

UCLV
Universidad Central
"Marta Abreu" de Las Villas



MFC
Facultad de Matemática
Física y Computación

Departamento de Computación

TRABAJO DE DIPLOMA

**DISEÑO DE UN BLOQUE PARA LA
RECOMENDACIÓN DE RECURSOS
EDUCATIVOS EN MOODLE, BASADO EN LAS
EMOCIONES REFLEJADAS POR LOS
ESTUDIANTES**

Autor: AICENIS MARÍA CASTRO OVES-GARCÍA

Tutores: Ing. Roberto Vicente Rodríguez

Santa Clara, Cuba, noviembre 2021
Copyright©UCLV

UCLV
Universidad Central
"Marta Abreu" de Las Villas



MFC
Facultad de Matemática
Física y Computación

Department of Informatics

DIPLOMA THESIS

DESIGN OF A BLOCK FOR THE RECOMMENDATION OF EDUCATIONAL RESOURCES IN MOODLE, BASED ON THE EMOTIONS REFLECTED BY THE STUDENTS

Author: AICENIS MARÍA CASTRO OVES-GARCÍA

Tutors: Ing. Roberto Vicente Rodríguez

Santa Clara, Cuba, noviembre 2021
Copyright©UCLV

Este documento es Propiedad Patrimonial de la Universidad Central “Marta Abreu” de Las Villas, y se encuentra depositado en los fondos de la Biblioteca Universitaria “Chiqui Gómez Lubian” subordinada a la Dirección de Información Científico Técnica de la mencionada casa de altos estudios.

Se autoriza su utilización bajo la licencia siguiente:

Atribución- No Comercial- Compartir Igual



Para cualquier información contacte con:

Dirección de Información Científico Técnica. Universidad Central “Marta Abreu” de Las Villas.

Carretera a Camajuaní. Km 5½. Santa Clara. Villa Clara. Cuba. CP. 54 830

Teléfonos.: +53 01 42281503-14190



Hago constar que el presente trabajo fue realizado en la Universidad Central “Marta Abreu” de Las Villas como parte de la culminación de los estudios de la especialidad de Ciencia de la Computación, autorizando a que el mismo sea utilizado por la institución, para los fines que estime conveniente, tanto de forma parcial como total y que además no podrá ser presentado en eventos ni publicado sin la autorización de la Universidad.

Firma del autor

Los abajo firmantes, certificamos que el presente trabajo ha sido realizado según acuerdos de la dirección de nuestro centro y el mismo cumple con los requisitos que debe tener un trabajo de esta envergadura referido a la temática señalada.

Firma del tutor

Firma del tutor

Firma del jefe del Dpto

DEDICATORIA

Este trabajo lo dedico con mucho amor a mi abuela; mi más fiel compañera durante mi vida y mi carrera. Gracias, por atravesar conmigo todos mis estados de ánimo y siempre confiar ciegamente en mí. A ti, mi mayor admiración y respeto.

AGRADECIMIENTOS

Agradecer a mi abuela; ya mencionada en la dedicatoria. A mi mamá que siempre ha estado ahí, esperando mis llamadas de última hora pidiendo rescate y siempre con su inagotable paciencia que la caracteriza me responde. A mi papá, que fue mi inspiración para comenzar esta carrera y me impulsó a adentrarme en la computación desde que tengo memoria. A mi tía Haydee, por su inmensa incondicionalidad hacia mí y por siempre ser mi ejemplo. Además, a mi pareja que con calidez me ha acompañado durante este proceso final. De manera general; a toda la familia que desde los comienzos de mi carrera hasta la actualidad me han apoyado y brindado amor.

A mis amigos tanto a los de la universidad como a los de fuera, que fueron parte de este proceso formativo y me apoyaron en todos los sentidos. Hacer mención con mucho cariño a Hayder, Gabi y Edelvi, que marcaron mi paso por la universidad.

Al profesor y tutor Roberto Vicente, por atenderme siempre con tanta paciencia y responderme atentamente en cualquier horario; a él, infinitas gracias por su asesoría y recomendaciones.

RESUMEN

Dentro de los Entornos Virtuales de Aprendizaje (EVA) la plataforma Moodle resalta por ser de las más usadas a nivel mundial y de las más completas. La Universidad Central “Marta Abreu” de Las Villas (UCLV) forma parte de la comunidad que se beneficia y que apoya al constante crecimiento de Moodle. Una carencia de los entornos de aprendizaje en línea es que no cuentan con facilidades que permitan reconocer el estado emocional de los estudiantes en su interacción con los materiales de estudio y actividades de los cursos. El presente trabajo propone un diseño de un bloque para Moodle que permite sugerir a los estudiantes recursos educativos que se ajusten a su estado de ánimo y a las necesidades educativas particulares de cada uno. Este bloque se apoya en el bloque Emociones ya implementado en la UCLV, a partir del cual se obtienen las emociones de los alumnos; díganse alegría, sorpresa, tristeza, asco, ira y miedo; en un curso y tiempo establecido por el profesor. Se fundamenta cómo el algoritmo de filtrado colaborativo que brinda la biblioteca MLlib de Apache Spark, es capaz de sugerir materiales con precisión.

Palabras clave: reconocimiento de emociones, filtrado colaborativo, sistemas de recomendación, Moodle.

ABSTRACT

Within Virtual Learning Environments (EVA), the Moodle platform stands out for being one of the most used worldwide and one of the most complete. The Central University "Marta Abreu" of Las Villas (UCLV) is part of the community that benefits and supports the growth of Moodle. A shortcoming of the online learning environment is that it does not have facilities that allow recognizing the emotional state of students in their interaction with study materials and course activities. This work presents the design of a block for Moodle that allows students to suggest educational resources that adjust to their state of mind and the particular educational needs of each one. This block is based on the Emotions block already implemented in the UCLV from which the emotions of the students are obtained; tell yourselves joy, surprise, sadness, disgust, anger and fear; in a course and time established by the teacher. It is justified, how the collaborative filtering algorithm provided by the Apache Spark MLlib library is capable of accurately suggesting materials.

Abstract Keywords: emotion recognition, collaborative filtering, recommendation systems, Moodle.

TABLA DE CONTENIDO

| | |
|--|-----------|
| INTRODUCCIÓN | 1 |
| 1. TECNOLOGÍAS PARA LA IMPLEMENTACIÓN DE SISTEMAS DE RECOMENDACIÓN MEDIANTE FILTRADO COLABORATIVO, APLICADOS EN ENTORNOS VIRTUALES DE APRENDIZAJE | 6 |
| 1.1. Las TIC y su papel en la educación | 6 |
| 1.2. Los Entornos Virtuales de Aprendizaje (EVA), herramientas tecnológicas educativas | 8 |
| 1.2.1. Plataforma Moodle | 10 |
| 1.3. Las emociones y su vinculación en la educación | 15 |
| 1.3.1. Emociones, concepto | 15 |
| 1.3.2. Clasificación de las emociones | 16 |
| 1.4. Bloque Emociones, funcionalidades | 17 |
| 1.5. Sistemas de recomendación | 18 |
| 1.5.1. Clasificación | 18 |
| 1.6. Filtrado colaborativo | 23 |
| 1.6.1. Clasificación | 24 |
| 1.6.2. Ventajas fundamentales de los sistemas de FC | 24 |
| 1.6.3. Problemas de los SR con FC | 25 |
| 1.7. Conclusiones del capítulo | 26 |
| 2. PROPUESTA DE DISEÑO PARA UN RECURSO DE RECOMENDACIÓN DE MATERIALES EDUCATIVOS EN LA PLATAFORMA MOODLE | 28 |
| 2.1. Diferentes marcos de trabajo para el Aprendizaje Automático, los Sistemas de Recomendación y el Filtrado Colaborativo | 28 |
| 2.1.1. Mahout | 28 |
| 2.1.2. BigML | 29 |
| 2.1.3. H2O | 29 |
| 2.1.4. Spark | 30 |

| | | |
|------------------------|---|-----------|
| 2.2. | Comparación entre los sistemas de análisis de Big Data y AA descritos | 31 |
| 2.2.1. | Mahout vs. Spark MLlib | 31 |
| 2.2.2. | Spark MLlib vs. H2O | 33 |
| 2.2.3. | Conclusiones parciales de la sección | 34 |
| 2.3. | Spark para el desarrollo de SR y FC | 34 |
| 2.3.1. | Spark, características | 34 |
| 2.3.2. | Componentes de Spark | 35 |
| 2.4. | Ventajas de utilizar Apache Spark para el Big Data y el AA | 38 |
| 2.5. | MLlib, la biblioteca de Spark para el AA | 38 |
| 2.6. | Sistema de recomendaciones y filtrado colaborativo con MLlib | 39 |
| 2.6.1. | Retroalimentación explícita vs implícita | 40 |
| 2.6.2. | Escala del parámetro de regularización | 41 |
| 2.6.3. | Problema del arranque en frío | 41 |
| 2.6.4. | RDD | 41 |
| 2.6.5. | Algoritmo de mínimos cuadrados alternos (ALS) | 42 |
| 2.6.6. | PySpark para una implementación en Python | 43 |
| 2.6.7. | Conclusiones parciales de la sección | 44 |
| 2.7. | Diseño de un bloque para Moodle basado en SR y FC | 44 |
| 2.7.1. | Propuesta de diseño gráfico | 45 |
| 2.7.2. | Diseño UML para el bloque Recomendaciones | 46 |
| 2.7.3. | Actores | 47 |
| 2.7.4. | Casos de uso | 48 |
| 2.7.5. | Diagrama de actores y casos de uso | 48 |
| 2.7.6. | Descripción de los casos de uso | 49 |
| 2.7.7. | Descripción del diseño | 54 |
| 2.7.8. | Propuesta de estructura de base de datos para el bloque | 55 |
| 2.8. | Conclusiones del capítulo | 55 |
| CONCLUSIONES | | 57 |
| RECOMENDACIONES | | 59 |
| REFERENCIAS | | 60 |

FIGURAS

| | |
|---|----|
| 1.1. Esquema funcional de un SR | 19 |
| 1.2. Sistema de recomendación basado en Contenido donde CR es el componente de recomendación (Vera Sancho, 2016) | 20 |
| 1.3. Sistema de recomendación basado en conocimiento donde CR es el componente de recomendación(Vera Sancho, 2016) | 21 |
| 1.4. Sistema de recomendación basado en filtros colaborativos donde CR es el componente de recomendación(Vera Sancho, 2016) | 21 |
| 1.5. Sistema de recomendación híbrido donde CR es el componente de recomendación(Vera Sancho, 2016) | 21 |
| 2.1. Comparación de la eficiencia de ALS implementado en Spark y en Mahout según experimento realizado por (Yabuz y Meng, 2014) | 33 |
| 2.2. Estructura de Apache Spark (Karau et al., 2015) | 36 |
| 2.3. Vista del profesor | 45 |
| 2.4. Vista del estudiante | 45 |
| 2.5. Modelación de los actores | 47 |
| 2.6. Modelación de los casos de uso y actores | 48 |
| 2.7. Propuesta de tabla para el bloque Recomendaciones | 55 |

TABLAS

| | |
|---|----|
| 1.1. Diferencias entre los SR (Carrillo, 2013) | 22 |
| 2.1. Comparación breve de algunas herramientas para el trabajo con BigData y ML según (Zomaya y Sakr, 2017) y (H2O.ai, 2021) | 31 |
| 2.2. Especificación del caso de uso <i>Registrarse</i> | 49 |
| 2.3. Especificación del caso de uso <i>Entrar a un curso</i> | 50 |
| 2.4. Especificación del caso de uso <i>Seleccionar un material.</i> | 50 |
| 2.5. Especificación del caso de uso <i>Seleccionar nuevo material recomendado.</i> | 51 |
| 2.6. Especificación del caso de uso <i>Interactuar con el bloque Emociones.</i> | 52 |
| 2.7. Especificación del caso de uso <i>Interactuar con el bloque Recomendaciones.</i> | 53 |
| 2.8. Especificación del caso de uso <i>Añadir materiales.</i> | 54 |

INTRODUCCIÓN

En los últimos dos años aproximadamente, el mundo se ha enfrentado a una situación de pandemia que en sus inicios paralizó muchas de las actividades sociales básicas. La educación en todos los niveles de enseñanza fue uno de los procesos que se vio interrumpido ante la intratable situación mundial.

Las Tecnologías de la Información y las Comunicaciones (TIC) resultaron ser las heroínas del sistema educacional; pues brindaron una alternativa a través de los Entornos Virtuales de Aprendizaje (EVA), que daba continuidad al proceso educativo. Cuando se alude a los EVA; conocidos en inglés como Virtual Learning Environment (VLE) o Learning Management System (LMS); se hace referencia a entornos virtuales de aprendizaje en un ambientes web que buscan simular las componentes metodológicas, didácticas y psicosociales que se presentan en un aula de clases. Habitualmente los EVA están instalados en un servidor o Intranet que se encarga de mantener, gestionar y almacenar actividades de formación virtual. Estas formaciones pueden darse en un entorno totalmente virtual o como apoyo a formaciones presenciales, aunque su fin es común; facilitar al alumno el progreso en el proceso de aprendizaje.

Algunas de las características fundamentales y ventajas que identifican a los EVA:

- Flexibilidad: una de las particularidades más provechosas que presenta la modalidad de educación a distancia es la flexibilidad de los horarios.
- No requiere una gran inversión: la reducción de la inversión resulta una ventaja significativa de la educación a distancia, en contraposición con la modalidad presencial.
- Escalabilidad: los EVA tienen la capacidad de funcionar igualmente con un número pequeño o grande de usuarios.
- Necesidad de contar con servicio de Internet: una particularidad en este tipo de aprendizaje a distancia es la necesidad de poder contar con Internet para participar en las clases virtuales. Evita el desplazamiento hacia una institución y hasta se puede elegir el horario más conveniente, pero se debe estar pendiente de la conexión a Internet en el espacio elegido.

- Espacios de participación: un entorno virtual de aprendizaje debe contar con espacios de participación y/o discusión como los foros, donde se debaten preguntas y respuestas. En este contexto, los alumnos manifiestan sus críticas o sugerencias a los facilitadores del curso. Además, se agiliza la comunicación entre profesores y alumnos, lo que enriquece el aprendizaje y la colaboración en la resolución de inquietudes. Con los espacios de participación, se garantiza la interactividad. El término "interactividad" hace referencia a la forma en que estas plataformas propician la interacción de los estudiantes tanto con los materiales de enseñanza como con otros participantes.
- Estandarización: los EVA tienen la posibilidad de importar y exportar cursos en formatos estándar como SCORM.
- Constancia en las clases: la presencia en las clases se marca de acuerdo con el contenido. En los cursos de educación a distancia, se puede acceder al número de lecciones que se desee por día. Lo cual no significa que el alumno deba ser penalizado en el caso de no poder estudiar durante uno o más días.

Existen varias herramientas de EVA, una de las más conocidas y extendida a nivel mundial es Moodle. Moodle es una de las plataformas de aprendizaje en línea más completas y de código abierto; gracias a esta última característica, muchísimos desarrolladores pueden trabajar en la plataforma introduciéndole mejoras y contribuyendo a su crecimiento. Además, tiene licencia gratuita; permitiendo de esta forma el acceso libre de costo a los usuarios. Según el sitio Web oficial de Moodle (Moodle, 2021), la plataforma está facilitada en más de 120 idiomas y es utilizada por más de 200 millones de usuarios.

En la actualidad, otro campo de las TIC que ha apoyado a la sociedad con disímiles alternativas es la Inteligencia Artificial (IA). Dentro de la IA una rama que está en constante desarrollo es el Aprendizaje Automático (AA) o en inglés Machine Learning (ML). El AA fue definido por (Samuel, 1959) como el campo de estudio que da a los computadores la capacidad de aprender sin ser programados de manera explícita. Según (Mitchell, 1997), desde un punto de vista de ingeniería, lo define como un programa de computador que aprende de una experiencia E, con respecto a una tarea T y una medida de rendimiento R, si su rendimiento en T, medido por R, mejora con la experiencia E. También (Gerón, 2019) lo define como ciencia (y arte) de programar computadores para que aprendan a partir de los datos. Junto con estas definiciones, este tipo de IA ha ido creciendo con bastante rapidez debido a la cantidad de datos disponibles en la actualidad (Big Data) e Internet.

En la Universidad Central "Marta Abreu" de Las Villas (UCLV) en 2019 se implementó un bloque para Moodle llamado bloque Emociones para la detección de emociones en los estudiantes (Vázquez Basulto, 2019). Este bloque fue orientado fundamentalmente al profesor; su objetivo es guardar las emociones que se le reconocen a los estudiantes mientras consultan materiales de estudio o in-

teractúan con las actividades de la plataforma. Teniendo en cuenta el rol significativo que juegan las emociones en la vida y sobre todo en los procesos educativos; el bloque brinda este conocimiento al profesor sobre sus estudiantes, para que pueda planificar actividades de atención personalizada que respondan a las necesidades educativas de cada alumno.

Actualmente, los distintos niveles de enseñanza, casi en su totalidad, se han auxiliado y han sido sostenidos gracias a los EVA. A pesar de las experiencias positivas que estos entornos aportan; pues han facilitado la continuidad de la educación; también dejaron insatisfacciones en los estudiantes. La pérdida de la motivación hacia el estudio, la frustración ante no encontrar soluciones a problemáticas académicas, entre otras, son algunas de las reacciones que han tenido muchos alumnos a raíz de la ausencia de clases presenciales.

El trabajo para mejorar las plataformas de aprendizaje virtual no cesa; debido a que en las clases presenciales, el estudiante sentía que sus necesidades eran atendidas con mayor prioridad; hecho que los EVA no siempre garantizan. Además, al interactuar directamente y en tiempo real con profesores y compañeros; el estudiante encontraba el conocimiento con calidez y en un ambiente con el cual estaba familiarizado. Precisamente, estas características de las clases presenciales que son carencias en las aulas virtuales; resultan un reto para los desarrolladores y profesores.

Resulta de interés extender la funcionalidad del Moodle. A partir de las emociones que el bloque Emociones reconoce y guarda; una alternativa que tome en consideración las emociones y apoye a los estudiantes en su autoestudio podría ser la recomendación de materiales educativos, que les resulten de interés y sean afines con su perfil. En este sentido, se utilizarían las emociones que se registren para determinar si el material que el estudiante está consultando en un momento dado, le está resultando útil o no. Esta variante, facilitaría el trabajo de la búsqueda de materiales de estudio de los alumnos. Además, los intereses del estudiante estarían siendo reconocidos, al recibir sugerencias de materiales que necesita y que le pueden resultar más útiles que el que esté revisando en ese preciso momento.

El sistema que se propone incorporar al Moodle como complemento del bloque Emociones, sería otro bloque llamado Recomendaciones y se implementaría utilizando los Sistemas de Recomendación (SR) que brinda el AA. Existen varios tipos de SR, de entre ellos el Filtrado Colaborativo (FC) se ajusta a las necesidades técnicas de implementar un bloque en Moodle para sugerir materiales de estudio según las emociones e intereses de los alumnos. El papel del profesor en los resultados que arroje este bloque a los estudiantes es fundamental; pues de la variedad de documentos con diversos niveles de complejidad que el profesor añada a la base de datos donde se guardarán; depende la calidad y variedad de las sugerencias que reciban los estudiantes. Otro aspecto importante en este sentido es que, a medida que el SR se va nutriendo con nuevos estudiantes y materiales, el aprendizaje se hace más efectivo y a su vez la precisión de las recomendaciones.

La implementación de un bloque para Moodle que sea capaz de sugerir materiales de estudio a los alumnos según su estado de ánimo es una alternativa interesante para el crecimiento de la plataforma y resulta un apoyo espléndido tanto para estudiantes como para profesores.

A partir de la anterior afirmación y las razones expuestas previamente; se puede justificar el siguiente planteamiento.

La plataforma Moodle no cuenta con facilidades que sugieran al estudiante materiales de estudio acordes a su nivel y necesidad educativa; el estudiante debe realizar la búsqueda fuera del entorno; hecho que complejiza el estudio ocupándole tiempo y recursos.

Del planteamiento anterior se derivan las siguientes interrogantes investigativas:

- ¿Qué elementos deben ser considerados en la propuesta de una facilidad de recomendación de recursos educativos?
- ¿Cómo debe ser el diseño de un recurso de recomendación de materiales educativos que considere el estado emocional de los estudiantes?
- ¿Cómo lograr desde el Moodle implementar facilidades para que a partir de las emociones detectadas en el comportamiento de los estudiantes se puedan recomendar recursos educativos?

Se proponen los siguientes objetivo general y objetivos específicos, a ser cumplidos mediante el desarrollo del presente proyecto investigativo.

Objetivo General:

Diseñar un bloque para Moodle que recomiende recursos educativos a los alumnos basado en sus emociones.

Objetivos específicos:

- Estudiar las tendencias actuales sobre las tecnologías que soportan la implementación de Sistemas de Recomendación basados en Filtrado Colaborativo.
- Fundamentar los recursos informáticos (lenguajes y bibliotecas) que deben ser empleados en la implementación de un sistema de recomendación de recursos educativos.
- Realizar el diseño de un sistema de recomendación de recursos educativos, tomando en consideración las emociones del alumno al interactuar con estos.

El presente informe se estructura en dos capítulos. En el capítulo primero se hace una revisión del estado del arte con respecto a las características de los EVA, las potencialidades del Moodle como plataforma educativa, las emociones humanas y su influencia en la educación, así como las funcionalidades del bloque Emociones implementado para Moodle. En las últimas secciones se revisan los conceptos de Sistema de Recomendación y Filtrado Colaborativo. En el segundo capítulo se establece una comparación entre las diferentes tecnologías que cuentan con bibliotecas para implementar Sistemas de Recomendación y se explica con mayor énfasis el entorno Apache Spark y su biblioteca

de AA, MLlib. El capítulo finaliza con una propuesta de diseño de implementación de un bloque en Moodle para la recomendación de materiales educativos a los estudiantes, basadas en sus emociones.

1. TECNOLOGÍAS PARA LA IMPLEMENTACIÓN DE SISTEMAS DE RECOMENDACIÓN MEDIANTE FILTRADO COLABORATIVO, APLICADOS EN ENTORNOS VIRTUALES DE APRENDIZAJE

En este capítulo se presenta una revisión bibliográfica para introducir y facilitar la comprensión de los Sistemas de Recomendación (SR) mediante Filtrado Colaborativo (FC) aplicados a plataformas virtuales de aprendizaje como Moodle. En las primeras secciones se brinda una introducción a las Tecnologías de la Información y la Comunicación (TIC) y como parte de estas se explican los Entornos Visuales de Aprendizaje (EVA). Se dedica una sección a explicar las características fundamentales de la plataforma Moodle debido a que es uno de los EVA más utilizado y completo. En otra sección se explica el bloque de Moodle implementado en la Universidad Central “Marta Abreu de las Villas” (UCLV), llamado Emociones (Vázquez Basulto, 2019) para el reconocimiento de emociones faciales en los estudiantes mientras interactúan con la plataforma. Las secciones finales abordan lo relacionado con los SR dentro de la rama del Aprendizaje Automático (AA) de la Inteligencia Artificial (IA). Se finaliza el capítulo con conclusiones parciales que resumen toda la teoría que engloba el diseño de un recurso en Moodle de SR basado en FC que se propone en el capítulo dos.

1.1 Las TIC y su papel en la educación

La tecnología y en concreto las TIC, han influido considerablemente en el estilo de vida de la humanidad y fue haciéndose evidente la importancia de las TICs en todas las áreas de la vida y por supuesto en la educación. Las TICs se han implementado en los procesos educativos para complementar y facilitar el desarrollo de conocimientos en cualquier nivel de enseñanza. Según afirma la UNESCO en su página oficial (UNESCO, 2020), las TICs complementan, enriquecen y transforman la educación. La era del Internet exige que el mundo educativo evolucione, por lo tanto, los profesionales de la educación tienen múltiples razones para aprovechar las oportunidades que ofrecen las TICs, todo con la finalidad de dirigirse hacia un nuevo paradigma educativo que se centre en potenciar la eficiencia en la actividad independiente de los estudiantes.

En la actualidad existen muchas plataformas en Internet que ofrecen educación en línea, hecho que

se hace posible gracias a las TICs, debido a la conexión que ha sido posible establecer entre los individuos a nivel mundial. Teniendo en cuenta la situación mundial actual, la pandemia por Covid-19 y el hecho de que hace más de un año se han estado impartiendo clases en línea en un gran número de países, el uso de las TIC ha tenido un conspicuo incremento y ha sido crucial para la continuidad de los procesos educativos.

El uso de las tecnologías en el ámbito educativo impacta de manera positiva en el aprendizaje académico. Por un lado, aumenta la motivación e interactividad de los estudiantes. Por otro, fomenta la cooperación entre alumnos e impulsan la iniciativa y la creatividad (Ignite, 2021).

La transformación que ha sufrido las TIC, han logrado convertirse en instrumentos educativos, capaces de mejorar la calidad educativa del estudiante, revolucionando la forma en que se obtiene, se maneja y se interpreta la información. La incorporación de las TIC, a la educación se ha convertido en un proceso, cuya implicancia, va mucho más allá de las herramientas tecnológicas que conforman el ambiente educativo, se habla de una construcción didáctica y la manera cómo se pueda construir y consolidar un aprendizaje significativo en base a la tecnología (Hernandez, 2017).

Dentro de los roles que asumen cada agente educativo, los estudiantes actuales, utilizan las herramientas tecnológicas para facilitar el aprendizaje, esta evolución surgió desde las primeras concepciones con la calculadora, el televisor, la grabadora, entre otras. Sin embargo, el progreso ha sido tal que los recurso tecnológicos se han convertido en recursos educativos, donde la búsqueda por mejorar el aprendizaje trae consigo la tarea de involucrar la tecnología con la educación. En este sentido, la docencia viene completando el proceso de enseñanza-aprendizaje (Jiménez, 2015).

Según (Suárez y Najjar, 2014), la educación como aspecto relevante en la vida del ser humano ha combinado junto a las TIC un nuevo ambiente de aprendizaje donde el estudiante es capaz de convertirse en el protagonista de su propio aprendizaje, donde el tiempo y la flexibilidad, están jugando un rol importante en una educación que cada vez más, se virtualiza y donde lo virtual se ha convertido en una revolución y donde las nuevas tecnologías convergen en plantear nuevos paradigmas educativos y pedagógicos. La educación es parte de la tecnología y cada vez más se exige la alfabetización electrónica, considerándose una competencia indispensable para el estudiante.

Retos educativos frente a las TIC Son conocidos los grandes cambios que ha traído la tecnología a la educación, por lo cual se ha de considerar como un reto principal la forma de unificar el enfoque tecnológico con el proceso de enseñanza y aprendizaje. Otro desafío también resulta modificar los métodos de aprendizaje de forma que este proceso vaya de la mano con el rápido avance y constante evolución tecnológica. En este sentido, el docente desempeña un papel fundamental actuando como guía ante un proceso educativo con nuevos métodos de enseñanza y en él recae la tarea de velar por la adaptación de los estudiantes a las nuevas técnicas, así como de estructurar de forma eficiente el

1.2 Los Entornos Virtuales de Aprendizaje (EVA), herramientas tecnológicas educativas

ambiente de aprendizaje.

Un aspecto importante a tener en cuenta es que actualmente, en las clases presenciales que se desarrollan en un aula, se necesitarán nuevos espacios que complementen el conocimiento mediante el uso de medios tecnológicos entre estudiantes y docentes, es decir, el avance que representan las TICs resulta innegable por lo que es fundamental desarrollar técnicas educativas que impulsen y aprovechen sus ventajas.

Para resumir, sobre la importancia de las TICs en la educación expresó (Coll, 2004) :

“No es en las TIC, sino en las actividades que llevan a cabo profesores y estudiantes gracias a las posibilidades de comunicación, intercambio, acceso y procesamiento de la información que les ofrecen las TIC, donde hay que buscar las claves para comprender y valorar el alcance de su impacto en la educación escolar, incluido su eventual impacto sobre la mejora de los resultados del aprendizaje.”

1.2 Los Entornos Virtuales de Aprendizaje (EVA), herramientas tecnológicas educativas

La revolución tecnológica ha cambiado nuestra manera de entender todo nuestro entorno. Las nuevas generaciones viven una realidad muy distinta de la que existía en nuestra infancia y, muy probablemente, en pocos años nada de lo que conocemos tendrá que ver con lo que existe ahora. Los cambios son muy rápidos y la educación debe facilitar la adaptación de los alumnos a ellos. Para ello, la utilización de un entorno virtual de aprendizaje o EVA puede ser una herramienta de gran utilidad.

Un entorno virtual de aprendizaje EVA, a un espacio educativo que se aloja en una web. Se trata de un conjunto de herramientas que facilitan el aprendizaje y que conforman un espacio en el que los alumnos y profesores pueden interactuar de forma remota y realizar todas las tareas relacionadas con la docencia sin necesidad de una interacción física.

Es una herramienta o conjunto de herramientas que surge inevitablemente de las necesidades de la sociedad de la información, que trata de imitar, y en muchos casos mejorar, la realidad de un entorno de aprendizaje como es un aula, pero de manera virtual, utilizando la informática. Pueden ser con formaciones completamente virtuales (e-learning) o servir de apoyo a formaciones presenciales (blended learning) (Baque y Marcillo, 2020).

Uno de los notables beneficios de utilizar un entorno virtual de aprendizaje es que dan libertad al alumno, tanto en el sentido de otorgarle flexibilidad horaria, como para orientar su aprendizaje por sí mismo. Este hecho hace evidente el cambio de roles entre docentes y alumnos, dejando los primeros de ser unos meros transmisores de información, para convertirse en guías que dirigen al alumno y proponen recursos, y convirtiéndose los segundos en personajes responsables que construyen su propio aprendizaje de manera activa y colaborativa (Martín y Sarmiento, 2019). Además, amplía sustancialmente las posibilidades de acceso a los contenidos, ya que el alumno puede completar su formación desde cualquier lugar o dispositivo.

1.2 Los Entornos Virtuales de Aprendizaje (EVA), herramientas tecnológicas educativas

A continuación se presentan algunas características fundamentales de los EVA según (Lima Montenegro y Fernandez Nodarse, 2017):

- **Flexibilidad:** una de las particularidades más provechosas que presenta la modalidad de educación a distancia es la flexibilidad de los horarios. En cuanto a la modalidad presencial, existe la necesidad de organizarse para asistir frecuentemente a clases en los horarios preestablecidos. Quienes prefieran el entorno virtual podrán hacerlo desde el sitio y hora de su preferencia. El EVA es quien se adapta a la estructura institucional y los planes de estudio.
- **No requiere una gran inversión:** la reducción de la inversión resulta una ventaja significativa de la educación a distancia, en contraposición con la modalidad presencial. En el caso del formato presencial, resulta necesario costear una gran cantidad de dinero, en comparación con la modalidad de educación a distancia.
- **Escalabilidad:** capacidad del EVA de funcionar igualmente con un número pequeño o grande de usuarios.
- **Necesidad de contar con servicio de Internet:** una particularidad en este tipo de aprendizaje a distancia es la necesidad de poder contar con Internet para participar en las clases virtuales. Evita el desplazamiento hacia una institución y hasta se puede elegir el horario más conveniente, pero se debe estar pendiente de la conexión a Internet en el espacio elegido. Si se está desconectado, no se puede acceder a los contenidos ni a los materiales instruccionales. De igual forma, una conexión lenta de Internet puede retrasar los estudios y hacer que se necesite más tiempo para completarlos.
- **Espacios de participación:** un entorno virtual de aprendizaje debe contar con espacios de participación y/o discusión, mediante los foros se debaten preguntas y respuestas. En este contexto, los alumnos manifiestan sus críticas o sugerencias a los facilitadores del curso. Además, se agiliza la comunicación entre profesores y alumnos, lo que enriquece el aprendizaje, y se puede colaborar en la resolución de inquietudes. El usuario es el protagonista de su formación. El término "interactividad" hace referencia a la forma en que estas plataformas propician la interacción de los estudiantes tanto con los materiales de enseñanza como con otros participantes. En lugar de ser textos en línea, los EVA se proponen como entornos interactivos de construcción de conocimiento.
- **Estandarización:** Posibilidad de importar y exportar cursos en formatos estándar como SCORM.
- **Constancia en las clases:** la presencia en las clases se marca de acuerdo con el contenido, en el contexto de la modalidad de educación presencial, el alumno debe asistir constantemente a clases. En los cursos de educación a distancia, se puede acceder al número de lecciones que se desee por día. Lo cual no significa que el alumno deba ser penalizado en el caso de no poder estudiar durante uno o más días.

1.2 Los Entornos Virtuales de Aprendizaje (EVA), herramientas tecnológicas educativas

Respecto a la comunicación en los EVA puede ser de dos formas según (Araque et al., 2018):

- Sincrónica: se utilizan mucho los encuentros virtuales en tiempo real mediante chat o una video-llamada.
- Asíncrona: la interacción entre alumno y profesor se basa principalmente en foros y mensajería.

Existen diversas herramientas de entornos virtuales para el aprendizaje, el ejemplo más extendido de este tipo de herramienta es Moodle, al ser un software gratuito y regula entre los más conocidos y utilizados. Otros entornos son Atutor o Docebo. Lo adecuado es que cada docente explore las herramientas y escoja la más adecuada a las necesidades de su curso.

1.2.1 Plataforma Moodle

Moodle es una plataforma de aprendizaje diseñada para proporcionarle a educadores, administradores y estudiantes un sistema integrado único, robusto y seguro para crear ambientes de aprendizaje personalizados. La plataforma tiene la confianza de tanto grandes como pequeñas instituciones y organizaciones, incluyendo a Shell, La Escuela Londinense de Economía (London School of Economics), La Universidad Estatal de Nueva York, Microsoft y la Universidad Abierta del Reino Unido (Open University). El número de usuarios de Moodle a nivel mundial, de más de 200 millones de usuarios (en agosto del 2020), entre usuarios académicos y empresariales, lo convierten en la plataforma de aprendizaje más ampliamente utilizada del mundo (Moodle, 2021).

La palabra Moodle es un acrónimo y significa: Modular Object Oriented Dynamic Learning Environment, en español, Entorno de Aprendizaje Dinámico Orientado a Objetos y Modular. La plataforma es un Sistema de Gestión del Aprendizaje (SGA), en inglés LMS (Learning Management System) que contiene las herramientas y recursos necesarios para la creación de cursos en línea (Yeou, 2016).

Respecto a la comunicación, Moodle soporta ambos tipos característicos de los EVA, síncrona y asíncrona. Precisamente las ventajas y opciones que brinda Moodle para la comunicación figuran entre sus tantas potencialidades, por ejemplo, permite el diálogo y la discusión entre todas las personas implicadas, (uno-uno, uno-muchos, muchos-muchos). La comunicación que facilita Moodle favorece la interacción en cuatro niveles: alumno-profesor, proporcionando motivación, retroalimentación (feedback) y diálogo entre profesor-alumno; alumno-contenido, donde el alumno recibe información intelectual del material; alumno-alumno, pues se produce intercambio de información, ideas, diálogo, etc.; y alumno-interfaz, es decir, entre el alumno y la tecnología que transmite la información.

La comunicación interpersonal es uno de los pilares fundamentales en Moodle, ya que permite el diálogo y la discusión entre todas las personas implicadas, (uno-uno, uno-muchos, muchos-muchos). De esta forma se favorece la interacción en cuatro niveles: alumno-profesor, proporcionando motivación, retroalimentación (feedback) y diálogo entre profesor-alumno; alumno-contenido, donde el alumno recibe información intelectual del material; alumno-alumno, pues se produce intercambio

1.2 Los Entornos Virtuales de Aprendizaje (EVA), herramientas tecnológicas educativas

de información, ideas, diálogo, etc.; y alumno-interfaz, es decir, entre el alumno y la tecnología que transmite la información (Muñoz et al., 2020).

El sistema hace posible manejar de forma eficiente la gestión académica y administración de los alumnos, facilitando las tareas de matriculación, consulta del expediente académico, expedición de certificados, etc. También, Moodle simplifica la tarea de control por parte del profesor gracias a los diferentes tipos de actividades de evaluación que brinda. Por ejemplo, permite la creación de ejercicios de evaluación y autoevaluación, así como la posibilidad de crear bases de datos con preguntas y la configuración de diferentes pruebas. Además, facilita el acceso directo a diversos recursos de aprendizaje como: hipermedias, textos, imágenes, secuencias de vídeo y/o audio, ejercicios y tutoriales (Nozawa, 2011).

1.2.1.1. Características fundamentales de la plataforma Moodle

El Moodle se identifica por tener las características fundamentales de los EVA mencionadas en subsecciones anteriores, a continuación se remune algunas de sus características principales según (Oi et al., 2017):

- Actualizado: debido a la gran comunidad que existe alrededor de Moodle, con frecuencia se lanzan nuevas versiones que incorporan mejoras y novedades para adaptarse cada vez más a las diferentes necesidades de docentes y alumnos.
- Flexible y personalizable: existen diversas plantillas predefinidas para seleccionar el aspecto que se quiera dar a la plataforma. Además, al ser un proyecto open source permite que usuarios con conocimientos técnicos puedan desarrollar sus propios temas.
- Interfaz intuitiva: la interfaz de Moodle facilita la rápida familiarización de nuevos usuarios con las opciones que brinda la plataforma, por ejemplo cuenta con un panel de control muy intuitivo para facilitar la gestión de los cursos.
- Tablero personalizado: en la plataforma se pueden organizar los cursos por semanas, temas, tipo de contenido, etc.
- Actividades y herramientas colaborativas: cuenta con wikis, foros, diarios, cuestionarios, recursos, consultas, encuestas, tareas, chats y talleres para facilitar el aprendizaje de los alumnos.
- Estadísticas: acceso a los informes de actividad de los alumnos, rendimiento por cada módulo e historial de participación en el curso.
- Calendario: Moodle incluye una aplicación todo en uno que te permite incluir el calendario académico, fecha de entregas de trabajos, exámenes, reuniones o incluso añadir notas personales.
- Editor de texto simple e intuitivo: la plataforma dispone de un editor para dar formato a los textos, compatible en todos los navegadores y dispositivos.

1.2 Los Entornos Virtuales de Aprendizaje (EVA), herramientas tecnológicas educativas

- **Multiidioma:** brinda la posibilidad de configurar el sitio o cambiar el idioma de Moodle para usuarios u organizaciones multilingües. Las capacidades multilingües de Moodle aseguran que no haya limitaciones lingüísticas para aprender en línea. La comunidad Moodle ha traducido Moodle a más de 120 idiomas según el sitio oficial de Moodle (Moodle, 2021).

1.2.1.2. Estructura de Moodle

La estructura básica de Moodle está organizada alrededor de cursos. Estos son básicamente, páginas o áreas dentro de Moodle en donde los profesores pueden presentar sus recursos y actividades a los estudiantes. Normalmente estas páginas incluyen un número de secciones centrales en donde se muestran los materiales y en donde hay bloques laterales que ofrecen información o características extra.

En dependencia de cómo el profesor lo establezca, los cursos pueden tener contenido para un año de estudios, para una sesión única o cualquier otra variante. Respecto a la inscripción de los estudiantes en los cursos, depende del establecimiento; por ejemplo, pueden tener auto-inscripción, ser inscritos de forma manual por su maestro o automáticamente por el administrador. Otro aspecto organizativo y estructural es que los cursos están organizados en categorías (Moodle, 2021). El núcleo de Moodle (conocido como el Moodle Core) proporciona los mecanismos necesarios para implementar la plataforma educativa y está rodeado de varios plugins con diferentes funciones (Rodríguez et al., 2019).

1.2.1.3. Módulos

Una de las potencialidades que brinda la estructura de Moodle es que al ser open source hace posible el desarrollo de módulos y herramientas no incluidas en el paquete estándar, útiles para la creación de nuevos escenarios didácticos que se ajusten a las necesidades educativas tanto de docentes como de alumnos.

A continuación se resumen algunas de las herramientas y módulos que conforman a la plataforma Moodle y resultan fundamental para su funcionamiento eficiente e innovador (Leris López et al., 2015):

- **Actividades:** las actividades garantizan la interacción del estudiante con el profesor y la plataforma en sí. Las actividades se pueden englobar según las utilidades y funcionalidades que brindan; primero las que tienen como fin la producción individual, dígame cuestionarios y tareas; las destinadas a la producción grupal y la interacción, es decir, chats, foros, talleres, bases de datos, glosarios, wikis y tareas grupales; por último las de carácter integrado, que son, lecciones o materiales empaquetados en estándares SCORM o IMS.
- **Recurso:** objeto presentado por el profesor para asistir el aprendizaje, que no implica la inter-

1.2 Los Entornos Virtuales de Aprendizaje (EVA), herramientas tecnológicas educativas

vención del alumno. Los más usuales son la creación de una carpeta o un archivo accesibles al alumnado, la edición de una "etiqueta" en algún lugar del curso, la creación de una página interna del curso o de un enlace a una URL externa o la edición de un "libro" (recurso no interactivo con páginas y subpáginas conectadas que admiten diversos elementos multimedia).

- **Bloques:** elementos situados en los laterales de la caja de navegación, son complementos o utilidades añadidos a voluntad al curso.
- **Plugins:** complementos o códigos contribuidos que permiten añadir funcionalidades extra a Moodle. La LMS posee un paradigma modular, es decir, un core al que le pueden ser instalados y desinstalado diferentes extensiones. Todas pueden desinstalarse salvo el módulo foro. Encontramos un conjunto flexible de herramientas que permiten a los usuarios de Moodle extender las características del sitio. Hay cientos de complementos que se ofrecen en la Comunidad, que amplían las posibilidades de la funcionalidad principal de Moodle. Cada complemento se mantiene en el directorio de complementos de Moodle. En octubre de 2020 hay disponibles 1738 complementos para Moodle.

Existen diferentes tipos de módulos denominados según (González y Medina, 2011) en: recursos transmisivos, interactivos y colaborativos. Los recursos transmisivos de los que dispone Moodle son las páginas de texto, páginas web, enlaces a archivos o web, directorio, etiqueta y libro. Los recursos interactivos son las lecciones, cuestionarios, SCORM, glosarios y tareas. Por último, los recursos o herramientas colaborativas son foros, talleres y wikis. Por otra parte están las herramientas de comunicación que Moodle brinda, que son el correo electrónico, chats, mensajes, consultas y encuestas.

1.2.1.4. Bloques

Tomado de la página oficial de documentos de Moodle (MoodleDocs, 2021), los bloques son ítems que pueden añadirse a la columna izquierda, derecha o central de cualquier página de Moodle, incluso al centro del Tablero. Los bloques permiten añadir al Moodle funcionalidades para una determinada necesidad educativa. Gracias a que la plataforma es de código abierto y libre, muchos colaboradores implementan bloques que amplían cada vez más la capacidad de Moodle de facilitar la interacción y el intercambio entre docentes y estudiantes, este hecho resulta ser uno de los pilares fundamentales de la plataforma. Gráficamente los bloques son paneles expandibles y suelen tener un nombre relativo a sus funciones. Cada usuarios tiene una vista determinada del bloque con el que interactúe, es decir, para un bloque específico el profesor es quien regula las opciones a las que tienen acceso los estudiantes así como el contenido que visualizan y en qué momento lo podrán hacer (Macías Álvarez, 2010).

A continuación se mencionan junto a una breve descripción algunos de los bloques que ya incluye Moodle (MoodleDocs, 2021):

1.2 Los Entornos Virtuales de Aprendizaje (EVA), herramientas tecnológicas educativas

- **Actividades:** el bloque de Actividades facilita la navegación entre las diferentes actividades disponibles en su curso (por ejemplo, Módulo de foro, Módulo de examen, Módulo de tarea, Módulo de lección) y proporciona enlaces hacia la página índice para cada actividad o recurso. En un primer momento cuando recién entre a su curso, la única categoría que estará listada será Foros porque por defecto siempre existe un foro. La lista de actividades crecerá conforme el usuario añade actividades a su curso.
- **Administración:** el bloque de administración es usado para proporcionar enlaces hacia páginas de configuraciones en dependencia del tema que se use de Moodle. Lo que aparece dentro del bloque depende la ubicación del el usuario y el rol que desempeña en ese entorno. Por ejemplo, un profesor en un curso tendrá acceso a más opciones de administración del curso que un estudiante.
- **Anuncios recientes:** en este bloque se muestran las publicaciones más recientes hechas en el Foro de anuncios junto con un enlace a las noticias anteriores archivadas.
- **Calendario:** el bloque de Calendario muestra eventos de: el sitio (evento visible en todos los cursos - creado por usuarios administradores); la Categoría Curso (evento visible solamente para los miembros del curso - creado por maestros); Grupos (evento visible solamente para los miembros de un grupo - creado por maestros); Usuario (evento personal que puede crear un usuario estudiante - visible solamente por el usuario)
- **Cursos:** el bloque de Cursos permite la navegación entre todos los cursos a los cuales el usuario haya ingresado o esté inscrito.
- **Eventos próximos:** el bloque de eventos próximos muestra los eventos futuros en una lista resumida.

Los EVA a pesar de ser los héroes de la enseñanza a distancia en estos últimos tiempos, debido a las condición de pandemia mundial que impedía el desarrollo de clases presenciales; han recibido críticas respecto a la falta de empatía educativa, carencia de trato especializado, poca motivación por parte de los estudiantes en la interacción con las plataformas y retroalimentación limitada. Un factor importante es impulsar la motivación de los estudiantes, ya sea mediante nuevas técnicas evaluativas o por un intercambio en tiempo real con los profesores. Las motivaciones, generalmente las generan las emociones, en este sentido, resulta de interés conocer las emociones de los estudiantes mientras interactúan con algún material de estudio determinado, o cuando realizan exámenes; en el 2019 (Vázquez Basulto, 2019) implementó un bloque llamado Emociones para la detección de emociones en los estudiantes en la plataforma Moodle. Al conocer la emoción de un estudiante, se pueden tomar varias alternativas para mejorar su experiencia con la plataforma de aprendizaje; una variante es la implementación de un bloque que basado en las emociones, sea capaz de recomendar materiales de estudio que satisfagan y simplifiquen la búsqueda del estudiante. Diversas son las alternativas que se

pueden implementar en las plataformas de aprendizaje virtual, a partir de la detección de emociones debido al importante papel que juegan en el aprendizaje.

1.3 Las emociones y su vinculación en la educación

Las emociones son estados afectivos que los seres humanos experimentamos. Cada individuo experimenta una emoción de forma particular, dependiendo de sus experiencias previas, aprendizaje, carácter y de la situación concreta. Algunas de las reacciones fisiológicas que desencadenan las emociones son innatas, mientras que otras pueden adquirirse. Para los seres humanos las emociones y las reacciones que a partir de estas se derivan, van marcando el desarrollo y los sucesos de la vida. Mediante la emoción que se manifiesta antes las diferentes situaciones de la cotidianidad se va formando la personalidad y el individuo en sí; el intercambio de emociones en una sociedad contribuye a que se maten unas emociones sobre otras en la personalidad. Las emociones definen al hombre pues incluso el cuerpo humano es una evidencia, debido a la gran influencia que tienen las emociones tanto en la personalidad, como en el físico y casi todas las áreas importantes de la vida (Bericat Alastuey, 2012).

1.3.1 Emociones, concepto

Las emociones desempeñan un papel muy importante en la adquisición, procesamiento de información y en el aprendizaje de las personas; en consecuencia, dentro de la inteligencia artificial, en los últimos años se ha visto un crecimiento vinculado a las investigaciones sobre el reconocimiento de las emociones como parte del área de la computación afectiva (Bosquez et al., 2018).

Se entiende por emoción el conjunto de reacciones orgánicas que experimenta un individuo cuando responden a ciertos estímulos externos que le permiten adaptarse a una situación con respecto a una persona, objeto, lugar, entre otros. La emoción se caracteriza por ser una alteración del ánimo de corta duración pero, de mayor intensidad que un sentimiento. En ocasiones son las causantes de diversas reacciones orgánicas que pueden ser de tipo fisiológico, psicológico o conductual, es decir, son reacciones que pueden ser tanto innatas como estar influenciadas por las experiencias o conocimientos previos (James, 1985).

Las emociones no sólo son reacciones a los estímulos del presente, sino que también se producen por el recuerdo o evocación de hechos sucedidos en el pasado o por el vaticinio o anticipo de posibles situaciones futuras. También generan respuestas físicas controladas por el cerebro que permiten sobrevivir a organismos hostiles, por ejemplo el miedo activa el instinto defensivo, la cólera activa el instinto ofensivo y el asco el instinto de rechazo (Guerri, 2021).

1.3.2 Clasificación de las emociones

Existen diversos tipos de emociones que se clasifican en un orden que va desde las más básicas a las emociones aprendidas en diversos contextos. Las básicas o primarias son aquellas que son innatas y que responde a un estímulo; la alegría, tristeza, ira, miedo, sorpresa, asco. Existen emociones secundarias que son aquellas que se generan a raíz de una emoción primaria; por ejemplo orgullo, ansiedad, celos, esperanza. Además, las emociones se agrupan en positivas y negativas; esta clasificación depende de cómo afectan las conductas de las personas, por lo que algunas emociones pueden generar acciones o reacciones positivas como alegría o satisfacción pero, hay otras que provocan sentimientos perjudiciales para el individuo (López Rosetti, 2018).

A continuación de resumen las seis emociones fundamentales según (Vindel, 2000):

- **Alegría:** es una emoción primaria y positiva. Los desencadenantes de la alegría o felicidad son el éxito y el logro.
- **Sorpresa:** generalmente es clasificada como emoción primaria y neutra. Puede ser positiva o negativa según las circunstancias y tiene la función de prepararnos para acontecimientos inesperados, pues ante una sorpresa, sobre todo si es negativa, hay que reaccionar rápidamente. Es causada por algo imprevisto o extraño.
- **Miedo:** emoción primaria y negativa que se produce por un peligro presente e inminente. Desarrolla la función de supervivencia.
- **Asco:** emoción primaria y negativa que se produce debido a la repugnancia que se tiene hacia algo que nos resulta especialmente desagradable. Esta emoción implica una respuesta de rechazo y además tiene funciones adaptativas.
- **Ira:** es una reacción emocional que se produce cuando una el individuo considera que sus intereses se verán injustamente afectados; se desencadena también producto de situaciones consideradas injustas provocando frustración. Sus efectos son enojo, furia, inconformidad.
- **Tristeza:** emoción primaria y negativa; la tristeza es una reacción emocional que surge cuando se pierde algo importante. La intensidad de la tristeza que se siente depende de cómo es cuantificada la pérdida. También se genera producto de la frustración de un deseo inmediato, cuya satisfacción resulta imposible. Se caracteriza por sentimientos de desánimo y melancolía.

Las emociones pueden determinar cuestiones tan importantes como el bienestar personal, la motivación escolar o el rendimiento académico por tanto juegan un papel importante en el proceso educativo en cualquiera de las etapas de enseñanza. Un sencillo ejemplo de la presencia de las emociones es cuando un estudiante está intentando dar solución a un problema o ejercicio, de cualquier rama del conocimiento; enfrenta diversos estados emocionales; desde confusión, frustración, aburrimiento; hasta satisfacción o alegría cuando desarrolla una solución correcta. Respecto a las emociones ne-

gativas, pueden resultar poco propicias para el aprendizaje causando desmotivación e incitando al estudiante a abandonar el proyecto en el que esté; aunque por otra parte pueden impulsar la creatividad en el alumno como respuesta de superación ante la frustración. La reacción del estudiante ante las emociones negativas en cualquier proceso educativo depende de su poder de investigación, reflexión, auto estudio e independencia cognitiva; estas características se van desarrollando y fortaleciendo a lo largo de las diferentes etapas educativas que vaya venciendo el alumno (Escárzaga et al., 2020).

De forma general, las emociones son determinantes en la vida de los seres humanos; específicamente en la educación; a partir de las emociones que surgen ante los retos o dificultades el individuo desarrolla la capacidad para sobreponerse o no. En un entorno de clases presenciales, el profesor tenía la oportunidad de interactuar en tiempo real con el alumno y detectar así cualquier emoción negativa ante una actividad determinada y podía apoyarlo a dar solución a su motivo de frustración o agobio. La educación a distancia a través de EVAs carece de ese intercambio en tiempo real y a pesar de que el profesor pueda guiar al estudiante mediante chats, emails, incluso llamadas de video; en muchas ocasiones el alumno sufre desmotivación y soledad al no encontrar respuesta a sus dudas como lo hacía en el aula. Por tanto es innegable la importancia de las emociones en el proceso educativo virtual; a partir de que el profesor pueda conocer las emociones del estudiante mientras interactúa con la plataforma; puede implementar estrategias que se ajusten a las necesidades individuales de los estudiantes.

1.4 Bloque Emociones, funcionalidades

El bloque “Emociones” implementado por Vázquez Basulto (2019) está dirigido fundamentalmente al profesor para que tenga acceso a las emociones de los estudiantes mientras interactúan con su curso. El bloque arroja resultados innovadores en el plano educativo pues añade a la plataforma Moodle nuevas funcionalidades orientadas al mejoramiento de la retroalimentación e interacción entre el docente y el alumno. Al interactuar el profesor con el bloque puede seleccionar los estudiantes a los cuales quiere solicitarle la detección de emociones; por supuesto, cada estudiante debe aprobar los permisos para que la cámara los reconozca. Además, el profesor puede personalizar la relación entre la cantidad de imágenes y el tiempo en que se capturan. También tendrá acceso a todos los análisis que ha planificado para ese curso y las emociones que presentaron sus estudiantes en cada uno de ellos.

Respecto a la vista de los estudiantes del bloque Emociones, solamente interactúan con una cámara que se muestra en pantalla. Para facilitar el reconocimiento facial, es necesario que el estudiante se presente adecuadamente frente a la cámara, es decir; en una posición correcta para que la cámara tenga el ángulo necesario para registrar el rostro en una foto. La iluminación del lugar de estudio

también influye en la claridad de la imagen por tanto debe ser buena. El análisis puede fallar si se capta un rostro sudoroso o si la imagen queda desenfocada o corrida (Vázquez Basulto, 2019).

El bloque Emociones resulta de notable importancia para mejorar las condiciones del proceso docente educativo a distancia. A partir de que el profesor tiene acceso a las emociones de los estudiantes que seleccione, se puede implementar una atención personalizada que responda a las necesidades educativas individuales de cada estudiante. Mediante el conocimiento que brinda el bloque se pueden generar alternativas que lo utilicen para impulsar y motivar la empatía de los estudiantes hacia el estudio mediante la plataforma.

1.5 Sistemas de recomendación

Los Sistemas de Recomendación (SR) son sistemas inteligentes que proporcionan a los usuarios una serie de sugerencias personalizadas sobre un determinado tipo de elementos. Estas recomendaciones se realizan a partir de la obtención de las características de cada usuario para, mediante un procesamiento de los datos, encontrar un subconjunto de ítems que pueden resultarle de interés (Olguín et al., 2019).

También se puede definir como una herramienta que establece un conjunto de criterios y valoraciones sobre los datos de los usuarios para realizar predicciones sobre recomendaciones de elementos que puedan ser de utilidad o valor para el usuario. Estos sistemas seleccionan datos proporcionados por el usuario de forma directa o indirecta, y procede a analizar y procesar información del historial del usuario para transformar estos datos en conocimiento de recomendación (Pajuelo Holguera, 2021).

Anteriormente los motores de búsqueda, plataformas de contenido y ventas de producto funcionaban con rankings o listas de popularidad. Estos sistemas eran funcionales hasta cierto punto, pero no podían personalizar la experiencia del usuario y mostraban elementos que no se correspondían a nuestros intereses (López, 2013). Actualmente las técnicas de Aprendizaje Automático (AA) o Machine Learning (ML) en inglés, han impulsado los SR considerablemente.

1.5.1 Clasificación

Para entender el funcionamiento de los SR es necesario conocer su composición y sus diferentes clasificaciones. Formalmente, puede definirse a un SR como una función que devuelve una lista de ítems ordenados por su utilidad con respecto a un usuario determinado. En la siguiente figura se ilustra la composición de un SR según (Olguín et al., 2019).

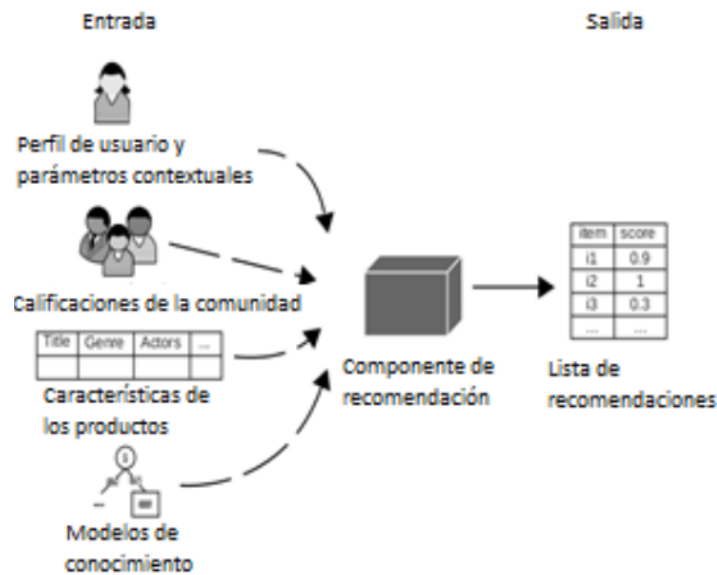


Figura 1.1: Esquema funcional de un SR

Existen varios tipos de SR, estos dependen de sus variables principales que son las que condicionan su funcionamiento; a continuación se mencionan y describen de forma breve según (Olguín et al., 2019):

- **Basados en contenido:** estos sistemas muestran al usuario sugerencias basadas en sus selecciones previas; es decir el proceso de filtrado realiza una comparación con elementos que el usuario ya haya calificado antes para así mostrar recomendaciones lo más cercanas posible a sus gustos. Dichas recomendaciones pueden realizarse usando calificaciones que han recibido los objetos de forma anónima o bien considerando sus características. Se basan en la premisa de que, si una persona escogió un elemento, entonces le gustarán otros similares. Generalmente estos sistemas tienen problemas con los usuarios nuevos, pues requieren saber los intereses del usuario en los objetos del catálogo para poder realizar su trabajo. Ver la figura 1.2 donde se resume en un esquema este SR.
- **Basados en conocimiento:** en este tipo de sistema las recomendaciones se hacen según el historial de navegación de un usuario; es decir, a partir de información que conocen debido a la interacción con el usuario. Su base sigue la idea de satisfacer necesidades en vez de intereses. Estos sistemas requieren de una base de conocimientos que defina los criterios que los ítems deben cumplir para poder ser recomendados. Cuando se tienen ítems que no coinciden con ninguno de los requerimientos establecidos se implementa la variante de relajar algunas restricciones o comenzar a interactuar con el usuario para enriquecer la información que se tiene. Estos sistemas requieren de un discriminante para ponderar las recomendaciones cuando existe

más de un artículo que cumpla con las especificaciones solicitadas; esta característica puede resultar en desventaja respecto a otros SR. Ver la figura 1.3.

- **Basados en colaboración:** este tipo de sistema genera recomendaciones analizando datos, identificando perfiles y haciendo contraste entre la información del perfil del usuario y la del colectivo de usuarios. De esta forma el modelo puede aprender a agrupar perfiles similares y aprender de los datos que recibe de forma general, para desarrollar recomendaciones individuales. Las recomendaciones también se basan en las calificaciones que los usuarios han hecho sobre los ítems, ya sea de forma explícita o implícita. Estos sistemas trabajan sobre una matriz de calificaciones de los objetos y se basan en la idea de que; a un usuario X le han gustado ciertos elementos porque los ha calificado de forma positiva, entonces le gustarán también los elementos que otros han calificado de forma similar. Como desventaja, estos sistemas presentan problemas con los comportamientos de las calificaciones, pues puede ser que no haya ninguna o bien fueron asignadas al azar; igualmente con los nuevos elementos en el catálogo, puesto que no han recibido ninguna calificación. Ver la figura 1.4.
- **Híbridos:** combinan diferentes tipos de SR para generar las recomendaciones; por ejemplo, los basados en conocimiento y contenido, para subsanar las debilidades de cada uno de los métodos por separado, o bien basados en colaboración y conocimiento. Ver la figura 1.5.

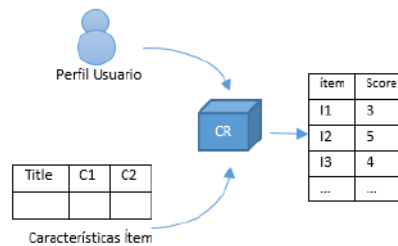


Figura 1.2: Sistema de recomendación basado en Contenido donde CR es el componente de recomendación (Vera Sancho, 2016)

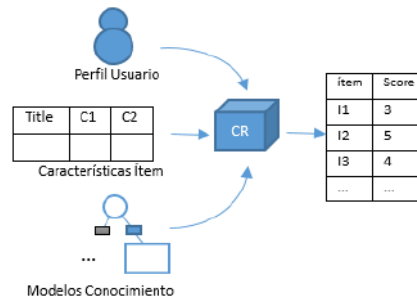


Figura 1.3: Sistema de recomendación basado en conocimiento donde CR es el componente de recomendación(Vera Sancho, 2016)

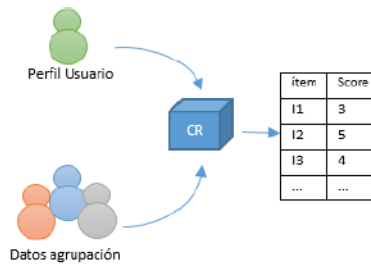


Figura 1.4: Sistema de recomendación basado en filtros colaborativos donde CR es el componente de recomendación(Vera Sancho, 2016)

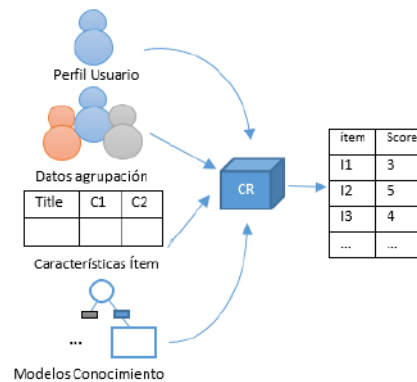


Figura 1.5: Sistema de recomendación híbrido donde CR es el componente de recomendación(Vera Sancho, 2016)

En la siguiente tabla se muestra una comparación entre los tipos de SR antes descritos.

| Características | Basado en contenido | Basados en colaboración | Basado en conocimiento | Híbrido |
|--|---------------------|-------------------------|------------------------|---------|
| Análisis de los atributos del ítem | Si | No | Si | Si |
| Análisis del perfil del usuario | Si | Si | No | Si |
| Análisis de las interacciones del sistema | Si | Si | Si | Si |
| Análisis de las calificaciones de los usuarios sobre los ítems | No | Si | No | Si |
| Problemas de arranque frío | No | Si | Si | No |
| Problemas de escalabilidad | Si | Si | Si | Si |

Tabla 1.1: Diferencias entre los SR (Carrillo, 2013)

Sea cual sea la técnica a seguir, el proceso de obtención de recomendaciones se basa en similitud. La similitud es un valor que indica cuán relacionados están dos componentes. Generalmente se usan las medidas de distancia entre ítems para representar esta medida, por ejemplo: la correlación de Pearson, la distancia cuadrática media, la distancia Manhattan, la distancia del coseno, la distancia del coseno ajustado, etc. Cada una de estas ha sido propuesta por diferentes investigadores como una forma de definir mejores relaciones entre los elementos basados en la información que se posee; aquellos más parecidos serán los que estarán involucrados dentro del cálculo de las recomendaciones (Chuqitarco Chasiluisa, 2018).

Los sistemas de recomendación han logrado cambiar la forma en la que los usuarios consumen nuevos contenidos y descubren productos nuevos. Uno de los ejemplos más claros de SR son las páginas de compra de productos como Amazon, Ebay o AliExpress. Estas plataformas web con pocos datos del usuario es capaz de sugerir ítems que realmente resultan de interés y tanto el usuario como el vendedor agradecen este servicio. De igual forma sucede con las plataformas de contenidos como Youtube, Spotify o Netflix. Sus recomendaciones precisas ayudan a descubrir nuevas series, videos o artistas al analizar nuestros gustos y preferencias.

Sin dudas, los SR han impulsado la satisfacción de las necesidades del cliente de forma eficiente. La experiencia del usuario en las diferentes plataformas se torna en una actividad agradable y sencilla, pues estos sistemas actúan como un asistente personal que estimula al usuario a seguir descubriendo elementos. Las recomendaciones de productos personalizados acerca al cliente a lo que desea y mejorando las posibilidades de que este compre o consuma el contenido sugerido. Otro aspecto importante es que estos sistemas ayudan a obtener datos para informes y reportes estadísticos que pueden alimentar de forma excepcional los modelos de AA e IA .

1.6 Filtrado colaborativo

Uno de los elementos más innovadores y que ayuda a optimizar los procesos de recomendación es el Filtrado Colaborativo (FC). Esto es una técnica que permite solventar los problemas de sobre información que pueden sufrir los usuarios de plataformas tecnológicas. El FC hace predicciones automáticas sobre los intereses de un usuario en particular mediante la recopilación de preferencias o gustos de un mismo consumidor comparados con los datos suministrados por personas con patrones similares. Es decir; las recomendaciones se hacen basándose en el grado de similitud entre usuarios. Se fundamentan en el hecho de que los ítems que le gustan a un usuario, les pueden interesar a otros usuarios con gustos similares (Vera Sancho, 2016).

En los sistemas de FC se distinguen tres características que resumen a grandes rasgos su funcionamiento, según (Herlocker et al., 2000):

- Creación del perfil de usuario (Ratings): las preferencias de los usuarios se pueden guardar de dos formas: implícito o explícito. Las calificaciones implícitas se extraen de los temas de interés de las páginas revisadas por el usuario, es decir, se analiza su historial. Las calificaciones de forma explícita son los ratings numéricos que los usuarios asignan a artículos de interés o las calificaciones que expresa directamente. Cuando las calificaciones que se tienen en cuenta son implícitas cualquier interacción con el sistema, podría potencialmente afectar el resultado de la recomendación. Es útil darle al usuario una retroalimentación sobre la calidad de su perfil, por tanto los ratings dados por él directamente resultan teniendo mayor peso sobre las recomendaciones que aquellos valores calculados implícitamente. Mientras mayor sea el número de ítems calificados por el usuario, mayor es la probabilidad de obtener mejores recomendaciones. Además, el propio usuario puede mejorar su perfil de forma explícita buscando la obtención de resultados con mejor precisión; a partir de que se proporcione una retroalimentación.
- Localización de usuarios con perfiles similares (vecinos): en este sentido, el sistema muestra el valor de las recomendaciones expresadas de forma explícita por los usuarios, pues examina grandes números de candidatos con diferentes preferencias para escoger de entre todo el conjunto los que sean más semejantes al usuario actual. Si los vecinos seleccionados por el sistema son los más semejantes al usuario, de acuerdo a sus necesidades, las recomendaciones resultantes serán también las más precisas.
- Combinación de los ratings de los vecinos para formar recomendaciones. En la mayoría de los casos, la predicción es simplemente un promedio ponderado de los ratings de usuarios.

1.6.1 Clasificación

Para la realización de un sistema de recomendación de FC que ofrezca alta precisión en las recomendaciones, es necesario utilizar un buen algoritmo de FC; el objetivo de estos algoritmos es sugerir nuevos ítems o predecir la utilidad de cierto ítem para un usuario particular basándose en las elecciones de otros usuarios similares. Estos algoritmos se clasifican en (Nieto, 2010) :

- Basados en memoria: estos utilizan valoraciones que otros usuarios han dado a un ítem determinado, para calcular la posible valoración para el usuario actual. Estos algoritmos utilizan toda la base de datos de elementos y usuarios para generar predicciones. Primeramente emplean técnicas estadísticas para encontrar a vecinos, es decir usuarios con un historial de valoraciones sobre los elementos similar al usuario actual. Una vez que se ha construido una lista de vecinos se combinan sus preferencias para generar una lista con los N elementos mas recomendables para el usuario actual. Entre sus inconvenientes se encuentra la necesidad de disponer de un número mínimo de usuarios con un número mínimo de predicciones cada uno, incluido el usuario para el que se pretende realizar la recomendación.
- Basados en modelos: los basados en modelos desarrollan primero un modelo de los ratings del usuario. Luego tratan el problema como un problema de predicción estadística y calculan el valor esperado para cada ítem en función de los ratings que se tienen previamente guardados. Para ello se utilizan distintos algoritmos de aprendizaje, clustering o redes neuronales. Por ejemplo utilizando clustering se trata de clasificar a un usuario en particular dentro de una clase de usuarios y a partir de ahí se estiman las probabilidades condicionadas de esa clase hacia los elementos a evaluar. En general, ante las consultas responden más rápido que los basados en memoria, pero por necesitan de un proceso de aprendizaje intensivo

Los SR de FC tienen como ventaja que un usuario puede recibir recomendaciones de ítems que no estaban siendo buscados de forma activa, la desventaja es el problema del primer evaluador o inicio en frío, cuando un nuevo objeto es agregado no existe manera de recomendarlo por este sistema. En la siguiente sección se describen algunas de las desventajas características de los sistemas de FC.

1.6.2 Ventajas fundamentales de los sistemas de FC

El FC para el filtrado de información tiene ventajas adicionales sobre los sistemas basados en contenido, (Monroy, 2003):

- Brinda soporte para el filtrado de ítems cuyo contenido no se analiza fácilmente de forma automática. Es decir, puede ejecutarse en ítems que resultan difíciles de analizar como películas, sentimientos, ideas, personas y políticas debido a que el usuario puede calificarlos en cuanto a su relevancia, el interés que tengan o la calidad.
- Cuenta con la habilidad para filtrar ítems basados en la calidad y gustos, midiendo con cuan-

ta efectividad el ítem satisface las necesidades o intereses del usuario, más allá del simple contenido. En el caso de la búsqueda basada en contenido devolvería como sugerencias a un usuarios todos los ítems relacionados mediante una característica específica; sin embargo; el FC arrojaría solamente resultados relevantes.

- Tiene la capacidad de proveer recomendaciones inesperadas, de ítems que resultan valiosos para el usuario pero que no contienen la información que el usuario había solicitado en un primer momento. Esto ocurre debido a que se analizan los ítems que fueron recomendados a usuarios similares.

Cabe destacar que aunque el potencial del FC para mejorar las herramientas de filtrado de información es muy grande; ciertamente la vinculación del FC con el filtrado por contenido puede arrojar resultados más eficientes para localizar contenidos específicos (López, 2013).

1.6.3 Problemas de los SR con FC

Los sistemas de FC a pesar de sus ventajas y aspectos positivos en cuanto a la precisión de las recomendaciones también presentan puntos negativos que vale mencionar. A continuación se resume de forma breve algunas desventajas de los sistemas con FC (B.Thorat et al., 2015):

- El problema del estado inicial (Early rater problem): el FC no puede hacer predicciones para un ítem que acaba de entrar al sistema, pues no ha sido calificado aun por ningún usuario. Además, las predicciones que se hacen sobre nuevos ítems (con muy pocas calificaciones) resultan inexactas. De forma similar, las predicciones para nuevos usuarios también resultan pobres. Este problema es conocido como “Cold Start problem” o problema del arranque en frío.
- Problema de los datos dispersos (Sparsity problem): en muchos dominios de información, el número de ítems excede en gran medida a la cantidad que un individuo puede asimilar, por lo tanto, las matrices que contienen todos los ratings de usuarios para todos los ítems, resultan quedando muy dispersas. Este hecho dificulta encontrar ítems que hayan sido calificados por suficientes personas para realizar las predicciones colaborativas.
- El problema de la oveja negra (Gray sheep): en una comunidad de usuarios pequeña o mediana, existen usuarios que no llegan a beneficiarse de un sistema de FC puro, pues sus opiniones no coinciden con las de la mayoría del grupo y por tanto no se encontrarán usuarios similares.
- Confianza entre usuarios: en las recomendaciones del tipo colaborativo, se puede dar el caso de que existan usuarios mal intencionados que intentan corromper el sistema de recomendación. Ante este hecho se propone la creación de una “red de confianza”. La idea sería que un usuario, además de puntuar los ítems del sistema, también exprese su confianza en las valoraciones que han hecho otros usuarios. El enfoque propuesto permite basar las recomendaciones únicamente

con las valoraciones de usuarios en los que se confía. La red de confianza es asimétrica, es decir, si un usuario A confía en las valoraciones de un usuario B, no implica que el usuario B confíe en las valoraciones del usuario A. El sistema permite, mediante los valores bajos de confianza, una rápida detección de usuarios malintencionados.

El filtrado basado en contenido no presenta estos inconvenientes, debido a que recomienda ítems basados en el perfil del usuario, construido analizando el contenido de los ítems que este ha calificado. Sin embargo, también puede ser poco efectivo, porque tiene la dificultad para distinguir entre información de baja y alta calidad, que pertenezcan al mismo tema, además, el número de ítems de cada tópico es cada vez mayor, decrementando la efectividad de estos sistemas.

1.7 Conclusiones del capítulo

Los SR constituyen una de las herramienta que más agradecen los usuarios en la actualidad, en el momento de interactuar con plataformas de compra o selección de contenido. Estos sistemas resultan útiles tanto para los usuarios como para los vendedores o la parte que presta el servicio web; pues simplifica el trabajo de búsqueda de los usuarios y a su vez potencia los resultados de los proveedores de servicios.

En las plataformas de enseñanza en línea o e-learning en inglés, resulta fundamental el apoyo de los SR mediante FC. Actualmente debido a la popularidad de las plataformas virtuales de aprendizaje, muchos usuarios se enfrentan a un proceso de adaptación ante los nuevos métodos de estudio. Son conocidas muchas de las carencias de los EVA; en su mayoría, relacionadas con la falta de motivación por parte de los estudiantes ante el estudio en solitario. Partiendo del hecho anterior, se han tenido que desarrollar estrategias para fomentar la interacción y la retroalimentación entre estudiantes y profesores en las plataformas virtuales.

Las emociones desempeñan un papel fundamental en el proceso educativo; en las clases presenciales el profesor atendía a los estudiantes de forma especializada para dar respuesta a sus dudas e insatisfacciones; en entornos virtuales, el profesor asesora al estudiante, pero no puede identificar en tiempo real su reacción ante las diferentes actividades educativas.

En la UCLV (Vázquez Basulto, 2019) se implementó un bloque para la plataforma Moodle llamado Emociones; orientado fundamentalmente a que el profesor pueda identificar las emociones de los estudiantes mientras interactúan con materiales de estudio o actividades de autoevaluación. Inicialmente el estudiante debe aceptar la solicitud de permisos para que la cámara del dispositivo del cual accede a Moodle lo pueda identificar, luego el profesor puede conocer la reacción y el estado de ánimo de sus estudiantes. Con esta alternativa, el profesor puede elaborar estrategias de atención especializada que respondan a las necesidades educativas de los estudiantes.

Utilizar las emociones registradas por el bloque Emociones para determinar la aceptación de un

material de estudio por un estudiante resulta de interés; de esta forma se puede calificar los materiales de estudio que el profesor propone a los estudiantes y elaborar un sistema de recomendación que muestre al estudiante recursos educativos que le han resultado interesantes a otros estudiantes con perfiles similares a él. Este procedimiento podría implementarse mediante la creación de un bloque para Moodle que facilite el autoestudio del alumno sugiriéndole materiales que realmente le resulten de interés y sean útiles.

2. PROPUESTA DE DISEÑO PARA UN RECURSO DE RECOMENDACIÓN DE MATERIALES EDUCATIVOS EN LA PLATAFORMA MOODLE

En las primeras secciones del capítulo se describen y comparan algunas de las herramientas con soporte para la implementación de Sistemas de Recomendación (SR) basados en Filtrado Colaborativo (FC). A partir de la sección 2.4 se comienza a trabajar sobre una propuesta de diseño de un bloque de recomendaciones en la plataforma Moodle. Se describe el diseño y se formaliza la propuesta mediante el modelado UML. Además, se explican las ventajas y lo innovador que resulta la idea de contar con un bloque para recomendaciones de materiales educativos a los estudiantes, que se complementa con el bloque Emociones mencionado en el capítulo anterior.

2.1 Diferentes marcos de trabajo para el Aprendizaje Automático, los Sistemas de Recomendación y el Filtrado Colaborativo

Diversos son los entornos que existen para el desarrollo del Aprendizaje Automático (AA) o según su nombre en inglés Machine Learning (ML). Cada herramienta brinda bibliotecas que facilitan la implementación de SR sobre la base del FC. A continuación se describen las características fundamentales de algunos entornos que ofrecen bibliotecas para el AA y el manejo de grandes volúmenes de información.

2.1.1 Mahout

Apache Mahout es un entorno de código abierto que proporciona algoritmos de AA aplicables a entradas de gran tamaño. En sus comienzos, el proyecto Mahout proporcionó implementaciones de algoritmos de aprendizaje automático ejecutables en la parte superior del marco Apache Hadoop, pero la comparación del rendimiento de los algoritmos de Mahout en Hadoop con otras bibliotecas de AA, demostró que Hadoop dedica la mayor parte del tiempo de procesamiento a cargar el estado desde el sistema de archivos en cada paso intermedio.

Por lo antes explicado, la última versión de Mahout va más allá de Hadoop y proporciona varios algoritmos de AA para FC, clasificación y agrupación en clústeres, implementados no solo en Hadoop-MapReduce, sino también en Spark y H2O (Zomaya y Sakr, 2017). Además, introduce un nuevo

2.1 Diferentes marcos de trabajo para el Aprendizaje Automático, los Sistemas de Recomendación y el Filtrado Colaborativo

entorno matemático, llamado Samsara para apoyar a los usuarios proporcionando operaciones de álgebra lineal general y estadísticas (Anil et al., 2020).

2.1.2 BigML

BigML es un sistema que se centra en proporcionar software como servicio (SaaS) para descubrir modelos predictivos a partir de datos, utilizando algoritmos de regresión y clasificación de datos. La característica distintiva de BigML es que los modelos predictivos se presentan a los usuarios como árboles de decisiones interactivos. Los árboles de decisión se pueden visualizar y explorar de forma dinámica dentro de la interfaz que brinda BigML.

Recientemente, BigML lanzó su solución PaaS, llamada *BigML PredictServer*, una imagen de máquina dedicada, que se puede desplegar en Amazon AWS ([Amazon Web Services](#)) (Zomaya y Sakr, 2017).

El eslogan de BigML es Machine Learning made beautifully simple for everyone, siguiendo la idea de que lo importante es que puede usarse por usuarios no expertos en inteligencia artificial (IA) y sin necesidad de programar. BigML tiene una sencilla interfaz gráfica web y cuenta con redes neuronales pre-entrenadas listas para usar, además de más de 100.000 clientes, desde grandes empresas como Rabobank, Pfizer, Cintra o Seagate, a universidades, estudiantes, particulares entre otros (Guillén, 2020).

2.1.3 H2O

H2O es una plataforma de AA de código abierto y rápida. Está respaldada por la compañía H2O.ai (anteriormente llamada Oxdia), que fue cofundada en 2011 por SriSatish Ambati y Cliff Click, el desarrollador líder de JVM (Bonaci y Zecevic, 2016).

H2O a pesar de contar con capacidades básicas de transformación de datos (análisis rápido, transformaciones en columnas y uniones), no es un motor informático general como Spark. Utilizándolos juntos, se puede obtener lo mejor de ambos mundos. Con Spark, se puede leer y unir datos de varias fuentes, analizar los datos, transformarlos, transferirlos a H2O y luego crear y usar los modelos de AA H2O. De esta manera, H2O y Spark pueden complementarse entre sí (H2O.ai, 2021).

Muchos de los algoritmos que brinda H2O se superponen con los algoritmos de AA de Spark, pero H2O tiene algunos que Spark todavía no tiene (aprendizaje profundo y modelado generalizado de rango bajo). A su vez, Spark tiene varios algoritmos que no están disponibles en H2O, como mínimos cuadrados alternos (ALS), asignación de Dirichlet latente, SVM, tf-idf, Word2vec y otros (Bonaci y Zecevic, 2016).

2.1 Diferentes marcos de trabajo para el Aprendizaje Automático, los Sistemas de Recomendación y el Filtrado Colaborativo

2.1.4 Spark

Apache Spark es otro entorno de Apache para el procesamiento de Big Data. A diferencia de Hadoop, en el que los datos intermedios siempre se almacenan en sistemas de archivos distribuidos, Spark almacena datos en la memoria RAM y los consulta repetidamente para obtener un mejor rendimiento para algunas clases de aplicaciones (por ejemplo, algoritmos iterativos de AA). Durante muchos años, Hadoop ha sido considerado el entorno de código abierto líder de Big Data, pero recientemente Spark ha ganado mucha popularidad y es compatible con los principales proveedores de Hadoop. Para algunas tareas específicas, Spark es hasta 100 veces más rápido que Hadoop en memoria y 10 veces más rápido en disco (Zomaya y Sakr, 2017).

Diversas bibliotecas se han creado sobre Spark, por ejemplo, Spark SQL para tratar con SQL y DataFrames, MLlib para AA, GraphX para grafos y cómputo paralelo de grafos y Spark Streaming para crear aplicaciones de transmisión escalables tolerantes a fallas. Debido a todas las áreas de trabajo que facilita Spark mediante sus diferentes bibliotecas, se está convirtiendo en el motor de ejecución principal para el procesamiento de datos y resulta una herramienta casi imprescindible para las aplicaciones de Big Data (Aziz et al., 2019).

En muchas aplicaciones Spark puede considerarse una mejor alternativa que Hadoop, en otras tiene limitaciones que los hacen complementarios. La principal limitación de Spark es que no proporciona su propio sistema de almacenamiento distribuido y escalable, requisito fundamental para las aplicaciones de Big Data que utilizan un volumen enorme y en constante aumento de datos almacenados en una gran cantidad de nodos. Para dar solución a la falta antes mencionada, Spark ha sido diseñado para ejecutarse sobre varias fuentes de datos, como almacenamiento de objetos en la nube (por ejemplo, Amazon S3 Storage, Swift Object Storage), sistema de archivos distribuido (por ejemplo, HDFS), bases de datos no SQL (por ejemplo, HBase, Apache Cassandra) y otros. Actualmente, un creciente número de grandes proveedores, como Microsoft Azure o Cloudera, ofrecen tanto Spark como Hadoop, por tanto, los desarrolladores pueden seleccionar el marco más adecuado para cada aplicación de análisis de datos. Spark, respecto a Hadoop, carga datos de fuentes de datos y ejecuta la mayoría de sus tareas en la memoria RAM, de esta forma, reduce significativamente el tiempo dedicado a escribir y leer desde discos duros, por lo que la ejecución es mucho más rápida que usando Hadoop (Aljunid y Manjaiah, 2019).

Respecto a la recuperación de tareas en caso de fallas, Hadoop devuelve todos los datos al almacenamiento después de cada operación. De forma similar, Spark permite la recuperación pero mediante la organización de los datos en conjuntos de datos distribuidos resistentes (Resilient Distributed Dataset (RDD)), que son colecciones de registros inmutables y tolerantes a fallas que se pueden almacenar en la memoria volátil o en un almacenamiento persistente (por ejemplo, HDFS, HBase) (Penchikala, 2018).

2.2 Comparación entre los sistemas de análisis de Big Data y AA descritos

Los analistas de Big Data utilizan cada vez más la capacidad de procesamiento en tiempo real de Spark en aplicaciones que requieren extraer información rápidamente de los datos, como los SR y monitoreo. Grandes empresas y organizaciones utilizan Spark para fines de análisis de Big Data: por ejemplo, Ebay utiliza Spark para la agregación y análisis de transacciones de registro, Kelkoo para recomendaciones de productos, SK Telecom analiza los patrones de uso móvil de los clientes entre otros ejemplos (Salloum et al., 2016).

2.2 Comparación entre los sistemas de análisis de Big Data y AA descritos

Establecer una comparación entre los marcos de trabajo o sistemas de análisis de Big Data, descritos previamente, es una tarea que resulta no ser exacta o sencilla, debido a las diferencias, ventajas y desventajas que caracterizan a cada herramienta. Algunos de los sistemas tienen características comunes y usar uno en lugar de otro es una selección difícil, por ejemplo, dada una tarea analítica específica, como una aplicación de AA, es posible utilizar varias herramientas. Algunas son herramientas comerciales ampliamente utilizadas, orientadas fundamentalmente a simplificar y agilizar el trabajo, al poder ser utilizadas por usuarios sin experiencia (por ejemplo, AMS). Otras herramientas son marcos de código abierto que requieren usuarios más capacitados que programan su aplicación utilizando un enfoque más técnico.

La elección de la mejor solución para desarrollar una aplicación de análisis de datos puede depender de muchos factores, como el presupuesto, por ejemplo, en muchos casos los servicios de alto nivel son fáciles de usar pero más costosos que las soluciones de bajo nivel. También influye el formato de los datos, su fuente, la cantidad que se analizarán y su velocidad, entre otros. En la siguiente tabla se presenta una breve comparación de los sistemas de análisis de Big Data descritos en la sección previa.

| Herramientas | Streaming | Grafos | En-memoria | ML | SQL | Procesamiento de datos | Open-Source | Cloud model |
|---------------|-----------|--------|------------|----|-----|------------------------|-------------|-------------|
| Mahout | | | x | x | | | x | IaaS |
| BigML | | | | x | | | | SaaS, PaaS |
| H2O | | | x | x | | | x | IaaS, PaaS |
| Spark | x | x | x | x | x | x | x | IaaS |

Tabla 2.1: Comparación breve de algunas herramientas para el trabajo con BigData y ML según (Zomaya y Sakr, 2017) y (H2O.ai, 2021)

2.2.1 Mahout vs. Spark MLlib

Tanto Apache Spark como H2O son herramientas que procesan los datos en la memoria logrando así un aumento del rendimiento en comparación con el entorno Hadoop en aplicaciones como trabajos

interactivos, consultas en tiempo real y transmisión de datos.

Mahout es un entorno maduro y tiene muchas potencialidades, por ejemplo, brinda varios algoritmos de ML y está construido sobre la base de MapReduce. Esta última característica implica una limitación por accesos al disco y dado que los algoritmos de AA generalmente usan muchas iteraciones, Mahout se ejecuta con lentitud respecto a otras herramientas con algoritmos similares y no maneja correctamente los trabajos iterativos (Anil et al., 2020). Por el contrario, MLlib está construido sobre Spark, lo que lo hace mucho más rápido en este sentido, pues Spark almacena los datos en la memoria RAM y los consulta repetidamente, obteniendo así un mejor rendimiento en algoritmos de AA. Spark también ejecuta algoritmos iterativos mucho más rápido que Hadoop, ya que intenta evitar escribir en el disco tanto como sea posible y almacena los datos en la memoria RAM como se explicó en el párrafo anterior. Por tanto, como MLlib se creó sobre Spark para aprovechar su eficiencia al ejecutar algoritmos iterativos de AA, sus algoritmos terminan siendo mucho más rápidos que los equivalentes de Mahout. Además, MLlib es una de varias bibliotecas creadas sobre Spark, existen también, Streaming, SparkSQL y GraphX, esta diversidad facilita el trabajo del desarrollador teniendo un entorno más completo a su disposición. También cabe destacar que la biblioteca MLlib se ha caracterizado por su rápido crecimiento. (Mohanty, 2018).

Spark MLlib viene en 2 paquetes: Mllib y ML, la primera contiene la API original construida sobre RDD y la segunda proporciona una API de nivel superior construida sobre DataFrames para la construcción de pipelines. La biblioteca ML brinda un trabajo con mayor rapidez que MLlib (Karau et al., 2015).

La biblioteca Spark ML, utiliza el formato DataFrame, diferente a los RDD de MLlib, sin embargo, no soporta todos los tipos de algoritmos de AA. Por tanto, aunque Spark ML puede ser más rápida que MLlib, esta última casi siempre termina siendo necesaria o al menos complementaria para el desarrollo eficiente y completo de aplicaciones de ML (Assefi et al., 2017).

En 2014 según (Yabuz y Meng, 2014), se realizó un experimento para comparar implementaciones de ALS en Spark MLlib a gran escala. La evaluación comparativa se realizó en Amazon Elastic Compute Cloud (Amazon EC2), un servicio web que proporciona capacidad informática en la nube segura y de tamaño modificable, utilizando instancias configuradas por el script Spark EC2. También se creó un scrip en Mahout desde GitHub. Los scrips se pueden reproducir en EC2 y se encuentran en <https://github.com/databricks/als-benchmark-scripts>.

Se ejecutaron 5 iteraciones de ALS en copias escaladas del conjunto de datos de Amazon Reviews, que contiene 35 millones de calificaciones recopiladas de 6,6 millones de usuarios en 2,4 millones de productos. Para cada usuario, se creó pseudo-usuarios que tienen las mismas calificaciones. Es decir, para cada calificación como (userId, productId, rating), se generó (userId + i, productId, rating) donde $0 \leq i < s$ y s es el factor de escala.

2.2 Comparación entre los sistemas de análisis de Big Data y AA descritos

A continuación se muestra el gráfico con los resultados arrojados.

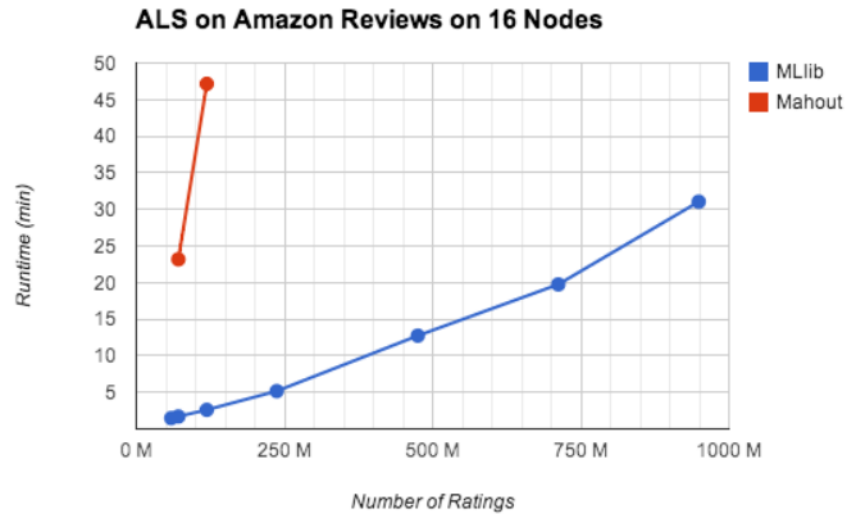


Figura 2.1: Comparación de la eficiencia de ALS implementado en Spark y en Mahout según experimento realizado por (Yabuz y Meng, 2014)

Respecto a Mahout al ejecutarse en Hadoop MapReduce, ralentiza ALS debido a la sobrecarga de programación y la falta de soporte para el cálculo iterativo, característica explicada en subsecciones anteriores.

Spark MLlib demostró un excelente rendimiento y escalabilidad, como se muestra en el gráfico anterior. MLlib también puede escalar a conjuntos de datos mucho más grandes y a una mayor cantidad de nodos, gracias a su diseño de tolerancia a fallas. Con 50 nodos, se ejecutaron 10 iteraciones de ALS de MLlib en 100 copias del conjunto de datos de Amazon Reviews en solo 40 minutos. Spark MLlib reduce el tiempo de entrenamiento del modelo y el costo de ALS (Yabuz y Meng, 2014).

2.2.2 Spark MLlib vs. H2O

Un entorno bastante eficiente y moderno es H2O. Creado por H2O.ai, una empresa emergente de Silicon Valley. H2O forma a su vez un nuevo entorno, Sparkling Water, equivalente a H2O más Apache Spark. Según (Mohanty, 2018), Sparkling Water resulta ser de las herramientas más eficientes para el trabajo con ML de nueva generación y ofrece las mejores oportunidades de AA de código abierto de alto rendimiento en el ecosistema de Hadoop.

Por otra parte, H2O es una biblioteca independiente gratuita, como antes se mencionó, se puede integrar con Spark siendo Sparkling Water el conector. En términos de velocidad de convergencia, H2O afirma tener un mejor rendimiento según la página oficial de H2O (H2O.ai, 2021), que Spark MLlib, pero se concentra únicamente en algoritmos ML, mientras que Spark es un entorno más

completo.

2.2.3 Conclusiones parciales de la sección

Con la revisión de varios marcos de trabajo para el desarrollo de aplicaciones de ML se analizaron sus principales características y funcionalidades. La selección de una herramienta por encima de otra, se basa no solo en un estudio de sus potencialidades y desventajas, sino en cuánto se ajusta a las particularidades de la aplicación que se quiere implementar.

En el caso objeto de estudio, se necesita un entorno que brinde completitud, rapidez en el trabajo con algoritmos de AA y una biblioteca de ML enriquecida con métodos para la implementación de SR mediante FC.

Tras analizar las potenciales herramientas a utilizar, se seleccionó Apache Spark. Este entorno brinda la biblioteca MLlib para el AA y garantiza buena velocidad de procesamiento en algoritmos iterativos debido a la característica de Spark de evitar las consultas en disco realizándolas a la memoria RAM, como se explicó en subsecciones anteriores. Además, Spark al tener varias bibliotecas que abarcan diferentes áreas del desarrollo de software, ofrece un entorno de trabajo completo en una misma herramienta. También se tuvo en cuenta, las facilidades de integración que brinda Apache Spark con otros marcos de trabajo que ofrecen diferentes funcionalidades como H2O por ejemplo. Spark es una herramienta que reúne eficiencia, robustez y diversidad en cuanto a las bibliotecas que brinda.

2.3 Spark para el desarrollo de SR y FC

En esta sección se describe la herramienta Apache Spark seleccionada para el desarrollo de un sistema de recomendación mediante FC.

2.3.1 Spark, características

Spark es un entorno de código abierto y se encuentra gestionado por la [Apache Software Foundation](#). Por tanto, la herramienta se conoce como Apache Spark y es uno de sus proyectos más activos (Sierra, 2019).

Apache Spark es una plataforma de computación en clúster caracterizada por su rapidez y diversas funcionalidades. Respecto a la velocidad, Spark amplía el popular modelo MapReduce para admitir de manera eficiente más tipos de cálculos, incluidas las consultas interactivas y el procesamiento de flujos. La velocidad es un factor importante en el procesamiento de grandes conjuntos de datos pues es la que marca la diferencia entre explorar datos de forma interactiva y esperar minutos u horas. Una de las principales características que ofrece Spark para la velocidad es la capacidad de ejecutar cálculos en la memoria, pero el sistema es más eficiente que MapReduce incluso para aplicaciones complejas que se ejecuten en disco (Omar y Jumaa, 2019).

Desde el punto de vista general, Spark está diseñado para cubrir una amplia gama de cargas de traba-

jo que antes requerían sistemas distribuidos separados, incluidas aplicaciones por lotes, algoritmos iterativos, consultas interactivas y transmisión. Al admitir estas cargas de trabajo en el mismo motor, Spark hace que sea fácil y económico combinar diferentes tipos de procesamiento, lo que frecuentemente es necesario en las canalizaciones de análisis de datos de producción. Además, reduce la carga administrativa de mantener herramientas separadas (Karau y Warren, 2017).

Spark está diseñado para ser altamente accesible, ofreciendo API simples en Python, Java, Scala, SQL y enriquecidas bibliotecas integradas. También se integra fácilmente con otras herramientas de Big Data, por ejemplo como se mencionó en la sección anterior, con H2O. Además, Spark puede ejecutarse en clústeres de Hadoop y acceder a cualquier fuente de datos de Hadoop, incluida Cassandra. Destaca su potencia de procesamiento la que agiliza la detección de patrones en los datos, la clasificación organizada de la información, la ejecución de cómputo intensivo sobre los datos y el procesamiento paralelo en clústers en implementaciones de Big Data y ML.

También, esta herramienta cuenta con la comunidad de código abierto más grande que existe a nivel mundial en cuanto a Big Data, con más de mil colaboradores pertenecientes a más de 250 organizaciones que valoran lo que es Spark y su impacto en el Big Data (Karau et al., 2015).

2.3.2 Componentes de Spark

Existen 4 componentes que integran y potencian Apache Spark, ellos son según (Sierra, 2019):

- Spark SQL: permite acceder a los datos de manera estructurada. También facilita la integración de Spark con Hive, ODBC, JDBC y herramientas de inteligencia de negocios (business intelligence).
- Spark Streaming: brinda soporte para el procesamiento de datos en tiempo real mediante un sistema de empaquetamiento de pequeños lotes.
- MLlib (Machine Learning Library): ofrece una biblioteca de potentes algoritmos de AA.
- GraphX: proporciona una API de procesamiento de grafos para la computación paralela de los mismos.

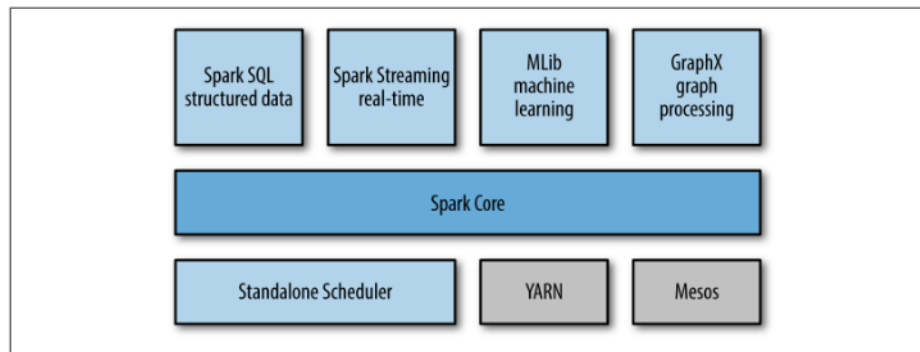


Figura 2.2: Estructura de Apache Spark (Karau et al., 2015)

2.3.2.1. Spark Core

Spark Core contiene la funcionalidad básica de Spark, incluidos componentes para la programación de tareas, administración de memoria, recuperación de fallas, interacción con sistemas de almacenamiento y otros. Spark Core también alberga la API que define conjuntos de datos distribuidos resistentes (RDD), que son la principal abstracción de programación de Spark. Los RDD representan una colección de elementos distribuidos en muchos nodos que permiten su manipulación en paralelo. Spark Core proporciona varias API para crear y manipular estas colecciones (Karau y Warren, 2017).

2.3.2.2. Spark SQL

Spark SQL es el paquete de Spark para trabajar con datos estructurados. Permite consultar datos a través de consultas SQL, así como la variante de SQL de Apache Hive, denominada Hive Query Language (HQL), y admite muchas fuentes de datos, incluidas tablas de Hive, Parquet y JSON (Karau et al., 2015).

Más allá de proporcionar una interfaz SQL para Spark, Spark SQL permite a los desarrolladores mezclar consultas SQL con manipulaciones de datos compatibles con RDD en Python, Java y Scala, todo dentro de una sola aplicación, combinando así SQL con análisis complejos. Esta estrecha integración con el entorno proporcionado por Spark hace que Spark SQL sea diferente a cualquier otra herramienta de almacenamiento de datos de código abierto. Spark SQL se agregó a Spark en la versión 1.0. Shark era un proyecto SQL-on-Spark más antiguo de la Universidad de California, Berkeley, que modificó Apache Hive para que se ejecutara en Spark. Actualmente se ha reemplazado por Spark SQL para proporcionar una mejor integración con el motor Spark y las API de lenguaje (Schiavio et al., 2020).

2.3.2.3. Spark Streaming

Spark Streaming es un componente de Spark que permite el procesamiento de transmisiones de datos en vivo. Los ejemplos de flujos de datos incluyen archivos de registro generados por servidores web de producción o colas de mensajes que contienen actualizaciones de estado publicadas por los usuarios de un servicio web (Karau y Warren, 2017).

Spark Streaming proporciona una API para manipular flujos de datos que se asemeja mucho a la API RDD de Spark Core, lo que facilita a los programadores aprender el proyecto y moverse entre aplicaciones que manipulan datos almacenados en la memoria, en el disco o que llegan en tiempo real. Spark Streaming fue diseñado para proporcionar el mismo grado de tolerancia a fallas, rendimiento y escalabilidad que Spark Core (Nabi, 2016).

2.3.2.4. MLlib

Spark viene con una biblioteca que contiene la funcionalidad del AA, llamada MLlib. MLlib proporciona varios tipos de algoritmos de AA, que incluyen clasificación, regresión, agrupación y FC, así como funciones de apoyo como la evaluación de modelos y la importación de datos. También proporciona algunas primitivas de ML de nivel inferior, incluido un algoritmo genérico de optimización del descenso de gradientes. Todos estos métodos están diseñados para escalar horizontalmente en un clúster (Karau y Warren, 2017).

2.3.2.5. GraphX

GraphX es una biblioteca para manipular grafos (por ejemplo, el grafo de amigos de una red social) y realizar cálculos de grafos paralelos. Al igual que Spark Streaming y Spark SQL, GraphX amplía la API Spark RDD, lo que nos permite crear un grafo dirigido con propiedades arbitrarias adjuntas a cada vértice y borde. GraphX también proporciona varios operadores para manipular grafos (por ejemplo, *subgraph* y *mapVertices*) y una biblioteca de algoritmos de grafos comunes (por ejemplo, PageRank y conteo de triángulos) (Salloum et al., 2016).

2.3.2.6. Cluster Managers

Spark está diseñado para escalar eficientemente de uno a miles de nodos de cómputo, para lograr esto mientras se maximiza la flexibilidad, Spark puede ejecutar una variedad de administradores de clústeres, incluidos Hadoop YARN, Apache Mesos y un administrador de clúster simple incluido en el propio Spark llamado Programador Independiente. Si solo está instalando Spark en un conjunto vacío de máquinas, el Programador Independiente proporciona una manera fácil de comenzar, sin embargo, si ya tiene un clúster Hadoop YARN o Mesos, la compatibilidad de Spark con estos administradores de clústeres permite que sus aplicaciones también se ejecuten en ellos (Aziz et al., 2019).

2.4 Ventajas de utilizar Apache Spark para el Big Data y el AA

Una de las mayores ventajas de Apache Spark es que su velocidad de procesamiento es 100 veces mayor que Apache Hadoop si se ejecuta en memoria y 10 veces mayor si se ejecuta en disco (Sierra, 2019).

También ofrece otras ventajas de relevante importancia, a continuación se mencionan algunas:

- Permite su adaptación a distintas necesidades gracias a que es open source.
- Simplifica el proceso de desarrollo de soluciones inteligentes.
- Mejora el desempeño de aplicaciones dependientes de datos.
- Unifica algoritmos para que trabajen conjuntamente en diversas tareas.
- Integra dentro de sí el modelado analítico de datos.
- Otorga escalabilidad en su potencia al introducir más procesadores en el sistema.
- Reduce los costes al poder utilizarse en hardware estándar de uso común.
- Promueve workflows basados en Grafos Acíclicos Dirigidos que aceleran el procesamiento.
- Dispone de una API para Java, Python y Scala, así como APIs para transformar y manipular datos semiestructurados.
- Facilita la integración con sistemas de archivos como HDFS de Hadoop, Cassandra, HBase, MongoDB y el S3 de AWS.
- Ofrece bibliotecas de alto nivel para mejorar la productividad de los desarrolladores.
- Posee tolerancia a fallos implícita.
- Combina SQL, streaming y análisis de gran complejidad.

2.5 MLlib, la biblioteca de Spark para el AA

MLlib es la biblioteca de AA de Spark. Su objetivo es hacer que el AA práctico sea escalable y fácil según el sitio oficial de Spark (Spark, 2021). Además, contiene variedad de algoritmos de aprendizaje y es accesible desde todos los lenguajes de programación de Spark.

A alto nivel, proporciona herramientas como:

- Algoritmos ML: algoritmos de aprendizaje comunes como clasificación, regresión, agrupamiento y FC.
- Caracterización: extracción, transformación, reducción de dimensionalidad y selección de características.
- Pipelines: herramientas para construir, evaluar y ajustar ML Pipelines.
- Persistencia: guardar y cargar algoritmos, modelos y canalizaciones.
- Utilidades: álgebra lineal, estadística, manejo de datos, entre otros.

Aspecto importante a tener en cuenta sobre MLlib es que contiene solo algoritmos paralelos que se ejecutan bien en clústeres. Algunos algoritmos ML clásicos no están incluidos porque no fueron diseñados para plataformas paralelas, pero en contraste MLlib contiene varios algoritmos de investigaciones recientes para clústeres, como bosques aleatorios distribuidos, K-means++ y mínimos cuadrados alternos (ALS) (Assefi et al., 2017).

MLlib resulta más adecuado para ejecutar sus algoritmos en un conjunto de datos grande. En cambio, si se tienen muchos conjuntos de datos pequeños en los que desea entrenar diferentes modelos de aprendizaje, sería mejor usar una biblioteca de aprendizaje de un solo nodo, por ejemplo, Weka o SciKit-Learn. De igual forma, es común que las canalizaciones de AA requieran entrenar el mismo algoritmo en un pequeño conjunto de datos con muchas configuraciones de parámetros, para poder elegir el mejor. En Spark se puede lograr esto usando *parallelize()* sobre su lista de parámetros para entrenar diferentes parámetros en diferentes nodos, usando una biblioteca de aprendizaje de un solo nodo en cada nodo (Karau et al., 2015).

MLlib en sí, resalta cuando tiene un conjunto de datos grande distribuidos en el que necesita entrenar un modelo.

En Spark 1.0 y 1.1, la interfaz de MLlib es de nivel relativamente bajo, lo que le brinda las funciones para solicitar diferentes tareas, pero no el flujo de trabajo de nivel superior que normalmente se requiere para una canalización de aprendizaje (por ejemplo, dividir la entrada en datos de entrenamiento y prueba, o probando muchas combinaciones de parámetros). En Spark 1.2, MLlib obtiene una API de canalización adicional para construir dichas canalizaciones. Esta API se asemeja a bibliotecas de nivel superior como SciKit-Learn y facilita la escritura de canalizaciones completas y autoajustables (Aziz et al., 2019).

2.6 Sistema de recomendaciones y filtrado colaborativo con MLlib

El FC es una técnica para SR en la que las calificaciones de los usuarios y las interacciones con varios productos se utilizan para recomendar nuevos. Solo necesita incluir una lista de interacciones usuario/producto: interacciones "explícitas" (es decir, calificaciones en un sitio de compras) o interacciones "implícitas" (por ejemplo, un usuario navegó por un producto pero no lo calificó) (Chuquitarco Chasiluisa, 2018).

Basándose únicamente en estas interacciones, los algoritmos de FC aprenden qué productos son similares entre sí (porque los mismos usuarios interactúan con ellos) y qué usuarios son similares entre sí, y pueden hacer nuevas recomendaciones. Si bien la API de MLlib habla de "usuarios" y "productos", también puede utilizar el FC para otras aplicaciones (Karau et al., 2015).

En un sistema de FC se pueden distinguir tres pasos fundamentales según (Monroy, 2003):

- Creación del perfil de usuario (Ratings). Es posible recolectar las preferencias de los usuarios

de dos modos: Implícito, por ejemplo al extraer los temas de interés de las páginas revisadas por el usuario, o explícito, como los ratings numéricos (calificaciones a ciertos ítems que agrega el usuario). Cualquier interacción con el sistema, podría potencialmente afectar el resultado de la recomendación. Es útil darle al usuario una retroalimentación sobre la calidad de su perfil, por cuanto los ratings dados por él, tienen mayor peso sobre las recomendaciones que aquellos valores calculados implícitamente. A mayor número de ítems calificados por el usuario, mayor es la probabilidad de obtener mejores recomendaciones. Al entregar los resultados acompañados de una retroalimentación, el usuario podría mejorar su perfil de forma explícita para buscar mejores recomendaciones en el futuro.

- El FC localiza personas con perfiles similares (vecinos). Es aquí donde el sistema muestra su verdadero valor sobre las recomendaciones hechas directamente por los usuarios, al permitir examinar miles de candidatos con diferentes preferencias y escoger de entre todo el conjunto los que sean más semejantes al usuario actual. Si los vecinos seleccionados por el sistema son los mejores para dicho usuario, de acuerdo a sus necesidades, las recomendaciones resultantes serán las mejores posibles.
- Los ratings de los vecinos se combinan para formar recomendaciones. En la mayoría de los casos, la predicción es simplemente un promedio ponderado de los ratings de usuarios.

La biblioteca MLlib como herramienta de FC sí presenta inconvenientes que no son novedad pues son desventajas generales de los SR con FC, consultar subsección equation (??) del capítulo primero para más detalles.

2.6.1 Retroalimentación explícita vs implícita

El enfoque estándar para el FC basado en factorización matricial trata las entradas en la matriz del elemento del usuario como preferencias explícitas dadas por el usuario al elemento, por ejemplo, los usuarios que otorgan calificaciones a películas (Gosh et al., 2021).

En muchos casos de uso del mundo real, es común tener acceso solo a comentarios implícitos (por ejemplo, vistas, clics, compras, me gusta, acciones, etc.). El enfoque utilizado en spark.ml para tratar estos datos se toma de FC para conjuntos de datos de retroalimentación implícita. Esencialmente, en lugar de intentar modelar la matriz de calificaciones directamente, este enfoque trata los datos como números que representan la fuerza en las observaciones de las acciones del usuario (como el número de clics o la duración acumulada que alguien pasó viendo una película). Luego, esos números se relacionan con el nivel de confianza en las preferencias observadas del usuario, en lugar de calificaciones explícitas otorgadas a los elementos. Luego, el modelo intenta encontrar factores latentes que pueden usarse para predecir la preferencia esperada de un usuario por un artículo (Spark, 2021).

2.6.2 Escala del parámetro de regularización

Se escala el parámetro de regularización *regParam* al resolver cada problema de mínimos cuadrados por el número de calificaciones que el usuario generó al actualizar los factores del usuario, o el número de calificaciones que recibió el producto al actualizar los factores del producto. Existe el parámetro *regParam* que se hace menos dependiente de la escala del conjunto de datos, por lo que podemos aplicar el mejor parámetro aprendido de un subconjunto muestreado al conjunto de datos completo y esperar un rendimiento similar (Spark, 2021).

2.6.3 Problema del arranque en frío

Al realizar predicciones con un modelo ALS, es común encontrar usuarios y/o elementos en el conjunto de datos de prueba que no estaban presentes durante el entrenamiento del modelo.

Generalmente esto puede ocurrir en dos escenarios:

- En producción, para nuevos usuarios o artículos que no tienen historial de calificación y en los que el modelo no ha sido entrenado (este es el llamado “problema del arranque en frío” o en inglés *cold-start strategy*).
- Durante la validación cruzada, los datos se dividen entre los conjuntos de formación y evaluación. Cuando se utilizan divisiones aleatorias simples como en *CrossValidator* o *TrainValidationSplit* de Spark, en realidad es muy común encontrar usuarios y/o elementos en el conjunto de evaluación que no están en el conjunto de entrenamiento.

De forma predeterminada, Spark asigna predicciones NaN al ejecutar *ALSModel.transform* cuando un usuario y/o factor de elemento no está presente en el modelo. Esto puede ser útil en un sistema de producción, ya que indica un nuevo usuario o elemento, por lo que el sistema puede tomar una decisión sobre alguna alternativa para usar como predicción (Spark, 2021).

Sin embargo, esto no es deseable durante la validación cruzada, ya que cualquier valor predicho de NaN dará como salida resultados de NaN para la métrica de evaluación (por ejemplo, cuando se usa *RegressionEvaluator*). Esto hace que la selección del modelo sea imposible. Spark permite a los usuarios establecer el parámetro *coldStartStrategy* en "drop" para eliminar cualquier fila en el *DataFrame* de predicciones que contengan valores NaN. La métrica de evaluación se calculará luego sobre los datos que no son de NaN y será válida (Gosh et al., 2021).

2.6.4 RDD

La abstracción de datos en Spark se realiza por medio de los RDD, mencionados en secciones anteriores. Un RDD es una colección de objetos de solo lectura particionada sobre un conjunto de máquinas que puede ser reconstruida si se pierde alguna de las particiones. Spark ofrece cuatro formas de crear un RDD (Equiza, 2015):

- A partir de un fichero perteneciente a un sistema de ficheros compartidos (como HDFS).
- Paralelizando una colección Scala, es decir, dividiéndola en segmentos que serán enviados a diferentes nodos.
- Transformando otro RDD existente.
- Cambiando la persistencia de otro RDD existente.

2.6.5 Algoritmo de mínimos cuadrados alternos (ALS)

Mínimos cuadrados alternos (ALS) es un algoritmo iterativo, en cada iteración, el algoritmo fija alternativamente una matriz de factores y resuelve la otra, y este proceso continúa hasta que haya convergencia. La implementación que presenta MLlib del algoritmo ALS aprovecha el soporte eficiente de Spark para el cálculo iterativo distribuido. Utiliza LAPACK nativo para lograr un alto rendimiento y escala a miles de millones de calificaciones en grupos de productos básicos (Yabuz y Meng, 2014).

ALS es también un algoritmo de factorización matricial y se ejecuta de forma paralela. En Spark se crea fundamentalmente para problemas de FC a gran escala.

Algunas ideas de alto nivel detrás de ALS son las siguientes según (Liao, 2018):

- Su función objetivo es ligeramente diferente a Funk SVD: ALS usa la regularización L2 mientras que Funk usa la regularización L1.
- Su rutina de entrenamiento es diferente: ALS minimiza dos funciones de pérdida alternativamente, primero mantiene fija la matriz de usuario y ejecuta el descenso de gradiente con la matriz de elementos, luego mantiene fija la matriz de elementos y ejecuta el descenso de gradiente con la matriz de usuario.
- Su escalabilidad: ALS ejecuta su descenso de gradiente en paralelo a través de múltiples particiones de los datos de entrenamiento subyacentes de un grupo de máquinas

La implementación que incluye MLlib de ALS se encuentra en la clase *mllib.recommendation.ALS*. ALS funciona determinando un vector de características para cada usuario y producto, de modo que el producto escalar del vector de un usuario y el de un producto se acerque a su puntuación. ALS en Spark toma los siguientes parámetros (Karau et al., 2015):

- Rango (rank): Es el tamaño de los vectores de características a utilizar, rangos más grandes pueden conducir a mejores modelos pero son más costosos de calcular (predeterminado: 10).
- Iteraciones (iterations): El número de iteraciones para ejecutar (predeterminado: 10).
- Lambda (lambda): El parámetro de regularización (predeterminado: 0.01).
- Alfa (alpha): Es una constante utilizada para calcular la confianza en ALS implícita (predeterminado: 1.0).

- *numUserBlocks*, *numProductBlocks*: Número de bloques en los que dividir los datos del usuario y del producto para controlar el paralelismo, puede pasar `-1` para permitir que MLlib determine esto automáticamente, siendo este el comportamiento predeterminado.

Para usar ALS, debe recibir un RDD de objetos del tipo *mllib.recommendation.Rating*, cada uno de los cuales contiene una ID de usuario, una ID de producto y una calificación (ya sea una calificación explícita o un comentario implícito). Un desafío con la implementación es que cada ID debe ser un número entero de 32 bits. Si sus ID son cadenas o números mayores, se recomienda usar el código hash de cada ID, incluso si dos usuarios o productos se asignan al mismo ID, los resultados generales pueden ser buenos. Alternativamente, puede devolver, *broadcast()*, una tabla de asignaciones de ID de producto a enteros para darles ID únicos (Gosh et al., 2021).

ALS devuelve un modelo de matriz factorizado (*MatrixFactorizationModel*) que representa sus resultados, se puede usar para predecir *predict()* calificaciones para un RDD de pares (ID de usuario, ID de producto), es decir, productos recomendados para un usuario determinado. Tenga en cuenta que, a diferencia de otros modelos en MLlib, el modelo *MatrixFactorizationModel* es grande y contiene un vector para cada usuario y producto. Esto significa que no se puede guardar en el disco y luego volver a cargarlo en otra ejecución de su programa. En su lugar, puede guardar los RDD de los vectores de características producidos en él, *model.userFeatures* y *model.productFeatures*, en un sistema de archivos distribuido (Karau et al., 2015).

Existen dos variantes de ALS: para calificaciones explícitas (el valor predeterminado) y para calificaciones implícitas (que se habilita llamando a *ALS.trainImplicit()* en lugar de *ALS.train()*). Con calificaciones explícitas, la calificación de cada usuario para un producto debe ser una puntuación (ejemplo de 1 a 5 estrellas) y las calificaciones previstas serán puntuaciones. Con retroalimentación implícita, cada calificación representa una confianza en que los usuarios interactuarán con un elemento dado (por ejemplo, la calificación puede subir mientras más veces un usuario visita una página web), y los elementos antes expresados serán valores de confianza (Gosh et al., 2021).

2.6.6 PySpark para una implementación en Python

PySpark es una interfaz para Apache Spark en Python. No solo le permite escribir aplicaciones Spark utilizando las API de Python, sino que también proporciona el shell de PySpark para analizar interactivamente sus datos en un entorno distribuido. PySpark es compatible con la mayoría de las funciones de Spark, como Spark SQL, DataFrame, Streaming, MLlib y Spark Core previamente descritas (PySpark, 2021).

2.6.7 Conclusiones parciales de la sección

Se analizaron las diferentes funcionalidades y ventajas que brinda Apache Spark y su biblioteca MILib para el AA y los SR. El FC se implementa mediante el algoritmo ALS que a partir de un modelo de matriz factorizada (*MatrixFactorizationModel*) permite asociar el rating que le da cada usuario a cada producto y luego seleccionar los top N productos de mayor rating para un usuario X. Spark permite que se implemente el sistema de recomendación en Java, Scala o Python. En un comienzo es necesario parsear los datos que se tienen, dígame, usuarios, materiales y ratings de forma que sean un tipo de dato RDD para luego poder aplicarle el algoritmo ALS. La implementación en Python utiliza la interfaz PySpark que permite el trabajo con Python en Spark.

Una propuesta de algoritmo, a grandes rasgos, de FC en Spark con MILib, que de solución a la recomendación de materiales de estudio a los estudiantes en la plataforma Moodle puede ser la siguiente:

1. Cargar y parsear los datos de entrada, de forma que sean estructuras de datos RDD.
2. Construir el modelo de recomendación utilizando ALS con retroalimentación explícita (en este caso, la calificación son las emociones obtenidas con el bloque Emociones).
3. Evaluar el modelo a partir del valor de la medida Mean Square Error (MSE).

2.7 Diseño de un bloque para Moodle basado en SR y FC

Partiendo de los resultados que arroja el bloque Emociones, descrito en el capítulo primero, se propone la implementación de un bloque complementario llamado Recomendaciones. A grandes rasgos, el objetivo central de la propuesta de creación del bloque Recomendaciones es utilizar las emociones reconocidas en los estudiantes por el bloque Emociones para determinar cuán interesante le resulta un material de estudio determinado y recomendar así materiales a fines con el ritmo de aprendizaje de cada alumno.

A continuación se describe de manera genérica el flujo de trabajo del bloque Recomendaciones:

1. Con el bloque Emociones se guardan las reacciones de estudiantes seleccionados al leer un material de estudio de un tema en un curso específico.
2. Se recomienda al estudiante mientras lee, otros materiales similares que le pueden interesar.
3. El estudiante selecciona uno de los nuevos materiales recomendados y según la emoción detectada ante la interacción, se le asigna un rating al material que indica la aceptación del alumno.

El sistema de recomendación mediante FC que facilita la biblioteca MILib de Spark a partir del algoritmo ALS, hace posible generar recomendaciones de materiales de estudio. Los materiales recomendados el profesor los introduce asociados a un tema específico de un curso y se guardan en una base de datos. El algoritmo ALS explicado en secciones anteriores, toma como entradas, parseadas

en formato de RDD, un vector tipo (*idEstudiante*, *idMaterial*, *rating*). El procesamiento se realiza con valores numéricos que se parsean a partir de los datos originales guardados en una base de datos. La estructura de base de datos se propone en subsecciones posteriores.

2.7.1 Propuesta de diseño gráfico

El diseño gráfico se define como "el arte y la práctica de planificar y proyectar ideas y experiencias con contenido visual y textual". También se define como comunicación visual que consiste en la proyección y producción de objetivos visuales con el propósito de transmitir un mensaje de interés público de manera eficaz (Significados, 2013).

Los actores que interactúan con el sistema son profesores y estudiantes. Cada uno obtiene una vista diferente del bloque Recomendaciones. A continuación se define una sencilla propuesta de diseño gráfico para ambos actores.



Figura 2.3: Vista del profesor



Figura 2.4: Vista del estudiante

El profesor sí tiene acceso a editar el bloque Recomendaciones, es decir, a añadirlo en el curso que desee, eliminarlo e introducir nuevos materiales para que sean sugeridos a los estudiantes en un tema

determinado. El diseño de este bloque está orientado fundamentalmente al profesor; pues la idea es que pueda nutrir lo temas de los diferentes cursos con suficiente material de estudio; material acorde a las diferentes necesidades de los estudiantes. En la medida que el profesor sea capaz de ofrecer a los estudiantes, materiales que oscilen desde los niveles de comprensión más básicos y sencillos, hasta los más complejos; el bloque Recomendaciones mediante su SR basado en FC será más capaz de facilitar el autoestudio de los alumnos facilitándole materiales acordes a su perfil.

El estudiante solo tiene contacto con el bloque a través del cuadro con recomendaciones con diferentes materiales de estudio. El formato que caracteriza a los materiales sugeridos es primero; una descripción con el nombre del material, número de edición y autor o autores; segundo, una imagen de portada identificativa del material que se sugiere y por último; el URL mediante el cual se accede directamente al material, este debe estar en la nube de la UCLV para garantizar el acceso gratuito y rápido.

El bloque Recomendaciones resulta de interés para potenciar una interacción productiva y motivadora del estudiante con la plataforma; de forma que este encuentre una respuesta en tiempo real a sus preocupaciones educativas mientras lee un material determinado y no están comprendiendo el contenido o quieren consultar otra bibliografía.

Cuando el alumno encuentra respuestas, comienza a sentirse motivado; esta motivación muchas veces faltaba en los EVA al no tener asistencia directa como en las clases presenciales. El bloque Recomendaciones resulta una alternativa para que el estudiante se sienta familiarizado con el entorno virtual al reconocer que sus necesidades están siendo identificadas y atendidas.

2.7.2 Diseño UML para el bloque Recomendaciones

Tener una notación expresiva y bien definida resulta fundamental para el proceso de desarrollo de software. El lenguaje de modelado unificado (UML) es el lenguaje de modelado principal utilizado para analizar, especificar y diseñar sistemas de software (Booch et al., 2008).

El modelo UML que se construya representará, con cierto nivel de fidelidad, el sistema real a implementar. Debido a que resulta imposible capturar todos los detalles de un sistema de software complejo en un solo diagrama, UML tiene numerosos tipos de diagramas, proporcionando cada uno diferentes vistas del sistema (Booch et al., 2008).

Los diagramas UML se pueden clasificar en dos grupos: diagramas de estructura y diagramas de comportamiento. Los diagramas de estructura E se utilizan para mostrar la estructura estática de los elementos del sistema y los de comportamiento describen los eventos que suceden dinámicamente en todos los sistemas intensivos en software. Por ejemplo, los objetos se crean y se destruyen, se envían mensajes entre sí de manera ordenada, y en algunos sistemas, los eventos externos desencadenan operaciones en determinados objetos.

Algunos de los diagramas de comportamiento que se existen según (Booch et al., 2008) son los que se mencionan a continuación:

- Diagrama de casos de uso.
- Diagrama de actividad.
- Diagrama de la máquina de estado.
- Diagramas de interacción.
- Diagrama de secuencia.
- Diagrama de comunicación.
- Diagrama de descripción general de la interacción.
- Diagrama de tiempos.

Es de interés la modelación de el diagrama de casos de uso de entre los diagramas de comportamiento existentes, para que ilustren el diseño y funcionamiento del sistema de recomendación a implementar. La notación de UML empleada para la modelación sigue las normas planteadas en (Booch et al., 2008) y el software a utilizar es Star UML.

2.7.3 Actores

Los actores son entidades que interactúan con el sistema, pueden ser personas u otro sistema. Una forma de pensar en los actores es considerar los roles que desempeñan. Por ejemplo, en el mundo real las personas o los sistemas pueden desempeñar diferentes funciones, por ejemplo, una persona puede ser un vendedor, un gerente, un padre, un artista, etc (Booch et al., 2008).

Para el problema actual se definen dos actores que interactúan con el sistema, el estudiante y el profesor. En el siguiente gráfico se muestra su modelación en UML.



Figura 2.5: Modelación de los actores

2.7.4 Casos de uso

Los casos de uso representan lo que los actores quieren que su sistema haga por ellos y se modelan como óvalos. Un caso de uso debe ser un flujo completo de actividad, desde el punto de vista del actor. Para el problema actual se tienen casos de uso para el actor estudiante y para el profesor, a continuación se modelan por separados para cada actor respectivamente (Booch et al., 2008).

2.7.5 Diagrama de actores y casos de uso

Un diagrama de caso de uso puede incluir varios casos de uso y las relaciones entre casos de uso y las personas, los grupos o los sistemas que interactúan para llevar a cabo el caso de uso (Booch et al., 2008).

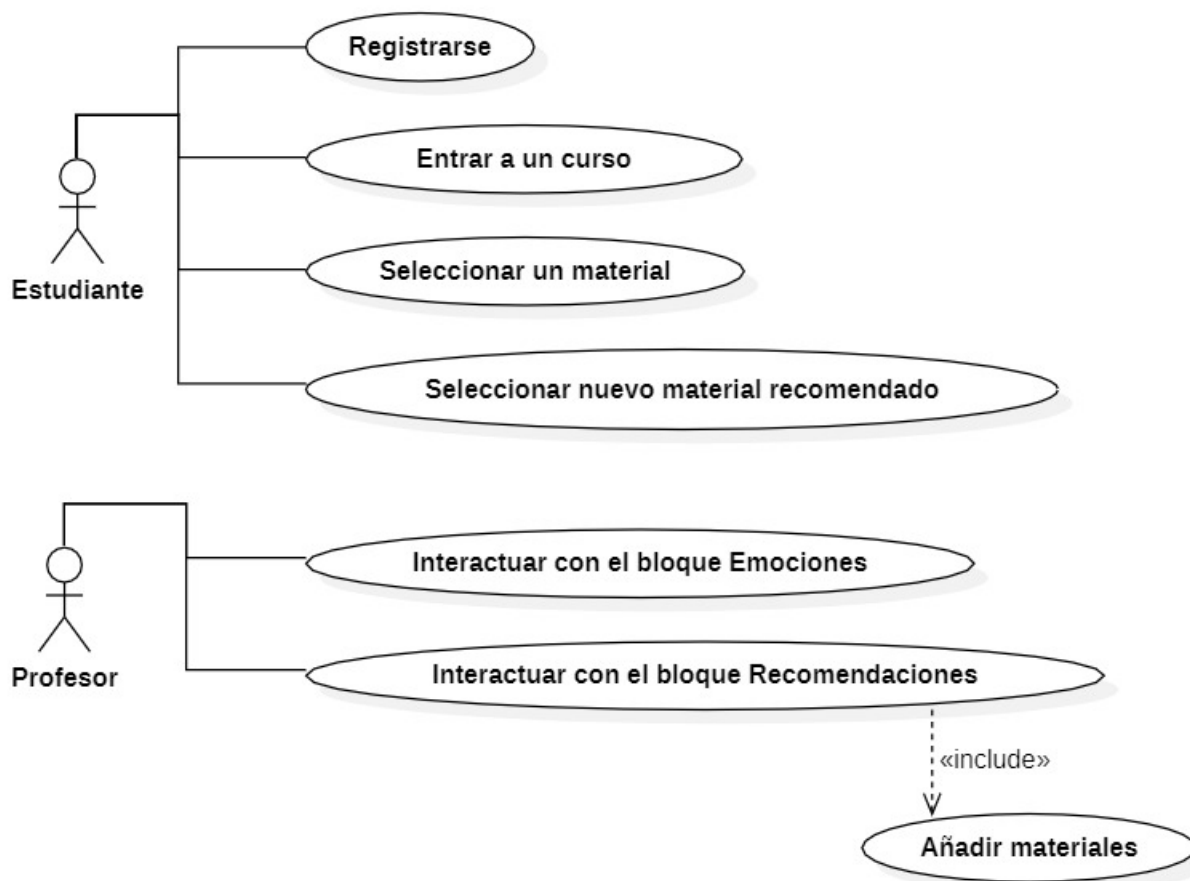


Figura 2.6: Modelación de los casos de uso y actores

2.7.6 Descripción de los casos de uso

Una especificación de caso de uso proporciona detalles textuales de un caso de uso. Los parámetros a partir de los cuales se describen los casos de uso, para cada actor, se tomaron de la página oficial de IBM (IBM, 2021) que brinda información formal sobre el modelado UML.

Se describen a continuación los casos de uso correspondientes al actor Estudiante.

| | |
|---------------------------------|---|
| Nombre | Registrarse. |
| Actor/es que interactúan | Estudiante. |
| Descripción | El estudiante se registra en la plataforma Moodle. |
| Flujo básico | El estudiante introduce su usuario y contraseña en la plataforma Moodle para tener acceso a sus cursos. |
| Flujo alternativo | Si el estudiante olvidó su usuario o contraseña, el sistema brinda la opción de recuperación de datos mediante email. |
| Condiciones previas | El estudiante debe disponer de conectividad a internet y conocer su usuario y contraseña. |
| Condiciones posteriores | El estudiante está en la plataforma y selecciona el curso al que quiere acceder |

Tabla 2.2: Especificación del caso de uso *Registrarse*

2.7 Diseño de un bloque para Moodle basado en SR y FC

| | |
|---------------------------------|---|
| Nombre | Entrar a un curso. |
| Actor/es que interactúan | Estudiante. |
| Descripción | El estudiante selecciona el curso en el cual va a trabajar. |
| Flujo básico | El estudiante selecciona uno de los temas de estudio que tiene el curso hasta seleccionar uno. |
| Flujo alternativo | En caso de que el estudiante no pueda acceder al curso seleccionado, ya sea porque esta protegido por contraseña o porque no está disponible, el sistema brinda la opción de contactar al profesor responsable del curso. |
| Condiciones previas | El estudiante debe estar registrado en el Moodle y conocer sus cursos. |
| Condiciones posteriores | El estudiante selecciona el tema del cual va a revisar materiales de estudio. |

Tabla 2.3: Especificación del caso de uso *Entrar a un curso*

| | |
|---------------------------------|--|
| Nombre | Seleccionar un material. |
| Actor/es que interactúan | Estudiante. |
| Descripción | El estudiante selecciona un material de estudio. |
| Flujo básico | El estudiante accede a uno de los materiales de estudio disponibles para un tema de un curso. |
| Flujo alternativo | En caso de que no hayan materiales disponibles en la plataforma para el tema seleccionado, el sistema brinda la opción de contactar al profesor responsable del curso. |
| Condiciones previas | El estudiante debe seleccionar el curso que va a revisar. |
| Condiciones posteriores | El estudiante recibe una lista de sugerencias de otros materiales de estudio relacionados que le pueden interesar. |

Tabla 2.4: Especificación del caso de uso *Seleccionar un material*.

2.7 Diseño de un bloque para Moodle basado en SR y FC

| | |
|---------------------------------|---|
| Nombre | Seleccionar nuevo material recomendado. |
| Actor/es que interactúan | Estudiante. |
| Descripción | El estudiante selecciona un nuevo material de los sugeridos por el sistema. |
| Flujo básico | El estudiante selecciona un nuevo material de estudio de una lista que el sistema le sugiere. La lista brinda una breve descripción de los materiales y los URL en la nube de la universidad para acceder. |
| Flujo alternativo | En caso de que no hayan nuevos materiales disponibles como sugerencia, el sistema brinda la opción de contactar al profesor para informar la falta de documentación. |
| Condiciones previas | El estudiante debe seleccionar el tema al cual va a acceder. |
| Condiciones posteriores | El estudiante recibe una lista de nuevas sugerencias de materiales de estudio relacionados que le pueden interesar. |

Tabla 2.5: Especificación del caso de uso *Seleccionar nuevo material recomendado*.

Se describen a continuación los casos de uso correspondientes al actor Profesor.

2.7 Diseño de un bloque para Moodle basado en SR y FC

| | |
|---------------------------------|--|
| Nombre | Interactuar con el bloque Emociones. |
| Actor/es que interactúan | Profesor. |
| Descripción | El profesor puede acceder al bloque Emociones y revisar las emociones de estudiantes seleccionados. |
| Flujo básico | El profesor al acceder al bloque Emociones puede seleccionar estudiantes para guardar sus emociones al navegar por el curso, puede visualizar y descargar las emociones reconocidas, así como editar la aparición del bloque o no en un curso determinado. |
| Flujo alternativo | El sistema brinda la opción al profesor de contactar con el administrador de red en caso de que desconozca como trabajar con el bloque y editarlo. |
| Condiciones previas | Registrarse como profesor en la plataforma Moodle y conocer el funcionamiento del bloque Emociones. |
| Condiciones posteriores | El profesor pasa el bloque Recomendaciones para añadir nuevos materiales de trabajo para los estudiantes en los diferentes temas del curso. |

Tabla 2.6: Especificación del caso de uso *Interactuar con el bloque Emociones*.

2.7 Diseño de un bloque para Moodle basado en SR y FC

| | |
|---------------------------------|--|
| Nombre | Interactuar con el bloque Recomendaciones. |
| Actor/es que interactúan | Profesor. |
| Descripción | El profesor puede añadir materiales para sugerir en un tema de un curso determinado. |
| Flujo básico | El profesor añade una descripción, imagen y URL en la nube de la UCLV del material a guardar en la base de datos para que sea sugerido por el sistema de recomendación. |
| Flujo alternativo | Puede pasar que el profesor desconozca como añadir, mover o eliminar el bloque Recomendaciones, en este caso el sistema brinda la opción de contactar al administrador de red. |
| Condiciones previas | Tener instalado el bloque Emociones y Recomendaciones. |
| Condiciones posteriores | El profesor puede seguir añadiendo materiales o salir del bloque. |

Tabla 2.7: Especificación del caso de uso *Interactuar con el bloque Recomendaciones*.

Otro caso de uso incluido en el caso de uso *Interactuar con el bloque Recomendaciones* es el caso de uso *Añadir materiales*, a continuación se describe.

2.7 Diseño de un bloque para Moodle basado en SR y FC

| | |
|---------------------------------|--|
| Nombre | Añadir materiales. |
| Actor/es que interactúan | Profesor. |
| Descripción | El profesor añade un material a la base de datos. |
| Flujo básico | El profesor guarda un material a recomendar con una breve descripción que incluye autor y año de lanzamiento del material, imagen de la portada oficial y URL en la nube de la UCLV para que el alumno acceda. |
| Flujo alternativo | Si el profesor desconoce como guardar los datos necesarios de cada material que va a añadir, el sistema brinda la opción de contactar al administrador de red para ayuda. |
| Condiciones previas | Tener instalado el bloque Recomendaciones y acceder a la opción de añadir material. |
| Condiciones posteriores | El profesor puede seguir añadiendo materiales o salir del bloque. |

Tabla 2.8: Especificación del caso de uso *Añadir materiales*.

2.7.7 Descripción del diseño

La principal novedad que trae el bloque Recomendaciones es el uso de la tecnología que brinda Spark con su algoritmo ALS para recomendar materiales a los estudiantes según sus emociones; es decir, gracias al AA, el algoritmo toma las emociones de un estudiante ante un material como rating y el mecanismo de ALS recomiendan materiales que hayan sido calificados de forma similar por usuarios con emociones similares. En este caso se debe utilizar el método *ALS.train()* debido a que la calificación que expresan los usuarios se basa en las emociones; es decir son explícitas; pues el usuario con su expresión facial está enviando una calificación a un material de estudio, equivalente a si lo hiciera de forma manual mediante una puntuación. A medida que incrementa la cantidad de estudiantes que han expresado su emoción ante varios materiales, el sistema mejora sus recomendaciones pues se va superando el llamado inicio en frío, explicado en secciones anteriores. En un inicio el sistema hace recomendaciones básicamente aleatorias pues al no tener usuarios que hayan calificado los documentos brindados por el profesor, las recomendaciones se realizan por defecto según las estrategias de ALS ante el inicio en frío mencionado en subsecciones anteriores. El algoritmo ALS devuelve un modelo de matriz factorizada y mediante el método *predict()* se obtiene



Figura 2.7: Propuesta de tabla para el bloque Recomendaciones

una sugerencia para un par del tipo (*idEstudiante*, *idMaterial*). Sin embargo, mediante una llamada a *model.recommendProducts(idEstudiante, idMaterial)* se pueden obtener los top N materiales recomendados a un usuario específico. Este último método sería el utilizado para obtener los materiales más afines con a un estudiante determinado. A medida que se va enriqueciendo el sistema con usuarios que reaccionan ante los materiales que examinan, el sistema realmente facilita el trabajo del estudiante en cuanto a la búsqueda de bibliografía.

2.7.8 Propuesta de estructura de base de datos para el bloque

Cada tabla que se cree en Moodle debe tener un campo identificador. El nombre de cada tabla debe contener en su nombre el mismo del módulo al que pertenece precedidos del prefijo *mdl_*. Al bloque Recomendaciones se le asocia una tabla para guardar los datos necesarios, como el curso y tópico donde el profesor va a agregar materiales, así como un identificador para estos. El nombre de esta tabla es *mdl_block_recomendaciones* con seis campos incluyendo el identificador. Con la herramienta ERStudio se modeló una propuesta de estructura sencilla. Los campos se definieron de tipo bigint debido a que son identificadores.

El algoritmo ALS define un modelo de matriz factorizado, el cual se explicó en secciones anteriores, donde se va guardando el aprendizaje en dos tablas, una que asocia los estudiantes con los materiales que se le asocian a cada uno y otra con los materiales y los estudiantes asociados. La tabla antes propuesta tiene el objetivo principal de guardar los materiales que el profesor va añadiendo a cada tema dentro de un curso determinado.

2.8 Conclusiones del capítulo

En este capítulo se analizó una propuesta de bloque para la plataforma Moodle que complementa el bloque Emociones explicado en el capítulo primero. El bloque Recomendaciones tiene el objetivo de que el profesor pueda añadir materiales de estudio referentes a un tema de un curso específico, que se guardan en una base de datos para luego ser recomendados a los estudiantes según su necesidad educativa. Los estudiantes interactúan con este bloque solamente seleccionando materiales de entre los que el sistema le sugiere cuando están examinando otro. Para que las recomendaciones se lleguen a ejecutar, es necesario que el estudiante permita el acceso a la cámara del dispositivo del cual esté

utilizando Moodle para que el bloque Emociones reconozca su estado mientras explora un material; de esta forma se obtiene una calificación de cuán útil e interesante le está resultando ese material y poder recomendarle mejores propuestas.

CONCLUSIONES

1. El filtrado colaborativo es un tipo de sistema de recomendación que hace posible agrupar a los usuarios según sus características y sugerirle ítems con precisión.
2. Existen diversas herramientas que brindan bibliotecas para el filtrado colaborativo como Mahout, H2O, BigML, Spark entre otras. Apache Spark resulta ser de los entornos más completos para los trabajos de Aprendizaje Automático; gracias a su biblioteca MLlib que brinda los algoritmos más eficientes para la implementación de sistemas de recomendación mediante filtrado colaborativo.
3. La implementación de un bloque de Moodle que facilite al alumno materiales de estudio en forma de sugerencia, basadas en las emociones que muestra es una buena propuesta para mejorar los procesos de enseñanza y aprendizaje que se desarrollan en este ambiente virtual de aprendizaje.
4. La tecnología que hace posible esta implementación es la biblioteca MLlib del entorno Apache Spark. El algoritmo ALS, de esta biblioteca, es el encargado de generar las recomendaciones e ir las mejorando a medida que nuevos usuarios y materiales entran al sistema. La biblioteca MLlib permite implementaciones en Java, Scala y Python; resulta de interés desarrollar la implementación en Python a través de la biblioteca PySpark, garantizando así la compatibilidad con el bloque Emociones implementado en este lenguaje.
5. La calificación que el algoritmo ALS asigna a los materiales, en este caso, es explícita, porque se toma la emoción del estudiante mientras consulta el material para calificar este último.
6. El bloque Recomendaciones, propuesto en el presente trabajo de diploma, tiene dos vistas fundamentales; la del estudiante y la del profesor. La vista del estudiante tiene recursos académicos sugeridos; es decir, materiales similares al que está revisando en ese momento y que se ajustan a su perfil. El perfil del alumno se ha construido mediante el autoaprendizaje del algoritmo ALS basado en la información que se extrae de las emociones que brinda el bloque Emociones. La vista del profesor es solamente el acceso a una base de datos donde se guarda la información de los materiales que va añadiendo. El papel del profesor en los resultados que

arroje este bloque a los estudiantes es fundamental; pues de la variedad de documentos con diversos niveles de complejidad que el profesor añada a la base de datos, depende la calidad y variedad de las sugerencias que reciban los estudiantes.

RECOMENDACIONES

1. Implementar el bloque Recomendaciones utilizando Apache Spark y su biblioteca MLlib; debido a lo novedoso y útil que resulta contar con un bloque en Moodle que sea capaz de sugerir materiales de estudio a los estudiantes según sus emociones; es decir, materiales que se ajusten a su perfil y necesidad académica.
2. Incorporar al bloque la capacidad de acceder a Internet de forma rápida para que el profesor pueda añadir materiales de estudio en línea y no solamente los que añade a la nube de la UCLV para garantizar el acceso gratuito.
3. Incorporar al bloque, opciones para que el estudiante además de expresar la utilidad de los materiales mediante sus expresiones faciales; también puedan expresar una opinión explícita directa, por ejemplo mediante un número del uno al cinco. De esta forma se complementa la calificación extraída de la emoción y se añade más información al sistema de recomendación.

REFERENCIAS

- Aljunid, M. F. y Manjaiah, D. H. (2019). Movie Recommender System Based on Collaborative Filtering Using Apache Spark, in V. E. Balas, N. Sharma y A. Chakrabarti (eds), *Data Management, Analytics and Innovation*, Advances in Intelligent Systems and Computing, Springer, Singapore, pp. 283–295.
- Anil, R., Capan, G., Drost-Fromm, I., Dunning, T., Friedman, E., Grant, T., Quinn, S., Ranjan, P. y Schelter, S. (2020). Apache Mahout: Machine Learning on Distributed Data *Flow Systems*, p. 6.
- Araque, I., Montilla, L., Meleán, R. y Arrieta, X. (2018). Entornos virtuales para el aprendizaje: una mirada desde la teoría de los campos conceptuales, *Góndola, enseñanza y aprendizaje de las ciencias* **13**(1): 86–100. Number: 1.
URL: <https://revistas.udistrital.edu.co/index.php/GDLA/article/view/11721>
- Assefi, M., Behraves, E., Liu, G. y Tafti, A. P. (2017). Big data machine learning using apache spark MLlib, *2017 IEEE International Conference on Big Data (Big Data)*, pp. 3492–3498.
- Aziz, K., Zaidouni, D. y Bellafkih, M. (2019). Leveraging resource management for efficient performance of Apache Spark, *Journal of Big Data* **6**(1): 78.
URL: <https://doi.org/10.1186/s40537-019-0240-1>
- Baque, P. G. C. y Marcillo, C. (2020). Estrategias pedagógicas innovadoras en entornos virtuales de aprendizaje, *Dominio de las Ciencias* **6**(3): 56–77. Publisher: Polo de Capacitación, Investigación y Publicación (POCAIP) Section: Dominio de las Ciencias.
URL: <https://dialnet.unirioja.es/servlet/articulo?codigo=7539680>
- Bericat Alastuey, E. (2012). Emociones. Accepted: 2016-10-19T09:28:52Z Publisher: Arrangement.
URL: <https://idus.us.es/handle/11441/47752>
- Bonaci, M. y Zecevic, P. (2016). Chapter 14. Deep learning on Spark with H2O. Spark in Action.
URL: <https://livebook.manning.com/book/spark-in-action/chapter-14/>

- Booch, G., Maksimchuk, R. A., Engle, M. W., Young, B. J., Connallen, J. y Houston, K. A. (2008). Object-oriented analysis and design with applications, third edition, *ACM SIGSOFT Software Engineering Notes* **33**(5): 29–29.
URL: <https://dl.acm.org/doi/10.1145/1402521.1413138>
- Bosquez, V., Sanz, C., Baldassarri, S., Ribadeneira, E., Valencia, G., Barragan, R., Camacho, a., Shauri-Romero, J. y Camacho-Castillo, L. A. (2018). LA COMPUTACIÓN AFECTIVA: EMOCIONES, TECNOLOGÍAS Y SU RELACIÓN CON LA EDUCACIÓN VIRTUAL, *Revista de Investigación Talentos* **5**(1): 94–103. Number: 1.
URL: <https://talentos.ueb.edu.ec/index.php/talentos/article/view/35>
- B.Thorat, P., M. Goudar, R. y Barve, S. (2015). Survey on Collaborative Filtering, Content-based Filtering and Hybrid Recommendation System, *International Journal of Computer Applications* **110**(4): 31–36.
URL: <http://research.ijcaonline.org/volume110/number4/pxc3900760.pdf>
- Carrillo, G. (2013). Recomendación de Objetos de Aprendizaje basado en el Perfil del Usuario y la Información de Atención Contextualizada.
- Chuquitarco Chasiluisa, J. A. (2018). Estudio de algoritmos de filtrado colaborativo para sistemas recomendadores de información. Accepted: 2020-07-31T18:50:36Z Publisher: Ecuador: Latacunga: Universidad Técnica de Cotopaxi: Facultad de Ciencias de la Ingeniería y Aplicadas.
URL: <http://repositorio.utc.edu.ec/handle/27000/5630>
- Coll, C. (2004). Psicología de la educación y prácticas educativas mediadas por las tecnologías de la información y la comunicación, p. 25.
- Equiza, E. I. (2015). Sistemas de Recomendación en Spark, p. 62.
- Escárzaga, J. F., Varela, J. G. D. y Martínez, P. L. M. (2020). De la educación presencial a la educación a distancia en época de pandemia por Covid 19. Experiencias de los docentes, *Revista Electrónica Sobre Cuerpos Académicos y Grupos de Investigación* **7**(14): 87–110. Number: 14.
URL: <http://www.cagi.org.mx/index.php/CAGI/article/view/212>
- Gerón, A. (2019). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 2nd Edition [Book]. ISBN: 9781492032649.
URL: <https://www.oreilly.com/library/view/hands-on-machine-learning/9781492032632/>
- González, R. R. y Medina, N. M. (2011). Implementación de una estrategia de aprendizaje cooperativo en la plataforma educativa Moodle, *Cuadernos de Educación y Desarrollo* (27). Publisher:

- Servicios Académicos Intercontinentales SL.
URL: <https://ideas.repec.org/a/erv/cedced/y2011i2754.html>
- Gosh, S., Nahar, N., Wahab, M. A., Biswas, M., Hossain, M. S. y Andersson, K. (2021). Recommendation System for E-commerce Using Alternating Least Squares (ALS) on Apache Spark, in P. Vasant, I. Zelinka y G.-W. Weber (eds), *Intelligent Computing and Optimization*, Advances in Intelligent Systems and Computing, Springer International Publishing, Cham, pp. 880–893.
- Guerri, M. (2021). Qué son las EMOCIONES? Tipos y componentes conductuales. Section: Emociones.
URL: <https://www.psicoactiva.com/blog/que-son-las-emociones/>
- Guillén, F. (2020). 484. BigML - Machine Learning fácil para no programadores.
URL: <https://acelerandoempresas.com/484-bigml-machine-learning-facil-para-no-programadores/>
- H2O.ai, H. (2021). H2O Sparkling Water.
URL: <https://www.h2o.ai/products/h2o-sparkling-water/>
- Herlocker, J. L., Konstan, J. A. y Riedl, J. (2000). Explaining collaborative filtering recommendations, *Proceedings of the 2000 ACM conference on Computer supported cooperative work, CSCW '00*, Association for Computing Machinery, New York, NY, USA, pp. 241–250.
URL: <https://doi.org/10.1145/358916.358995>
- Hernandez, R. M. (2017). Impacto de las TIC en la educación: Retos y Perspectivas, *Propósitos y Representaciones* **5**(1): 325–347. Number: 1.
URL: <https://revistas.usil.edu.pe/index.php/pyr/article/view/149>
- IBM, I. (2021). IBM Docs.
URL: <https://prod.ibmdocs-production-dal-6099123ce774e592a519d7c33db8265e-0000.us-south.containers.appdomain.cloud/docs/es/elm/6.0.3?topic=requirements-defining-use-cases>
- Ignite (2021). La importancia de las TIC en la educación durante el Coronavirus | Ignite Online.
URL: <https://igniteonline.la/la-importancia-de-las-tic-en-la-educacion-durante-el-coronavirus/>
- James, W. (1985). What is an emotion?, *Studies in Psychology* **6**(21): 57–73. Publisher: Routledge
 _eprint: <https://doi.org/10.1080/02109395.1985.10821418>.
URL: <https://doi.org/10.1080/02109395.1985.10821418>

- Jiménez, A. M. H. (2015). Una mirada reflexiva acerca de las TIC en Educación Superior, *Revista Electrónica de Investigación Educativa* **17**(1). Number: 1.
URL: <https://redie.uabc.mx/index.php/redie/article/view/726>
- Karau, H., Konwinski, A., Wendell, P. y Zaharia, M. (2015). *Learning Spark*, first edition edn, O'Reilly, Beijing ; Sebastopol. OCLC: ocn844872440.
- Karau, H. y Warren, R. (2017). *High Performance Spark: Best Practices for Scaling and Optimizing Apache Spark*, .O'Reilly Media, Inc.". Google-Books-ID: 90glDwAAQBAJ.
- Leris López, D., Velamazán Gimeno, a. y Vea Muniesa, F. (2015). Aprendizaje adaptativo en moodle: tres casos prácticos, *Educ. knowl. soc.* Number: ART-2015-103697.
URL: <https://zaguan.unizar.es/record/64384>
- Liao, K. (2018). Prototyping a Recommender System Step by Step Part 2: Alternating Least Square (ALS) MatrixâŠ.
URL: <https://towardsdatascience.com/prototyping-a-recommender-system-step-by-step-part-2-alternating-least-square-als-matrix-4a76c58714a1>
- Lima Montenegro, S. y Fernandez Nodarse, F. A. (2017). La educación a distancia en entornos virtuales de enseñanza aprendizaje. Reflexiones didácticas. Accepted: 2021-07-17T21:25:05Z
 Publisher: Facultad de Educación de la Universidad de Matanzas.
URL: <https://acervodigitaleducativo.mx/handle/acervodigitaledu/46955>
- López Rosetti, D. (2018). *Emoción y sentimientos: no somos seres racionales, somos seres emocionales que razonan*, editorial Planeta, Barcelona.
- López, V. F. (2013). *Técnicas eficientes para la recomendación de productos basadas en filtrado colaborativo*, <http://purl.org/dc/dcmitype/Text>, Universidade da Coruña.
URL: <https://dialnet.unirioja.es/servlet/tesis?codigo=38286>
- Macías álvarez, D. (2010). *Plataformas de enseñanza virtual libres y sus características de extensión: Desarrollo de un bloque para la gestión de tutorías en Moodle*, Thesis. Accepted: 2019-08-27T15:29:44Z.
URL: <http://148.202.167.116:8080/xmlui/handle/123456789/2671>
- Martín, B. R. y Sarmiento, C. A. C. (2019). *Entornos virtuales de aprendizaje: Posibilidades y retos en el ámbito universitario*, Ediciones de la Universidad de Castilla La Mancha. Google-Books-ID: 1wDIDwAAQBAJ.

- Mitchell, T. M. (1997). Does Machine Learning Really Work?, *AI Magazine* **18**(3): 11–11. Number: 3.
URL: <https://ojs.aaai.org/index.php/aimagazine/article/view/1303>
- Mohanty, D. (2018). Choosing Machine Learning Frameworks: Apache Mahout vs. Spark ML vs. Killer H2O.
URL: <https://www.linkedin.com/pulse/choosing-machine-learning-frameworks-apache-mahout-vs-debajani>
- Monroy, O. (2003). Algoritmos de Filtrado Colaborativo basados en Memoria, p. 103.
- Moodle, M. (2021). Acerca de Moodle - MoodleDocs.
URL: https://docs.moodle.org/all/es/Acerca_de_Moodle
- MoodleDocs, M. (2021). 36/Bloques - MoodleDocs.
URL: <https://docs.moodle.org/all/es/36/Bloques>
- Muñoz, W. M., León, A. M., Mora, G. V. y Molina, N. C. (2020). Aprendizaje autónomo en Moodle, *Journal of Science and Research* **5**(CININGEC): 632–652. Number: CININGEC.
URL: <https://revistas.utb.edu.ec/index.php/sr/article/view/1027>
- Nabi, Z. (2016). *Pro Spark Streaming: The Zen of Real-Time Analytics Using Apache Spark*, Apress. Google-Books-ID: ZfdjDAAAQBAJ.
- Nieto, S. M. G. (2010). Filtrado Colaborativo y Sistemas de Recomendación, p. 8.
- Nozawa, K. (2011). To Moodle or not to Moodle: Can It Be an Ideal e-Learning Environment?, p. 24.
- Oi, V., Zv, S., Mv, M., Mm, K., Li, K. y Tl, C. (2017). THE ORGANIZATION OF THE TEST CONTROL OF STUDENTSâ KNOWLEDGE IN A VIRTUAL LEARNING ENVIRONMENT MOODLE, p. 1.
- Olguín, G. E. M., Jesús, Y. L. d. y Celis, M. d. I. C. P. d. (2019). Métricas de similaridad y evaluación para sistemas de recomendación de filtrado colaborativo, *Revista de Investigación en Tecnologías de la Información: RITI* **7**(14): 224–240. Publisher: Grupo de Investigación SERTEL Section: Revista de Investigación en Tecnologías de la Información: RITI.
URL: <https://dialnet.unirioja.es/servlet/articulo?codigo=7242764>
- Omar, H. K. y Jumaa, A. K. (2019). Big Data Analysis Using Apache Spark MLlib and Hadoop HDFS with Scala and Java, *Kurdistan Journal of Applied Research* **4**(1): 7–14. Number: 1.
URL: <http://www.spu.edu.iq/kjar/index.php/kjar/article/view/265>

- Pajuelo Holguera, F. (2021). Sistemas de recomendación basados en filtrado colaborativo: aceleración mediante computación reconfigurable y aplicaciones predictivas sensoriales. Accepted: 2021-09-22T11:46:16Z.
URL: <https://dehesa.unex.es:8443/handle/10662/12476>
- Penchikala, S. (2018). *Big Data Processing with Apache Spark*, Lulu.com. Google-Books-ID: eodaDwAAQBAJ.
- PySpark, P. (2021). PySpark Documentation â PySpark 3.2.0 documentation.
URL: <http://spark.apache.org/docs/latest/api/python/>
- Rodríguez, C. R., Rodríguez, R. V., Moure, G. C. y Pérez, C. L. (2019). Personalización del Moodle mediante la integración de las tecnologías educativas de la web más empleadas en la educación superior, *ITECKNE: Innovación e Investigación en Ingeniería* **16**(1 (January - June)): 48–63. Publisher: Universidad Santo Tomás (USTA) Section: ITECKNE: Innovación e Investigación en Ingeniería.
URL: <https://dialnet.unirioja.es/servlet/articulo?codigo=7037307>
- Salloum, S., Dautov, R., Chen, X., Peng, P. X. y Huang, J. Z. (2016). Big data analytics on Apache Spark, *International Journal of Data Science and Analytics* **1**(3): 145–164.
URL: <https://doi.org/10.1007/s41060-016-0027-9>
- Samuel, A. L. (1959). Some Studies in Machine Learning Using the Game of Checkers, *IBM Journal of Research and Development* **3**(3): 210–229. Conference Name: IBM Journal of Research and Development.
- Schiavio, F., Bonetta, D. y Binder, W. (2020). Dynamic speculative optimizations for SQL compilation in Apache Spark, *Proceedings of the VLDB Endowment* **13**(5): 754–767.
URL: <https://doi.org/10.14778/3377369.3377382>
- Sierra, Y. (2019). Qué es Spark y cómo revoluciona al Big Data y al Machine Learning? - ADN Cloud. Section: Infraestructura.
URL: <https://blog.mdcloud.es/que-es-spark-big-data-y-machine-learning/>
- Significados, S. (2013). Significado de Diseño gráfico.
URL: <https://www.significados.com/disenio-grafico/>
- Spark, S. (2021). MLlib: Main Guide - Spark 3.2.0 Documentation.
URL: <https://spark.apache.org/docs/latest/ml-guide.html>

- Suárez, N. E. S. y Najjar, J. C. (2014). Evolución de las tecnologías de información y comunicación en el proceso de enseñanza-aprendizaje, *Revista vínculos* **11**(1): 209–220. Number: 1.
URL: <https://revistas.udistrital.edu.co/index.php/vinculos/article/view/8028>
- UNESCO, U. (2020). Las TIC en la educación.
URL: <https://es.unesco.org/themes/tic-educacion>
- Vázquez Basulto, R. C. (2019). *Bloque en moodle para el reconocimiento de emociones faciales de los estudiantes*, Thesis, Universidad Central âMarta Abreuâ de Las Villas. Accepted: 2019-10-02T13:39:22Z.
URL: <http://dspace.uclv.edu.cu:8089/xmlui/handle/123456789/11416>
- Vera Sancho, J. A. (2016). Modelo de aprendizaje para sistemas de recomendación, caso: Curso Programación Web, *Universidad Nacional de San Agustín de Arequipa* . Accepted: 2018-06-25T15:46:49Z Publisher: Universidad Nacional de San Agustín de Arequipa.
URL: <http://repositorio.unsa.edu.pe/handle/UNSA/6126>
- Vindel, A. C. (2000). ORIENTACIONES EN EL ESTUDIO DE LA EMOCION, p. 59.
- Yabuz, B. y Meng, X. (2014). Scalable Collaborative Filtering with Apache Spark MLlib.
URL: <https://databricks.com/blog/2014/07/23/scalable-collaborative-filtering-with-spark-mllib.html>
- Yeou, M. (2016). An Investigation of Studentsâ Acceptance of Moodle in a Blended Learning Setting Using Technology Acceptance Model, *Journal of Educational Technology Systems* **44**(3): 300–318. Publisher: SAGE Publications Inc.
URL: <https://doi.org/10.1177/0047239515618464>
- Zomaya, A. Y. y Sakr, S. (2017). *Handbook of Big Data Technologies*, Springer. Google-Books-ID: SsQ2DgAAQBAJ.