar, si no se estudia en grupo las posibilidades de exoneración bajan a un 67

- Obtener entre 27 y 31 puntos en el parcial 1, asistir a la educación pública y dedicar menos de 6 horas favorece las posibilidades de exonerar.

En este modelo se puede observar que concurrir a las clases practicas y estudiar en grupo favorecen las posibilidades de exoneración. También se puede observar que aquellos estudiantes que dedican más de 6 horas, que pueden ser estudiantes con mayores dificultades para la comprensión o el aprendizaje de los temas, tienen menos probabilidad de exonerar.

7.4.7. Análisis Modelo 4 y registros de accesos

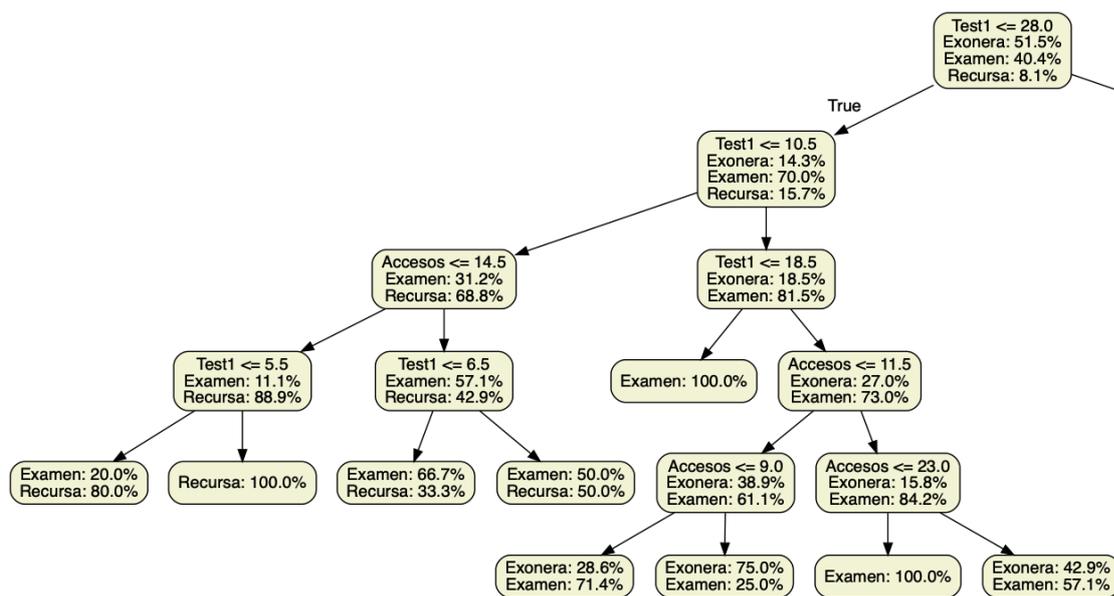


Figura 7.10: Modelo 4 con logs sección izquierda

En la Figura 7.10 se observa la sección izquierda del árbol obtenido del modelo 4 con datos de accesos, para todos estos casos se considera que el estudiante obtuvo menos de 29 puntos en el parcial 1, las conclusiones más relevantes obtenidas son:

- Obtener menos de 19 favorece las posibilidades de examen.
- Obtener entre 18 y 28 puntos en el parcial 1, tener 23 accesos semanales o menos implica examen y en cambio si se tienen más de 23 accesos semanales se exonera con un 43%.
- Obtener menos de 6 puntos en el parcial 1 implica recurrir.

La principal conclusión obtenida de este modelo es que el promedio de accesos semanales influye positivamente en el resultado del estudiante.

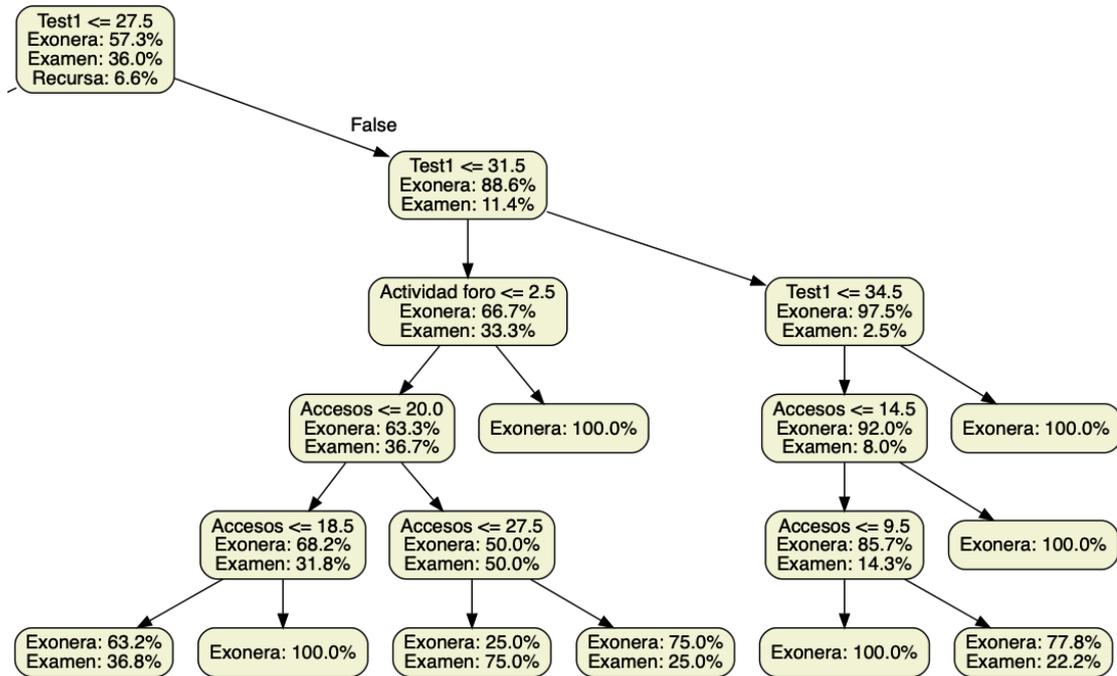


Figura 7.11: Modelo 4 con logs sección derecha

En la Figura 7.11 se observa la sección derecha del árbol obtenido del modelo 4 con datos de accesos, para todos estos casos se considera que el estudiante obtuvo más de 28 puntos en el parcial 1, las conclusiones mas relevantes obtenidas son:

- Obtener entre 32 y 35 puntos en el parcial 1 y tener un promedio de mas de 15 accesos semanales favorece las posibilidades de exoneración.
- Obtener menos de 31 puntos en el parcial 1 y tener un promedio de mayor a 3 en actividad de foro favorece las posibilidades de exoneración.
- Tener un promedio de actividad en foro menor a 3 y un promedio de accesos mayor a 20 si se tiene un promedio de accesos mayor a 27.5 se exonera con un 75 % sin embargo si es menor se va a examen con un 75 %.

De este modelo se puede extraer que los accesos semanales y la actividad en foro son muy positivos para la exoneración del curso.

7.4.8. Resultados

Dada la poca información con la que se contó a la hora de entrenar los distintos modelos, no se pueden sacar conclusiones generales confiables, de todas maneras los modelos en todos los casos dan información útil para que los docentes la utilicen como un insumo más para analizar la situación de cada estudiante en particular, del grupo, del curso y poder comprender las necesidades y eventualmente tomar decisiones para mejorar el desempeño de los estudiantes.

Capítulo 8

Conclusiones y trabajo a futuro

En este capítulo se mencionan las conclusiones finales en base al sistema construido y los resultados obtenidos, así como las posibles extensiones y trabajos a futuro.

8.1. Conclusiones generales

En base a los objetivos planteados al comienzo del proyecto, puede concluirse que se lograron cumplir con todos ellos y obtener los resultados esperados. El prototipo desarrollado cumple satisfactoriamente con todos los requisitos definidos tanto funcionales como no funcionales; gestionar, visualizar y analizar datos de los estudiantes del curso de Programación 2 de la Facultad de Ingeniería. Así como también lograr la integración con la plataforma Moodle como fuente centralizada de datos. Todo esto teniendo en cuenta que se enfocaron significantes esfuerzos en elaborar una interfaz gráfica que permita al usuario poder visualizar y entender la información presentada de una manera simple y clara.

Se destaca el hecho de que la solución diseñada aporta a la validación de la viabilidad de desarrollar sistemas de Learning Analytics con las herramientas y recursos de la Facultad de Ingeniería. En particular es un área en pleno desarrollo en nuestro país actualmente, por lo cual fue de gran interés y motivación poder aportar al desarrollo del mismo para así lograr una mejora en la educación y sumar recursos a la facultad que faciliten hacerlo. Como ya fue mencionado, se realizó una demostración de la plataforma con integrantes de la Unidad de Enseñanza de la Facultad de Ingeniería con el propósito de validar la viabilidad de utilizar la herramienta en futuros cursos de la Facultad.

Por otro lado, cabe mencionar que para validar de manera adecuada la viabilidad del uso del sistema en un contexto real y asegurarse que el valor que aporta lo justifica, es necesario probarlo con los docentes durante el transcurso de los cursos.

Uno de los grandes desafíos que presentó el proyecto, fue poder obtener una cantidad suficiente de información para poder aplicar los algoritmos de aprendizaje automático y que a su vez que esta información sea de calidad. Incluso contando con las respuestas de la encuesta realizada y el acceso a los registros de actividad en Moodle, la cantidad de datos obtenidos no representó un volumen significativo comparado al utilizado normalmente en problemas similares en el área del aprendizaje automático. De todos modos se logró crear modelos predictivos relevantes y de gran utilidad, permitiendo obtener conclusiones muy interesantes que a su vez pueden ser utilizadas en futuras predicciones.

Otro de los desafíos destacados del proyecto fue la investigación de las posibilidades de integración con la plataforma de Moodle utilizada por la Facultad, fundamental para el prototipo desarrollado. Esta investigación permitió entender el alcance y funcionamiento de la plataforma, lo cual llevó a tomar decisiones importantes en la arquitectura de la plataforma.

Si bien la plataforma construida es un prototipo con funcionalidades limitadas, posee características que hacen que pueda ser utilizado en un contexto real y se mantenga perdurable con el tiempo. La arquitectura diseñada permite extensibilidad a otros cursos más allá de Programación 2, permitiendo realizar las configuraciones necesarias para mantener interoperabilidad con Moodle aún con cursos con diferencias en los nombres de las actividades realizadas por los estudiantes. Para esto se debería adaptar o reemplazar el módulo de aprendizaje automático a las particularidades de un nuevo curso, pero la plataforma en sí es genérica en el sentido que no depende fuertemente de ningún curso en particular. Y por último, cabe destacar que la propia naturaleza de los algoritmos de aprendizaje automático utilizados, resultan en una mejora continua de las predicciones. A medida que pasan los cursos, se tienen más datos que hacen el entrenamiento de los modelos mejores y más relevantes.

8.2. Trabajo a futuro

A continuación se presentan diferentes trabajos futuros, extensiones y mejoras que podrían potenciar el uso del prototipo construido.

- **Generación automática de alertas para los estudiantes:** Se trata de una funcionalidad existente en sistemas de monitoreo estudiantil en varias universidades. La misma consiste en expandir el acceso a la plataforma para que los estudiantes puedan visualizar sus datos, y adicionalmente que cuenten con la opción de ser alertados en forma de emails o mensajes directos a través de Moodle en caso de que su predicción indique algún riesgo o situación a ser atendida.

- **Mayor análisis de registros de accesos en Moodle:** Como se mencionó en la sección 6.5, una fuente de datos importante del sistema es la extracción de información sobre los accesos a EVA por parte de los estudiantes de la Facultad, la cual es agrupada en cuatro categorías. Esta funcionalidad tiene un potencial de mayor análisis, buscando definir una mejor segmentación en los tipos de actividades que se pueden diferenciar, mejorando los modelos predictivos construidos.
- **Integración con nuevas fuentes de datos:** La encuesta demográfica realizada a los estudiantes puede considerarse como una prueba de concepto que permitió validar la viabilidad del prototipo construido. Sin embargo, existen diversas posibilidades de integración con otras fuentes de datos externas, tomando como ejemplos la posibilidad de una integración con sistemas de datos de la Unidad de Enseñanza de Facultad de Ingeniería, datos de Beldelías, o los resultados obtenidos en la Herramienta Diagnóstica al Inicio por los estudiantes al ingresar a la facultad.

Bibliografía

- [1] Web: Portal oficial de Moodle
Documentación oficial sobre la herramienta
https://docs.moodle.org/36/en/About_Moodle
Último acceso: 2019-05-27
- [2] Web: Portal oficial de Moodle
Documentación oficial sobre web services de Moodle
https://docs.moodle.org/dev/Web_services
Último acceso: 2019-05-27
- [3] Web: Organización de Estados Iberoamericanos
Entrada de blog sobre Learning Analytics
<https://www.oei.es/historico/divulgacioncientifica/?Learning-analytics-instrumento>
Último acceso: 2019-05-27
- [4] Informe Técnico de la Unidad de Enseñanza de la Facultad de Ingeniería:
Avance estudiantil – Año 2017
https://www.fing.edu.uy/sites/default/files/2017/31265/Informe%20avance%20en%20las%20carreras%20a%C3%B1o_2017.pdf
Fecha de publicación: Marzo 2018
- [5] Web: Entrada en el portal de EL Observador
<https://www.elobservador.com.uy/nota/el-72-de-estudiantes-de-udelar-se-atrasa-o-deserta-de-la-carrera--201210722390>
Fecha de publicación: Octubre 07, 2012
Último acceso: 2019-05-27
- [6] Web: Blog Aula 1
Entrada sobre entornos virtuales de aprendizaje
<https://www.aula1.com/entorno-virtual-aprendizaje-eva>
Último acceso: 2019-05-27

- [7] Web: Massive Online Open Courses
Portal de MOOC en español
<http://mooc.es/que-es-un-mooc>
Último acceso: 2019-05-27
- [8] Web: Nota publicada en La Diaria Educación
<https://educacion.ladiaria.com.uy/articulo/2018/8/el-entorno-virtual-de-aprendizaje-de-la-udelar-cumplio-diez-anos-y-se-desar>
Fecha de publicación: 22 de agosto 2018
Último acceso: 2019-05-27
- [9] Web: Aplicación de Learning Analytics, ¿Qué decisiones puedo tomar a partir de la actividad de los alumnos?
Entrada en blog sobre Learning Analytics
Autores: Miguel Ángel Conde González <http://scopeo.usal.es/aplicacion-de-learning-analytics/>
Último acceso: 2019-05-27
- [10] Learning Analytics: A Systematic Literature Review
Artículo publicado por Interdisciplinary Journal of Virtual Learning in Medical Sciences
Autores: Seyyed Kazem Banihashem, Khadijeh Aliabadi, et al.
DOI: 10.5812/ijvlms.63024
<http://ijvlms.com/en/articles/63024.html>
Fecha de publicación: Junio 12, 2018
- [11] Web: Conferencia Interacional de Learning Analytics
Entrada en blog sobre Learning Analytics
<https://tekri.athabasca.ca/analytics>
Último acceso: 2019-05-27
- [12] The Next Step for Learning Analytics
Autores: Jinan Fiaidhi
Revista: IT Professional volumen 16, edición 5
DOI: 10.1109/MITP.2014.78
Fecha de publicación: Septiembre 24, 2014
- [13] Learning Analytics Methods, Benefits, and Challenges in Higher Education: A Systematic Literature Review
Revista: Journal of Asynchronous Learning Networks volumen 20, edición 2
Autores: John T. Avella, Mansureh Kebritchi, Sandra G. Nunn, Therese Kanai

ISSN: ISSN-1939-5256

Fecha de publicación: Junio 29, 2016

- [14] Course signals at Purdue: Using learning analytics to increase student success
Libro: Proceedings of the 2nd International Conference on Learning Analytics and Knowledge

Editorial: Association for Computing Machinery

Autores: Kimberly E. Arnold, Matthew D. Pistilli

DOI: 10.1145/2330601.2330666

https://www.researchgate.net/publication/254462830_Course_signals_at_Purdue_Using_learning_analytics_to_increase_student_success
Fecha de publicación: Abril 29, 2012

- [15] Tesis: Learning Analytics in Massive Open Online Courses

Autores: Mohammad Khalil

ArXiv: 1802.09344

<https://arxiv.org/abs/1802.09344>

Fecha de publicación: Febrero 12, 2018

- [16] Let's not forget: Learning analytics are about learning

Revista: TechTrends volumen 59, edición 1 DOI: 10.1007/s11528-014-0822-x

Fecha de publicación: Enero, 2015

- [17] Web: ¿Qué es Machine Learning?

<https://cleverdata.io/que-es-machine-learning-big-data>

Último acceso: 2019-05-27

- [18] Web: CleverData

Conceptos básicos de Machine Learning

<https://cleverdata.io/conceptos-basicos-machine-learning/>

Último acceso: 2019-05-27

- [19] Web: spartanhack

Machine Learning y Deep Learning. Todo lo que necesitas saber

<https://spartanhack.com/machine-learning-y-deep-learning-todo-lo-que-necesitas-saber>

Último acceso: 2019-05-27

- [20] Web: Portal oficial de SciKit Learn

Documentación oficial sobre algoritmos para implementar un Decision Tree

[https://scikit-learn.org/stable/modules/tree.html#](https://scikit-learn.org/stable/modules/tree.html#tree-algorithms-id3-c4-5-c5-0-and-cart)

[tree-algorithms-id3-c4-5-c5-0-and-cart](https://scikit-learn.org/stable/modules/tree.html#tree-algorithms-id3-c4-5-c5-0-and-cart)

Último acceso: 2019-05-27

- [21] Web: rpubs
Evaluación de modelos de clasificación <https://rpubs.com/chzelada/275494>
Último acceso: 2019-05-27
- [22] Web: exsilio
Accuracy, Precision, Recall & F1 Score: Interpretation of Performance Measures
<https://blog.exsilio.com/all/accuracy-precision-recall-f1-score-interpretat>
Último acceso: 2019-05-27
- [23] Blog: PowerData
Procesos ETL: Definición, Características, Beneficios y Retos
<https://blog.powerdata.es/el-valor-de-la-gestion-de-datos/bid/312584/procesos-etl-definici-n-caracter-sticas-beneficios-y-retos>
Último acceso: 2019-05-27
- [24] Web: Oracle
Documentación sobre procesos de extracción
https://docs.oracle.com/cd/B19306_01/server.102/b14223/extract.htm
Último acceso: 2019-05-27
- [25] Web: ScienceDirect
A proposed model for data warehouse ETL processes
<https://www.sciencedirect.com/science/article/pii/S131915781100019X>
Último acceso: 2019-05-27
- [26] Web: Git
Portal oficial de Git
<https://git-scm.com>
Último acceso: 2019-05-27
- [27] Web: GitHub
Portal oficial de GitHub
<https://github.com>
Último acceso: 2019-05-27
- [28] Web: Trello
Portal oficial de Trello
<https://trello.com>
Último acceso: 2019-05-27

- [29] Web: RapidMiner
Portal oficial de RapidMiner
<https://rapidminer.com>
Último acceso: 2019-05-27
- [30] Web: KNIME
Portal oficial de KNIME
<https://www.knime.com/about>
Último acceso: 2019-05-27
- [31] Web: Orange
Portal oficial de Orange
<https://orange.biolab.si>
Último acceso: 2019-05-27
- [32] Web: Portal oficial de node.js
<https://nodejs.org/es>
Último acceso: 2019-05-27
- [33] Web: Portal oficial de Python
<https://www.python.org/>
Último acceso: 2019-05-27
- [34] Web: Portal oficial de Scikit-Learn
<https://scikit-learn.org/stable/>
Último acceso: 2019-05-27
- [35] Kanban - Successful Evolutionary Change for your Technology Business
Editorial: Blue Hole Press
Autores: David J. Anderson
ISBN: 0984521402
Fecha de publicación: Abril, 2010
- [36] Web: Gitflow
Guía de Gitflow publicada por Atlassian
[https://www.atlassian.com/git/tutorials/comparing-workflows/
gitflow-workflow](https://www.atlassian.com/git/tutorials/comparing-workflows/gitflow-workflow)
Último acceso: 2019-05-27
- [37] Web: Atlassian
Portal de Atlassian en español
<https://es.atlassian.com>
Último acceso: 2019-05-27

- [38] Web: Portal oficial de TypeScript
url<https://www.typescriptlang.org/>
Último acceso: 2019-05-27
- [39] Web: Portal oficial de React.js
url<https://reactjs.org/>
Último acceso: 2019-05-27
- [40] Software Engineering (10th Edition), Capítulo 16
Editorial: Pearson
Autores: Ian Sommerville
ISBN: 0133943038
Fecha de publicación: Abril 3, 2015
- [41] Web: Portal oficial de Express.js
<https://expressjs.com>
Último acceso: 2019-05-27
- [42] Web: Portal oficial de Objection.js
<https://vincit.github.io/objection.js>
Último acceso: 2019-05-27
- [43] Web: Portal oficial de Cheerio.js
<https://cheerio.js.org>
Último acceso: 2019-05-27
- [44] Web: GitHub de csvtojson
<https://github.com/Keyang/node-csvtojson>
Último acceso: 2019-05-27
- [45] Web: Portal oficial de Chart.js
https://docs.moodle.org/dev/Web_service_API_functions
Último acceso: 2019-05-27
- [46] Web: GitHub de randomcolor
<https://github.com/davidmerfield/randomColor>
Último acceso: 2019-05-27
- [47] Web: Portal oficial de Moodle
Documentación oficial sobre instalación de Moodle en MacOS
<https://download.moodle.org/macosex>
Último acceso: 2019-05-27

- [48] Web: Portal oficial de MAMP
<https://www.mamp.info/en>
Último acceso: 2019-05-27
- [49] Web: Portal oficial de Moodle
Documentación oficial sobre web services de Moodle
https://docs.moodle.org/dev/Web_service_API_functions
Último acceso: 2019-05-27
- [50] Web: Portal oficial de Postman
<https://www.getpostman.com/>
Último acceso: 2019-05-27
- [51] Web: Portal oficial de Django
<https://www.djangoproject.com/>
Último acceso: 2019-05-27
- [52] Web: Portal oficial de Graphviz
<https://www.graphviz.org/>
Último acceso: 2019-05-27
- [53] Web: Documentación oficial de Python
<https://docs.python.org/3/library/pickle.html>
Último acceso: 2019-05-27
- [54] Web: Portal oficial de Scikit-Learn
Documentación oficial sobre el clasificador Decision Tree
<https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html#sklearn.tree.DecisionTreeClassifier>
Último acceso: 2019-05-27
- [55] Web: Portal oficial de pm2
<http://pm2.keymetrics.io>
Último acceso: 2019-05-27

Anexo A

Manual de usuario

Fing Analytics es un sistema que brinda acceso a la información del curso de Programación 2 de forma unificada y extraída de Moodle. Como principal funcionalidad, permite predecir los resultados de los estudiantes utilizando algoritmos de aprendizaje automático y visualizar dicha predicción de manera simple y clara.

Este manual de usuario tiene como objetivo facilitar la tarea de conocimiento, uso y aprendizaje del sistema desarrollado. Contiene información acerca de todas las operaciones básicas que el sistema ofrece, así como capturas de pantallas útiles para el seguimiento de la explicación.

A.1. Login

El sistema cuenta con un mecanismo de autenticación integrado con Moodle, permitiendo así prescindir de un sistema de registro e ingresar con las mismas credenciales utilizadas en EVA.



The image shows a login form for Fing Analytics. At the top center is a circular logo with a blue background and white text that reads "UNIVERSIDAD DE VALLE" and "INGENIERIA". Below the logo, the text "Fing Analytics" is displayed. The form consists of two input fields: the first is labeled "Username" and contains the placeholder text "Ingrese su usuario de Moodle"; the second is labeled "Contraseña" (Password). Below these fields is a prominent orange button with the text "Log in" in white.

Figura A.1: Formulario de login

A.2. Roles

Existen tres roles para los usuarios: nuevo, profesor y administrador.

Para los usuarios que acceden por primera vez, es necesario que un administrador active la cuenta con un rol que permita acceder al resto de las funcionalidades. De lo contrario se mostrará un mensaje de alerta como se muestra en la Figura A.2.

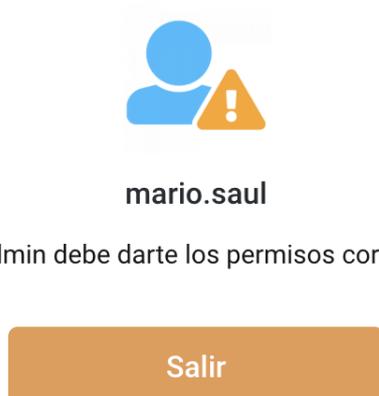


Figura A.2: Mensaje de alerta para usuarios con rol nuevo

Los usuarios con rol de profesor tienen acceso a todas las funcionalidades de visualización, mientras que los administradores también pueden acceder a funcionalidades de gestión.

A.3. Pantalla de Resumen

Una vez logueado, el usuario es redirigido a la pantalla de Resumen. A continuación se procede a explicar cada uno de los elementos de la misma como se muestra en la Figura A.3

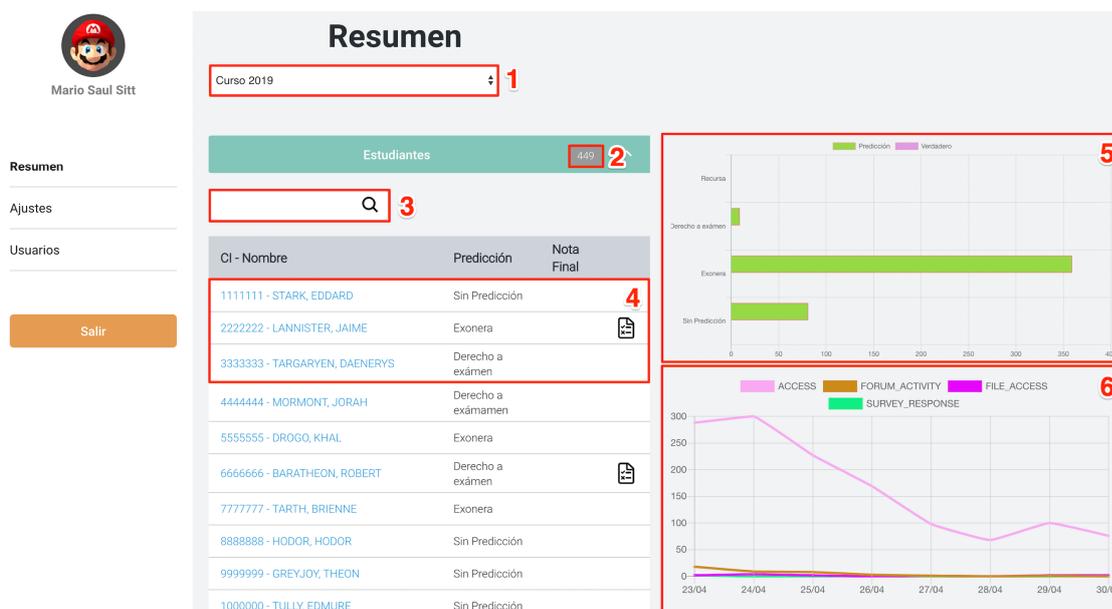


Figura A.3: Pantalla de Resumen

A.3.1. Selector de curso

Selector que sirve para navegar entre cursos disponibles, tanto el actual como históricos. En el caso de cursos pasados, la página entera se actualiza para mostrar los datos finales del curso en lugar de las predicciones ya que no son relevantes para dicho caso.

A.3.2. Total de estudiantes

Etiqueta que refleja la cantidad total de estudiantes en el curso seleccionado.

A.3.3. Campo de búsqueda

Campo de texto que permite filtrar estudiantes según cédula, nombre, apellido o predicción.

A.3.4. Tabla de estudiantes

Tabla que contiene los detalles de los estudiantes del curso seleccionado. La misma contiene una fila por cada estudiante y cuatro columnas con sus atributos:

Columna	Descripción
CI - Nombre	Cédula, nombres y apellidos. Clickeando sobre este campo se dirige a la pantalla de detalles del estudiante
Predicción	Para el caso que el curso actual en progreso esté seleccionado, muestra la predicción realizada por el algoritmo de aprendizaje automático o el valor “Sin Predicción” para los estudiantes que ya han desistido o perdido el curso.
Nota final	Para el caso que el curso actual en progreso esté seleccionado, no se muestra nada. Para el caso que un curso anterior al actual esté seleccionado, muestra la nota final obtenida.
Encuesta	Muestra el icono si el estudiante respondió la encuesta.

Tabla A.1: Explicación de las columnas

A.3.5. Gráfica de resultados

Para el caso que el curso actual en progreso esté seleccionado, muestra la cantidad de estudiantes que cayeron en cada categoría posible de predicciones. Para el caso que un curso anterior al actual esté seleccionado, muestra la cantidad de estudiantes que obtuvieron cada posible resultado final.

A.3.6. Gráfica de logs

Muestra la cantidad de logs de cada tipo existente de todos los estudiantes del curso actual de forma cronológica para la semana actual.

A.4. Pantalla de Detalles de Estudiante

Al hacer click sobre un estudiante en la tabla de estudiantes, se dirige a la pantalla de detalles del mismo. En esta se puede encontrar la información detallada sobre la actividad del estudiante en el curso así como información relacionada a las predicciones realizadas. A continuación se procede a explicar cada uno de los elementos como se muestran en la Figura A.4.

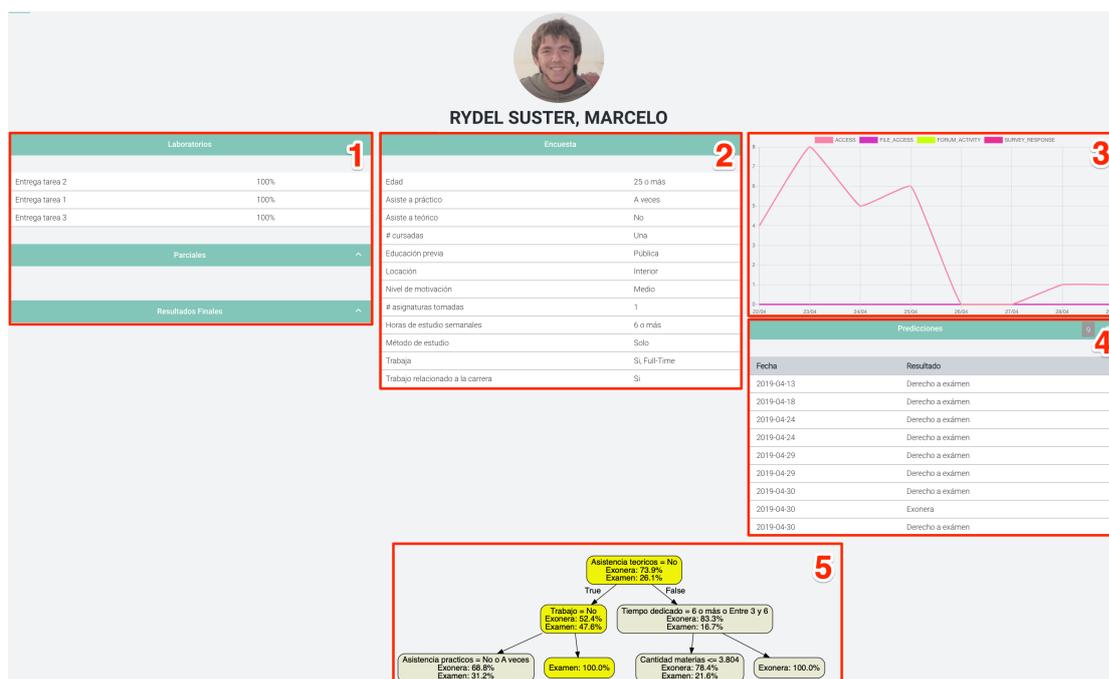


Figura A.4: Pantalla de detalles de estudiante

A.4.1. Detalles del curso

Sección que detalla los resultados obtenidos hasta el momento por el estudiante en el curso.

A.4.2. Detalles de la encuesta

Si el estudiante completó la encuesta demográfica, en esta sección se muestran sus respuestas.

A.4.3. Gráfica de logs

Muestra la cantidad de logs de cada tipo existente de forma cronológica para las últimas dos semanas.

A.4.4. Historial de predicciones

Historial de predicciones hechas en orden cronológico con el objetivo de entender la evolución del estudiante a lo largo del curso.

A.4.5. Árbol de decisión

Imagen del árbol de decisión utilizado en la última predicción. El camino recorrido se encuentra marcado con el color correspondiente al resultado (verde si exonera, amarillo si derecho a examen y rojo si recursa).

A.5. Ajustes

En la pantalla de ajustes accesible a través del menú de la izquierda, se pueden realizar diferentes configuraciones relacionado a los cursos, actividades del EVA y estudiantes. Esta sección es visible solo para usuarios con rol de administrador. A continuación se procede a explicar cada uno de los elementos de la misma como se muestra en la Figura A.5

Ajustes de cursos

Cursos

Curso 2019 1

Nuevo Curso 2

Sincronizar 4 Estudiantes 5 Resultados 6 Encuestas 7 Predecir Resultados

Cargar CSV 8 Resultados 8

Curso 2019 2

Año 2019 3

Eva Id 132 3 Editar

Nombre de encuesta Encuesta Estudiantil sobre el Curso y Actividades Académicas 3 Editar

Actividades 2019

Laboratorios 5

Parciales 6

Actividad	Actividad EVA	Actividad Re-entrega	Editar
Entrega tarea 1	Entrega tarea 1	Instancia ante entregas insuficientes de la tarea 1	Editar 9
Entrega tarea 2	Entrega tarea 2	Instancia ante entregas insuficientes de la tarea 2	Editar
Entrega tarea 3	Entrega tarea 3	Instancia ante entregas insuficientes de la tarea 3	Editar
Entrega tarea 4	Entrega tarea 4	Instancia ante entregas insuficientes de la tarea 4	Editar
Entrega tarea 5	Entrega tarea 5	Instancia ante entregas insuficientes de la tarea 5	Editar

Primer Parcial Editar

Segundo Parcial Editar

Resultados finales 1

Actividad	Actividad EVA	Editar
Nota Final	Nota Final	Editar

Estudiantes 2019

Estudiantes 449

10 Q

CI - Nombre	Predicción	Laboratorios	Parciales	Quitar del curso 11
1111111 - STARK, EDDARD	Exonera	2	0	Quitar del curso
2222222 - LANNISTER, JAIME	Exonera	2	0	Quitar del curso
3333333 - TARGARYEN, DAENERYS	Exonera	3	0	Quitar del curso

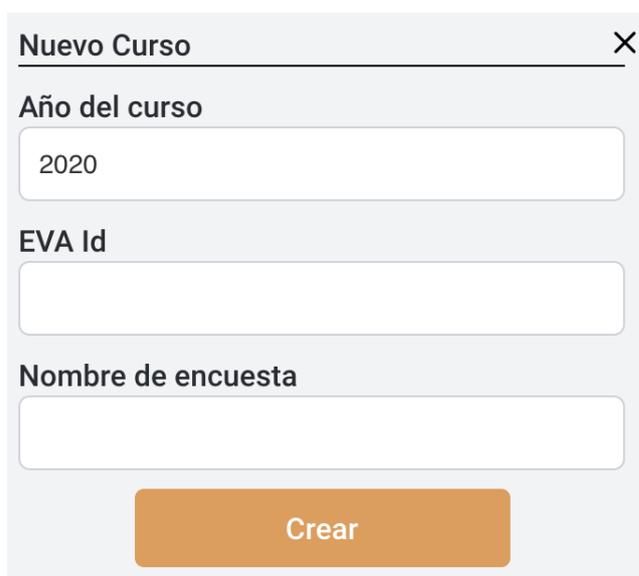
Figura A.5: Pantalla de Ajustes

A.5.1. Selector de curso

Selector que contiene todos los cursos disponibles en el sistema. Al cambiar la selección se actualizan todos los demás campos de la pantalla reflejando los datos del nuevo curso seleccionado.

A.5.2. Nuevo curso

Botón que al ser clickeado presenta un formulario como se muestra en la Figura A.6. Los campos *Año del curso* y *EVA id* son requeridos mientras que el campo *Nombre de encuesta* es opcional dejando abierta la posibilidad a cursos sin encuesta. Todos estos campos son utilizados al momento de sincronizar la distinta información del curso con el EVA. Para conocer el *EVA id* del curso que se quiere agregar a la plataforma, el mismo puede ser obtenido mirando el último parámetro en la URL del curso en EVA. Por ejemplo, para el curso de Programación 2 la URL tiene la forma `https://eva.fing.edu.uy/course/view.php?id=132`.



Nuevo Curso X

Año del curso

2020

EVA Id

Nombre de encuesta

Crear

Figura A.6: Crear nuevo Curso

A.5.3. Editar curso

Botón que al ser clickeado presenta un formulario como se muestra en la Figura A.7. Permite la edición del *EVA id* y el *Nombre de encuesta* del curso seleccionado.

Editar Curso 2019

EVA Id

132

Nombre de encuesta

Encuesta Estudiantil sobre el Curso y Actividades Académicas

Edit

Figura A.7: Editar Curso

A.5.4. Sincronizar estudiantes

Botón que al ser clickeado realiza un pedido a Moodle para importar al sistema todos los estudiantes matriculados en el curso seleccionado. Luego de obtener los estudiantes matriculados se ingresan al sistema en caso de no existir, de lo contrario se sobre escribe la información con lo último obtenido desde Moodle.

A.5.5. Sincronizar resultados

Botón que al ser clickeado realiza un pedido a Moodle para importar al sistema todos los resultados de los estudiantes del curso seleccionado. Luego de obtener los resultados se actualizan en el sistema para todos los estudiantes. Esta sincronización suele demorar unos pocos minutos en completarse.

A.5.6. Sincronizar encuestas

Botón que al ser clickeado, en caso que el curso seleccionado tenga un *Nombre de encuesta*, realiza un pedido a Moodle para importar al sistema todas las respuestas de encuestas de los estudiantes del curso seleccionado. Luego de obtener los resultados se actualizan en el sistema para todos los estudiantes. Para el correcto funcionamiento de la sincronización, el valor de *Nombre de encuesta* debe coincidir con el nombre de la encuesta en EVA.

A.5.7. Predecir resultados

Botón que al ser clickeado realiza un pedido al módulo de aprendizaje automático para entrenar los distintos modelos de aprendizaje automático y cargar en la base de datos todas las predicciones de los estudiantes del curso seleccionado.

A.5.8. Cargar resultados desde CSV

Botón que permite al usuario ingresar o actualizar notas de estudiantes de forma manual, mediante la carga de un archivo con formato CSV. Esta funcionalidad se ofrece para los casos donde, por algún motivo, Moodle no cuente con las notas de los estudiantes y no se pueda ejecutar la sincronización automática. Los valores que se dejen vacíos para notas ya existentes, no se actualizarán en la base de datos.

A continuación se detallan las columnas que se deben agregar en el archivo, y se muestra un ejemplo:

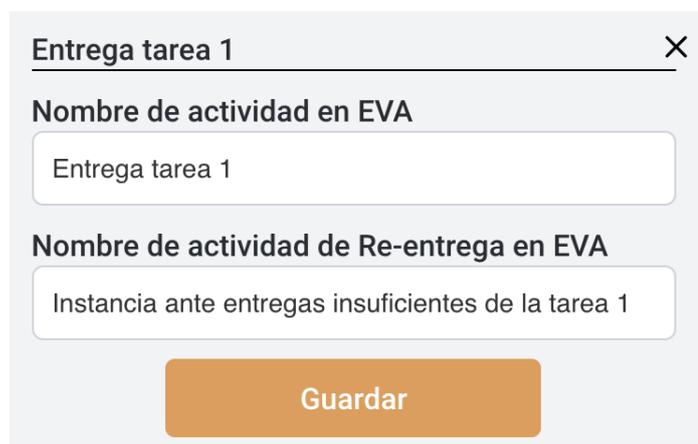
- CI (Sin dígito identificador)
- Test1 (Nota del primer parcial)
- Test2 (Nota del segundo parcial)
- FinalResult (Suma de los dos parciales)
- Assignment1 (Resultado del laboratorio 1)
- Assignment2 (Resultado del laboratorio 2)
- Assignment3 (Resultado del laboratorio 3)
- Assignment4 (Resultado del laboratorio 4)
- Assignment5 (Resultado del laboratorio 5)

CI	Test1	Test2	FinalResult	Assignment1	Assignment2	Assignment3	Assignment4	Assignment5
1234567	10	20	30	111101	111111	1001111		

Figura A.8: Ejemplo de archivo CSV para carga manual de notas

A.5.9. Editar actividad

Botón que al ser clickeado presenta un formulario como se muestra en la Figura A.9. El mismo permite editar la información de las distintas actividades del curso para la correcta sincronización con Moodle. Para el correcto funcionamiento de las herramientas de sincronización los nombres de las distintas actividades deben coincidir con los nombres de las mismas en Moodle. Aplica de igual manera para las actividades de los laboratorios, parciales y resultados finales.



Entrega tarea 1 ×

Nombre de actividad en EVA

Entrega tarea 1

Nombre de actividad de Re-entrega en EVA

Instancia ante entregas insuficientes de la tarea 1

Guardar

Figura A.9: Editar Actividad

A.5.10. Campo de búsqueda

Campo de texto que permite filtrar estudiantes según cédula, nombre, apellido o predicción.

A.5.11. Quitar estudiante del curso

Botón que al ser clickeado remueve al estudiante del sistema. Resulta de particular utilidad para poder quitar estudiantes que por alguna razón ya han perdido el curso.

A.6. Carga de Logs

Para realizar la actualización y carga de los logs, existe una tarea programada que es ejecutada cada 15 minutos de forma automática. Para el correcto funcionamiento de esta funcionalidad, es necesario brindar las credenciales de autenticación de manera manual ingresándolos en un archivo de configuración ubicado en el directorio raíz donde se encuentra instalado el proyecto. El nombre del archivo a editar se denomina *.env* y se encuentra oculto. Los valores que hay que editar son *EVA_STATIC_USERNAME* con el usuario y *EVA_STATIC_PASSWORD* con la contraseña. Alternativamente se pueden cargar estas dos variables como variables de entorno del sistema en la computadora donde este instalada la plataforma.

A.7. Usuarios

En la pantalla de usuarios accesible a través del menú de la izquierda, se listan todos los usuarios existentes en el sistema con sus respectivos roles. Esta sección es visible solo para usuarios con rol de administrador. A continuación se procede a explicar cada uno de los elementos de la misma como se muestra en la Figura A.10



Figura A.10: Pantalla de Usuarios

A.7.1. Editar rol de usuario

Botón que al ser clickeado presenta un formulario como se muestra en la Figura A.11. Los usuarios con acceso a este formulario pueden utilizarlo para cambiar los roles de otros usuarios y así asignar o quitar permisos.

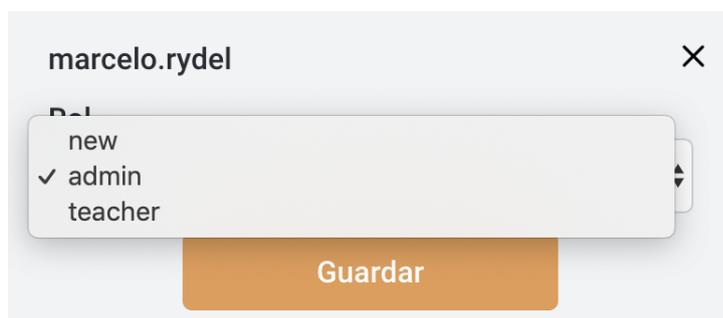


Figura A.11: Editar rol de Usuario

Anexo B

Manual de instalación

Fing Analytics consiste en una serie de herramientas que permiten la recolección y análisis de datos sobre alumnos de Programación 2 y sus contextos, con el propósito de comprender y optimizar el aprendizaje y el entorno en el cual ocurre. En este manual de instalación se detallan los pasos a seguir para instalar el sistema en su completitud. Este manual está intencionado para el sistema operativo Linux aunque no debería de haber mayores variaciones para otros distintos. El manual se divide en tres secciones, una para la instalación de la base de datos y una más por cada módulo del sistema.

B.1. Base de datos

Existe una única base de datos compartida por ambos módulos. Es una base de datos relacional MySQL sencilla de instalar. El primer paso es ejecutar los siguientes comandos desde la terminal de Linux:

```
1 $ sudo apt-get update
2 $ sudo apt-get install mysql-server
```

Luego es necesario crear una base de datos y popularla con el esquema y datos iniciales. Para esto se provee un archivo el cual contiene toda la información necesaria para ser importada a la base de datos. Esto se logra ejecutando los siguientes comandos desde la terminal:

```
1 $ mysql -u USUARIO -p
2 $ create database fing_analytics;
3 $ use fing_analytics;
4 $ SET GLOBAL sql_mode=(SELECT REPLACE (@@sql_mode, 'ONLY_FULL_GROUP_BY', ''));
5 $ exit;
6 $ mysql -u USUARIO NOMBRE_BASE RUTA_DEL_ARCHIVO_DUMP
```

Una vez ejecutados los pasos previamente descritos, la base de datos a para ser utilizada por la plataforma administrativa y el módulo de aprendizaje automático.

B.2. Plataforma administrativa

La plataforma administrativa consiste en una aplicación javascript, por lo que se necesita *node.js* para ejecutarla. Este programa se puede descargar desde la página oficial [32]. Para el desarrollo se utilizó la versión 11.10.0. A su vez, la instalación de *node.js* trae consigo la herramienta *npm* (Node Package Manager). Ésta es utilizada para instalar los paquetes de los que depende la aplicación.

B.2.1. Configuración

En primer lugar es necesario configurar las variables globales de la aplicación. En el archivo *.env* se pueden encontrar todas las variables disponibles.

- **DB_HOST.** Dirección donde se encuentra el servidor de base de datos.
- **DB_USER.** Usuario con el cual conectarse a la base de datos.
- **DB_PASSWORD.** Contraseña del usuario.
- **DB_NAME.** Nombre de la base de datos.
- **DB_PORT.** Puerto por el cual se accede al servidor de base de datos. MySQL por defecto se encuentra en el puerto 3306.
- **DB_CHARSET.** Conjunto de caracteres utilizado por el servidor de base de datos. Se recomienda utilizar *utf8mb4*, ya que soporta caracteres especiales como letras con tilde.
- **PORT.** Puerto por el cual se accede a los servicios del módulo.
- **UPLOADS_FOLDER.** Carpeta donde se guarda temporalmente los archivos subidos a la plataforma.
- **JWT_SECRET_KEY.** Cadena de caracteres utilizada para firmar los tokens de autenticación.
- **ANALYTICS_URL.** URL donde se encuentran los servicios del módulo de aprendizaje automático.

B.2.2. Instalación

Para levantar la plataforma se utiliza *pm2* [55]. Es un paquete de *node.js* utilizado para levantar automáticamente aplicaciones. Para instalarlo se debe correr el siguiente comando desde una terminal:

```
1 $ npm install -g pm2
```

Luego, sobre el directorio del backend se corren los siguientes comandos desde una terminal:

```
1 $ npm install
2 $ pm2 start app.js
```

El primer comando instalará las dependencias del proyecto, mientras que el segundo ejecutará la aplicación, a la cual se podrá acceder a través del navegador en la URL `http://localhost:PUERTO/app`, donde `PUERTO` es el `PORT` configurado.

B.3. Módulo de aprendizaje automático

El módulo de aprendizaje automático está implementado en Python sobre el framework web Django. Para la instalación del sistema es necesario instalar previamente estas herramientas necesarias.

B.3.1. Configuración

En primer lugar es necesario configurar las variables globales de la aplicación. En el archivo `.env` se pueden encontrar todas las variables disponibles.

- **DATABASE_NAME.** Nombre de la base de datos.
- **DATABASE_USER.** Usuario con el cual conectarse a la base de datos.
- **DATABASE_PASSWORD.** Contraseña con el cual conectarse a la base de datos.

B.3.2. Instalación

El primer paso es instalar Python con el siguiente comando desde la terminal de Linux:

```
1 $ sudo apt-get install python3
```

Para verificar la correcta instalación, correr el siguiente comando y ver que la versión de Python se imprima en pantalla:

```
1 $ python3 -V
```

Una vez instalado Python, se procede a instalar pip que es el instalador de paquetes de Python el cual necesitaremos para instalar posteriormente Django. Para esto, correr el siguiente comando:

```
1 $ sudo apt-get install -y python3-pip
```

De manera similar al paso anterior, podemos verificar la correcta instalación con el siguiente comando para verificar la versión:

```
1 $ pip3 -V
```

Con pip instalado, se puede proceder a instalar Django con el siguiente comando:

```
1 $ pip install django
```

Para verificar la correcta instalación, correr el siguiente comando:

```
1 $ django-admin --version
```

Con todas las herramientas necesarias instaladas, el último paso es situarse en el directorio donde se encuentra el proyecto y correr el siguiente comando:

```
1 $ cd RUTA_DEL_PROYECTO
2 $ python manage.py runserver NÚMERO_DE_PUERTO
```

Luego de este último paso, el servidor se encuentra corriendo en localhost y número de puerto especificado (si no se especifica se toma el 8000 por defecto)

Anexo C

Documentación de verificación del sistema

En este anexo se entra en detalle en las actividades de verificación del sistema. Se muestra cuales fueron las pruebas y los resultados obtenidos.

C.1. Login

En esta sección se engloban las pruebas realizadas sobre el formulario de ingreso al sistema. Este caso de uso es de especial interés dado que se integra directamente con Moodle.

En esta sección se engloban las pruebas realizadas sobre el formulario de ingreso al sistema. Este caso de uso es de especial interés dado que se integra directamente con Moodle. En esta sección se engloban las pruebas realizadas sobre el formulario de ingreso al sistema. Este caso de uso es de especial interés dado que se integra directamente con Moodle. En esta sección se engloban las pruebas realizadas sobre el formulario de ingreso al sistema. Este caso de uso es de especial interés dado que se integra directamente con Moodle.

Prueba 1

Pasos para reproducir

1. Se ingresan el usuario y contraseña de Moodle correctos.
2. Los datos ingresados pertenecen a un usuario no existente en el sistema.

Resultado esperado

- Se ingresa al sistema.
 - Se crea un registro de usuario, con el nombre de usuario de Moodle utilizado y con el rol *new*.
 - Se muestra una pantalla indicando que no se cuentan con los permisos necesarios.
-

Prueba 2

Pasos para reproducir

1. Se ingresan el usuario y contraseña de Moodle correctos.
2. Los datos ingresados pertenecen a un usuario con rol *teacher*.

Resultado esperado

- Se ingresa al sistema.
 - Se muestra la pantalla principal de la plataforma con las opciones habilitadas para un usuario de rol *teacher*.
-

Prueba 3

Pasos para reproducir

1. Se ingresan el usuario y contraseña de Moodle correctos.
2. Los datos ingresados pertenecen a un usuario con rol *admin*.

Resultado esperado

- Se ingresa al sistema.
- Se muestra la pantalla principal de la plataforma con las opciones habilitadas para un usuario de rol *admin*.

Prueba 4

Pasos para reproducir

1. Se ingresan el usuario y contraseña de Moodle incorrectos.

Resultado esperado

- No se ingresa al sistema.
 - Se muestra un error indicando que el usuario o contraseña es incorrecto.
-

Prueba 5

Pasos para reproducir

Para este caso se asume que Moodle no funciona como se espera.

1. Se ingresan el usuario y contraseña de Moodle.

Resultado esperado

- No se ingresa al sistema.
 - Se muestra un error indicando que hubo un error inesperado.
-

C.2. Gestión de un curso

Las funcionalidades que engloba esta sección son todas aquellas que involucran el ingreso o edición de datos de un curso en particular. Todas estas funcionalidades se encuentran en la pantalla de ajustes de la plataforma administrativa, para un usuario del rol *admin*.

Nuevo curso - Prueba 1

Pasos para reproducir

1. Se navega hasta el formulario “Nuevo curso”.
2. Se ingresan año, id de EVA y el nombre de la encuesta.
 - No existe actualmente un curso para el año ingresado.

Resultado esperado

- Se crea un curso con el año, id de EVA y nombre de encuesta ingresados.
 - Para el nuevo curso creado, se crean cinco actividades de “Laboratorio”.
 - Para el nuevo curso creado, se crean dos actividades de “Parcial”.
 - Para el nuevo curso creado, se crea una actividad de “Resultado final”.
-

Nuevo curso - Prueba 2

Pasos para reproducir

1. Se navega hasta el formulario “Nuevo curso”.
2. Se ingresan año, id de EVA y el nombre de la encuesta.
 - Existe un curso para el año ingresado.

Resultado esperado

- No se crea un nuevo curso.
 - Se muestra un error indicando que ya existe un curso para el año ingresado.
-

Nuevo curso - Prueba 3

Pasos para reproducir

1. Se navega hasta el formulario “Nuevo curso”.
2. Se omite cualquiera de los campos del formulario.

Resultado esperado

- No se crea un nuevo curso.
 - Se muestra un error indicando que faltan datos requeridos para la creación del curso.
-

Editar curso - Prueba 1

Pasos para reproducir

1. Se navega hasta la opción “Editar” de un laboratorio.
2. Se ingresa cualquiera de los campos del formulario.

Resultado esperado

- La información del laboratorio editado cambia según los valores ingresados.
-

Editar curso - Prueba 2

Pasos para reproducir

1. Se navega hasta la opción “Editar” de un parcial.
2. Se ingresa cualquiera de los campos del formulario.

Resultado esperado

- La información del parcial editado cambia según los valores ingresados
-

Editar curso - Prueba 3

Pasos para reproducir

1. Se navega hasta la opción “Editar” de un resultado final.
2. Se ingresa cualquiera de los campos del formulario.

Resultado esperado

- La información del resultado final editado cambia según los valores ingresados
-

Quitar estudiante - Prueba 1

Pasos para reproducir

1. Hacer click en la opción “Quitar del curso” en un estudiante.

Resultado esperado

- El estudiante es marcado como removido del curso.
 - El estudiante no aparece en la vista resumida.
-

Agregar estudiante - Prueba 1

Pasos para reproducir

1. Hacer click en la opción “Agregar al curso” en un estudiante quitado.

Resultado esperado

- El estudiante ya no esta marcado como removido.
 - El estudiante aparece en la vista resumida.
-

Cargar resultados CSV - Prueba 1

Pasos para reproducir

1. Hacer click en la opción “Cargar CSV Resultados”.
2. Se carga ingresa un archivo CSV con el formato tal cual se indica en el manual.

Resultado esperado

- Los resultados que no existían en el sistema son creados según la información del archivo ingresado.
 - Los resultados que existían previamente son sobre-escritos según la información del archivo ingresado.
-

Cargar resultados CSV - Prueba 2

Pasos para reproducir

1. Hacer click en la opción “Cargar CSV Resultados”.
2. Se carga ingresa un archivo CSV con un formato incorrecto.

Resultado esperado

- Los resultados no son creados ni editados.
 - Se muestra una alerta indicando un error en el formato.
-

Cargar resultados CSV - Prueba 2

Pasos para reproducir

1. Hacer click en la opción “Cargar CSV Resultados”.
2. Se carga un archivo que no tiene extensión CSV.

Resultado esperado

- Los resultados no son creados ni editados.
 - Se muestra una alerta indicando que el archivo no es válido.
-

Cargar resultados de Moodle - Prueba 1

Pasos para reproducir

1. Hacer click en la opción “Cargar EVA Resultados”.

Resultado esperado

- Los resultados que no existían en el sistema son creados según la información que tiene Moodle para las actividades asociadas al curso.
 - Los resultados que existían previamente son sobre-escritos según la información de Moodle.
-

Cargar resultados de Moodle - Prueba 2

Pasos para reproducir

Para este caso se asume que Moodle no funciona como se espera.

1. Hacer click en la opción “Cargar EVA Resultados”.

Resultado esperado

- Los resultados no son creados ni editados.
 - Se muestra un error indicando que hubo un error inesperado.
-

Cargar estudiantes de Moodle - Prueba 1

Pasos para reproducir

1. Hacer click en la opción “Cargar EVA Estudiantes”.

Resultado esperado

- Los estudiantes que no existían previamente son creados para el curso indicado según la información proveniente de Moodle.
 - Los estudiantes que ya existían previamente son sobre-escritos con la información proveniente de Moodle.
 - Se entiende por información proveniente de Moodle a los siguientes datos:
 - Id de EVA
 - Cédula de identidad
 - Nombres y Apellidos
 - Email
 - URL de la imagen de perfil
-

Cargar estudiantes de Moodle - Prueba 2

Pasos para reproducir

Para este caso se asume que Moodle no funciona como se espera.

1. Hacer click en la opción “Cargar EVA Estudiantes”.

Resultado esperado

- Ningún estudiante es creado ni editado.
 - Se muestra un error indicando que hubo un error inesperado.
-

Cargar encuesta de Moodle - Prueba 1

Pasos para reproducir

1. Hacer click en la opción “Cargar EVA Encuestas”.

Resultado esperado

- Se crean registros con las respuestas a la encuesta de Moodle a los estudiantes que la contestaron.
 - Para aquellos estudiantes que ya tenían sus respuestas cargadas en el sistema, se sobre-escriben las mismas.
-

Cargar encuesta de Moodle - Prueba 2

Pasos para reproducir

Para este caso se asume que Moodle no funciona como se espera.

1. Hacer click en la opción “Cargar EVA Encuestas”.

Resultado esperado

- Ningún registro de respuesta a la encuesta de Moodle es creado ni editado.
 - Se muestra un error indicando que hubo un error inesperado.
-

Predecir resultados - Prueba 1

Pasos para reproducir

1. Hacer click en la opción “Predecir Resultados”

Resultado esperado

- Todos los estudiantes del curso pasaran por el modelo de predicción adecuado y se creará un registro de predicción asociado.
 - Al ingresar al detalle de un estudiante, para el curso en cuestión, se muestra la imagen del árbol de decisión del modelo, y el camino recorrido según los datos del estudiante.
-

Predecir resultados - Prueba 2

Pasos para reproducir

Para este caso se asume que el módulo de aprendizaje automático no funciona correctamente.

1. Hacer click en la opción “Predecir Resultados”

Resultado esperado

- Ningún registro de predicción es creado o editado.
 - Se muestra un error indicando que ocurrió un error inesperado.
-

C.3. Gestión de usuarios

Las funcionalidades que engloba esta sección son todas aquellas que involucran la edición de los usuarios del sistema. Todas estas funcionalidades se encuentran en la pantalla de usuarios de la plataforma administrativa, para un usuario del rol *admin*.

Editar usuario - Prueba 1

Pasos para reproducir

1. Se navega hasta la opción “Editar” de un usuario.
2. Se ingresa un rol de la lista.

Resultado esperado

- El usuario editado, al ingresar a la plataforma, la misma se comporta acorde al rol elegido. Ver C.1.
-

Anexo D

Servicios web de Moodle

En este anexo se muestran ejemplos de pedidos HTTP a los servicios web de Moodle y sus respectivas respuestas. Todos los servicios web requieren de un token de autenticación, el cual es ocultado por cuestiones de seguridad. Por las mismas razones, la información de los estudiantes o profesores devuelta por Moodle es enmascarada.

D.1. moodle_mobile_app

Este servicio es utilizado para autenticarse en Moodle y obtener un *token* de acceso, con el cual se puede acceder al resto de los servicios.

Pedido

```
1 GET https://eva.fing.edu.uy/login/token.php
2   ? username = <usuario_de_EVA>
3   & password = <contraseña_de_EVA>
4   & service = moodle_mobile_app
```

Figura D.1: Ejemplo de pedido para obtener token de autenticación

Respuesta

```
1 {
2   "token": <wstoken>,
3   "privatetoken": <privatetoken>
4 }
```

D.2. core_enrol_get_enrolled_users

Este servicio es utilizado para obtener datos de los estudiantes matriculados a un curso.

Pedido

```
1 GET https://eva.fing.edu.uy/webservice/rest/server.php
2   ? wstoken = <wstoken>
3   & wsfunction = core_enrol_get_enrolled_users
4   & moodlewsrestformat = json
5   & courseid = 132
```

Figura D.2: Ejemplo de pedido web a core_enrol_get_enrolled_users

Respuesta

```
1 [
2   {
3     "id": 105,
4     "firstname": "Juan",
5     "lastname": "Fernández ",
6     "fullname": "Juan Fernández",
7     "email": "juan.fernandez@fing.edu.uy",
8     "idnumber": "1234567",
9     "interests": "Fisica",
10    "firstaccess": 1279565998,
11    "lastaccess": 1556105757,
12    "description": "??<br /><br />",
13    "descriptionformat": 1,
14    "city": "Montevideo",
15    "country": "UY",
16    "profileimageurlsmall": "https://eva.fing.edu.uy/...",
17    "profileimageurl": "https://eva.fing.edu.uy/...",
18    "customfields": [...],
19    "groups": [...],
20    "roles": [
21      {
```

```

22         "roleid": 5,
23         "name": "Estudiante",
24         "shortname": "student",
25         "sortorder": 0
26     }
27 ],
28     "enrolledcourses": [...]
29 },
30 ...
31 ]

```

JSON D.2: Ejemplo de respuesta del servicio web `gradereport_user_get_grade_items`

D.3. `gradereport_user_get_grade_items`

Este servicio es utilizado para obtener los resultados de las actividades cargadas en el curso.

Pedido

```

1 GET https://eva.fing.edu.uy/webservice/rest/server.php
2   ? wstoken = <wstoken>
3   & wsfunction = gradereport_user_get_grade_items
4   & moodlewsrestformat = json
5   & courseid = 132

```

Figura D.3: Ejemplo de pedido web a `gradereport_user_get_grade_items`

Respuesta

```

1 [
2   {
3     "courseid": 132,
4     "userid": 105,
5     "userfullname": "Juan Fernández",
6     "maxdepth": 2,
7     "gradeitems": [
8       {
9         "id": 10315,
10        "itemname": "Instancia ante entregas insuficientes
11        de la tarea 1",
        "itemtype": "mod",

```

```

12         "itemmodule": "assign",
13         "iteminstance": 5105,
14         "itemnumber": 0,
15         "categoryid": 165,
16         "outcomeid": null,
17         "scaleid": null,
18         "locked": false,
19         "cmid": 85756,
20         "graderaw": null,
21         "gradedatesubmitted": null,
22         "gradedategraded": null,
23         "gradehiddenbydate": false,
24         "gradeneedsupdate": false,
25         "gradeishidden": false,
26         "gradeislocked": false,
27         "gradeisoverridden": false,
28         "gradeformatted": "-",
29         "grademin": 1,
30         "grademax": 4,
31         "rangeformatted": "No entregó&ndash;Aprobado",
32         "feedback": "",
33         "feedbackformat": 0
34     },
35     {
36         "id": 10314,
37         "itemname": "Entrega tarea 1",
38         "itemtype": "mod",
39         "itemmodule": "assign",
40         "iteminstance": 5104,
41         "itemnumber": 0,
42         "categoryid": 165,
43         "outcomeid": null,
44         "scaleid": null,
45         "locked": false,
46         "cmid": 85755,
47         "graderaw": null,
48         "gradedatesubmitted": null,
49         "gradedategraded": null,
50         "gradehiddenbydate": false,
51         "gradeneedsupdate": false,
52         "gradeishidden": false,
53         "gradeislocked": false,
54         "gradeisoverridden": false,
55         "gradeformatted": "-",
56         "grademin": 1,
57         "grademax": 4,
58         "rangeformatted": "No entregó&ndash;Aprobado",
59         "feedback": "",
60         "feedbackformat": 0

```

```

61     },
62     {
63         "id": 10419,
64         "itemname": "Primer parcial",
65         "itemtype": "manual",
66         "itemmodule": null,
67         "iteminstance": null,
68         "itemnumber": 0,
69         "categoryid": 165,
70         "outcomeid": null,
71         "scaleid": null,
72         "locked": false,
73         "graderaw": null,
74         "gradedatesubmitted": null,
75         "gradedategraded": null,
76         "gradehiddenbydate": false,
77         "gradeneedsupdate": false,
78         "gradeishidden": true,
79         "gradeislocked": false,
80         "gradeisoverridden": false,
81         "gradeformatted": "-",
82         "grademin": 0,
83         "grademax": 36,
84         "rangeformatted": "0&ndash;36",
85         "feedback": "",
86         "feedbackformat": 0
87     },
88     ...
89 ]
90 },
91 ...
92 ]

```

JSON D.3: Ejemplo de respuesta del servicio web `gradereport_user_get_grade_items`

D.4. `mod_feedback_get_feedbacks_by_courses`

Este servicio es utilizado para obtener las encuestas de un curso.

Pedido

```
1 GET https://eva.fing.edu.uy/webservice/rest/server.php
2   ? wstoken = <wstoken>
3   & wsfunction = mod_feedback_get_feedbacks_by_courses
4   & moodlewsrestformat = json
5   & courseids[] = 132
```

Figura D.4: Ejemplo de pedido web a gradereport_user_get_grade_items

Respuesta

```
1 {
2   "feedbacks": [
3     {
4       "id": 2911,
5       "course": 132,
6       "name": "Encuesta Estudiantil sobre el Curso y
7         Actividades Académicas",
8       "intro": "<p></p><p>Esta encuesta se enmarca en un
9         proyecto de grado que tiene por objetivo...",
10      "introformat": 1,
11      "anonymous": 2,
12      "email_notification": false,
13      "multiple_submit": false,
14      "autonumbering": false,
15      "site_after_submit": "",
16      "page_after_submit": "",
17      "page_after_submitformat": 1,
18      "publish_stats": false,
19      "timeopen": 1554473520,
20      "timeclose": 1557065520,
21      "timemodified": 1554478908,
22      "completionsubmit": false,
23      "coursemodule": 87302,
24      "introfiles": []
25    },
26    ...
27  ],
28  "warnings": []
29 }
```

JSON D.4: Ejemplo de respuesta del servicio web mod_feedback_get_feedbacks_by_courses

D.5. mod_feedback_get_responses_analysis

Este servicio es utilizado para obtener las respuestas a una encuesta.

Pedido

```
1 GET https://eva.fing.edu.uy/webservice/rest/server.php
2   ? wstoken = <wstoken>
3   & wsfunction = mod_feedback_get_responses_analysis
4   & moodlewsrestformat = json
5   & feedbackid = 2911
```

Figura D.5: Ejemplo de pedido web a mod_feedback_get_responses_analysis

Respuesta

```
1 {
2   "attempts": [
3     {
4       "id": 45781,
5       "courseid": 132,
6       "userid": 15059,
7       "timemodified": 1554046432,
8       "fullname": "Juan Fernández",
9       "responses": [
10        {
11          "id": 46795,
12          "name": "Cédula (sin código identificador)<
13            span class=\"boundaries\"></span>",
14          "printval": "1234567",
15          "rawval": "1234567"
16        },
17        {
18          "id": 46796,
19          "name": "Edad",
20          "printval": "21-24 años",
21          "rawval": "2"
22        },
23        {
24          "id": 46797,
25          "name": "¿De dónde eres?",
26          "printval": "Montevideo",
27          "rawval": "1"
28        }
29      ]
30    }
31  ]
32 }
```

```

28     {
29         "id": 46798,
30         "name": "¿En qué tipo de institución cursaste
           bachillerato?",
31         "printval": "Privada",
32         "rawval": "2"
33     },
34     {
35         "id": 46799,
36         "name": "¿Trabajas?",
37         "printval": "Si, Full-Time",
38         "rawval": "3"
39     },
40     {
41         "id": 46800,
42         "name": "Si trabajas, ¿tu trabajo está
           relacionado a lo que estudias?",
43         "printval": "Si",
44         "rawval": "1"
45     },
46     {
47         "id": 46801,
48         "name": "¿Cuántas materias estás cursando?",
49         "printval": "1",
50         "rawval": "1"
51     },
52     {
53         "id": 46802,
54         "name": "¿Cuántas veces cursaste esta materia,
           antes de este año?",
55         "printval": "",
56         "rawval": ""
57     },
58     {
59         "id": 46803,
60         "name": "¿Asistes a las clases teóricas?",
61         "printval": "Sí",
62         "rawval": "1"
63     },
64     {
65         "id": 46804,
66         "name": "¿Asistes a las clases prácticas?",
67         "printval": "Sí",
68         "rawval": "1"
69     },
70     {
71         "id": 46805,
72         "name": "¿Estudias solo o en grupo?",
73         "printval": "Solo",

```

```

74         "rawval": "1"
75     },
76     {
77         "id": 46806,
78         "name": "¿Cuántas horas aproximadamente le
              dedicas a la materia por semana? (Sin
              contar asistencias a clase)",
79         "printval": "Entre 3 y 6",
80         "rawval": "2"
81     },
82     {
83         "id": 46807,
84         "name": "¿Cómo definirías tu nivel de motivaci
              ón con la materia?",
85         "printval": "Medio",
86         "rawval": "2"
87     }
88     ]
89 },
90 ...
91 ],
92 "totalattempts": 1,
93 "anonattempts": [],
94 "totalanonattempts": 0,
95 "warnings": []
96 }

```

JSON D.5: Ejemplo de respuesta del servicio web mod_feedback_get_responses_analysis