

**Sistema de recomendación de matrícula de cursos electivos para estudiantes  
de Ingeniería Electrónica e Ingeniería de Telecomunicaciones de la UNAD**

Adriana del Pilar Noguera Torres

Director

Santiago Rúa Pérez, Ph.D.

Universidad Nacional Abierta y a Distancia UNAD

Escuela de Ciencias Básicas Tecnología e Ingeniería ECBTI

Maestría en Gestión de Tecnología de Información

Yopal

2021

**Estudiante:**

**Nombre:** Adriana del Pilar Noguera Torres, IEO.<sup>1,2</sup>

**E-mail:** adriana.noguera@unad.edu.co

**Director:**

**Nombre:** Santiago Rúa Pérez, IEO, Ph.D.<sup>3</sup>

**E-mail:** srua@udem.edu.co

**Grupos de investigación:**

<sup>1</sup>Grupo de Investigación GIDESTEC, UNAD

<sup>2</sup>Semillero de Investigación Robotics, Control and Artificial Intelligence - RCAI

<sup>3</sup>Grupo de investigación de Arkadius, Universidad de Medellín

## Dedicatoria

Este documento de trabajo de grado se lo dedico a mi familia que gracias a su compañía, confianza y apoyo incondicional he podido crecer personal y profesionalmente.

A mis padres, Dionisia y Luis Eduardo, por ofrecerme su cariño, comprensión apoyo y atención en todo momento, que sin importar la distancia, siempre he sentido su compañía y apoyo incondicional, y sobre todo, su credibilidad en mi y mis capacidades.

A mi hermana Angela Yamile, por acompañar y apoyar mis sueños, ilusiones y locuras. Una mujer extraordinaria que siempre ha estado para animar mi caminar y no dejarme desfallecer, pues ha sabido ser fuente de energía cuando siento que no puedo más.

A mis sobrinos, Laura Vanessa, Laura Valentina y Gustavo Adrián, quienes se han convertido en mi principal motivo para buscar ser mejor persona cada día, recordando que con amor todo funciona mejor. Tres seres humanos extraordinarios, quienes a pesar de la distancia, buscan los medios para alegrar mis días y recargar mis deseos de seguir mis sueños.

## Agradecimientos

A Dios por regalarme la vida y darme la oportunidad de crecer en un hogar maravilloso que mis padres han sabido construir y guiar con amor, en el cual día tras día me han brindado su confianza y eterno apoyo, ese mismo que me ha acompañado en todos mis días, aportando a mi diario crecimiento como mujer y profesional, pero sobre todo, resaltando mis aciertos y celebrando mis triunfos.

A mi director Santiago Rúa Pérez por sus enseñanzas, por todo el apoyo brindado, la paciencia y las buenas sugerencias que me otorgó para la realización de esta propuesta, siendo el principal colaborador durante este proceso, evidenciando mediante su rol de Director, su amplio conocimiento y su espíritu de enseñanza, sus palabras que me invitan a crecer día a día como profesional, pero sobre todo, a creer en mi, en mis capacidades y a no dudar en ponerlas en práctica.

A la Universidad Nacional Abierta y a Distancia UNAD que con su equipo de docentes lograron el noble gesto humano de enseñar y tener siempre disposición para transmitir su conocimiento, atendiendo a su misión de contribuir a la educación para todos a través de la modalidad abierta, a distancia y en ambientes virtual de aprendizaje, mediante la acción pedagógica, la proyección social, el desarrollo regional y la acción comunitaria, la inclusión, la solidaridad, la investigación, la internacionalización y la innovación en todas sus expresiones, con el uso intensivo de las tecnologías, en particular de la información y las comunicaciones para fomentar y acompañar el aprendizaje autónomo, significativo y colaborativo.

Al cuerpo administrativo de la UNAD, en especial al personal del CEAD Yopal, cuya capacidad de servicio está siempre disponible y con calidad, para acompañar los procesos y procedimientos que enmarcan el ciclo de vida del estudiante Unadista con responsabilidad y respeto.

## Resumen

Los cambios en entornos competitivos de las organizaciones y las sociedades, generados por la irrupción de las TICs en diversos ámbitos de la sociedad; la globalización de las economías; la internacionalización de los mercados; y los desarrollos científicos y tecnológicos que han posibilitado y potencializado la libre movilidad de mercancías, personas y conocimientos a nivel mundial, han generado nuevos desafíos para la formación de las personas. Es importante resaltar que los cambios tecnológicos constituyen el motor que impulsa la exploración y búsqueda de nuevas opciones que posibiliten la educación de personas y la generación de condiciones para facilitar los procesos de aprendizaje en la llamada sociedad del conocimiento, dando respuesta a las necesidades de formación de comunidades y personas con dificultades para acceder a la formación tradicional con la educación a distancia, que ha evolucionado hacia la educación virtual soportado en formación E-learning (Castillo et al., 2017). E-learning usa tecnologías digitales para la generación de aprendizajes, conocida también como aprendizaje en medios electrónicos, basado en computadores, a través de internet o, basado en la web. La Universidad Nacional Abierta y a Distancia UNAD centra su formación en el modelo establecido por E-Learning, haciendo énfasis en que su misión es contribuir a la educación para todos a través de la modalidad abierta y a distancia utilizando como eje central lo que se conoce actualmente como ambientes virtuales de aprendizaje utilizando las tecnologías de la información y las comunicaciones para fomentar y acompañar el aprendizaje autónomo (Cardenas et al., 2017). De esta forma, el Modelo Pedagógico Unadista reconoce en su acción e-learning por su amplio potencial comunicativo e interactivo, y por la posibilidad de promover la construcción de sentidos y significados mediante el manejo de la información mediada por diferentes tipos de tecnologías.

Actualmente, para el procedimiento de matrícula, se han implementado filtros que le limitan los cursos a seleccionar cuando se trata de un estudiante nuevo y que no ingresa por convenio de homologación, pero en caso contrario, cuando el estudiante ingresa por convenio de homologación con el SENA, se presenta el inconveniente que no se ven relacionados los cursos homologados en el aplicativo de Registro y Control, y por lo tanto, en la plataforma de matrícula le presenta al estudiante una oferta completa llevando a que, en la mayoría de los casos, el estudiante matricule cursos homologados por acuerdos de convenio institucional.

Adicional a esto, estudiantes de últimos periodos de matrícula deben seleccionar los cursos electivos disciplinares y de profundización y se requiere de mayor acompañamiento de parte de la universidad aumentando así la atención in situ para asesoría de matrícula, con el fin de apoyar al estudiante para que seleccione cursos de acuerdo con su perfil académico y con el enfoque laboral en el cual se ve inmerso el estudiante.

Dentro del procedimiento de matrícula en periodos de 16 semanas se tiene la posibilidad de solicitar cambios de cursos, aplazamientos y/o cancelación de cursos, lo que se conoce en la UNAD como Solicitud de Novedades, estos procedimientos en los diferentes centros de la universidad generan un amplio número de solicitudes radicadas ante Registro y Control, siendo uno de los principales puntos críticos de atención en esta dependencia. Por otra parte, la Cadena de formación en Electrónica Telecomunicaciones y Redes (ETR) no cuenta con un sistema de recomendación de matrícula de cursos electivos que facilite al estudiante la selección de cursos electivos para dar cumplimiento con su respectiva malla curricular y de esta forma aportar en la disminución de los índices de novedades registradas en los diferentes centros a nivel nacional, donde solicitan cancelación, aplazamiento o cambio de cursos debido a matrícula errónea por causas diferentes, resaltando el desconocimiento de prerrequisitos, ya que actualmente la oferta de matrícula no tiene en cuenta todos los prerrequisitos de los cursos electivos.

Por lo antes expuesto, vale la pena evaluar qué grado de impacto se puede generar en estudiantes y docentes de los programas académicos mencionados de la Univer-

sidad Nacional Abierta y a Distancia UNAD y, a su vez, en los indicadores de registros de novedades de matrícula en los centros, con la implementación de un sistema basado en inteligencia artificial que genere recomendaciones de cursos electivos a los usuarios para generar matrículas oportunas de acuerdo a su avance en el programa de formación y gustos académicos, de esta manera se logra una propuesta que integre la tecnología con el sector académico que forma en ciencias de ingeniería, en concordancia con los nuevos enfoques tecnológicos, así como generar el beneficio de los futuros profesionales en ramas afines.

Este sistema propuesto le entrega al usuario una relación de cursos electivos a matricular teniendo en cuenta el historial académico de varios estudiantes que conforman la base de datos y con esta información, el sistema selecciona los que tendrían mayor afinidad de acuerdo con el perfil del estudiante usuario. Estos cursos corresponden a los que presentan un valor de métricas de medición con mejores características respecto a la matriz de Usuarios/Ítems que conforman el sistema. Esta matriz está conformada por 222 Usuarios (Estudiantes) y 104 Ítems (Cursos), información resultante de una base de datos inicial que contiene información de 253 estudiantes con un total de 11610 datos de cursos en su historial académico en un documento cuyo tamaño inicial era de 417 KB. Como se evidencia, dicha información es organizada y seleccionada teniendo en cuenta que contiene estudiantes matriculados por convenio institucional y eso genera calificaciones vacías en algunos de los ítems relacionados en la base de datos de Cursos. Por lo tanto, se evidencia la necesidad de optimizar la información de la base de datos para la creación del dataframe que comprende el sistema.

Adicional, en el desarrollo de la propuesta se generan dos algoritmos de sistemas de recomendación que aplican la técnica de filtrado colaborativo utilizando dos metodologías enfocadas en la factorización matricial, obteniendo así que el primer modelo se enfoca en la factorización matricial no negativa, mientras el segundo modelo propuesto se enfoca en la aplicación de la factorización matricial por SVD; siendo éste último el sistema que genera los mejores resultados en métricas de medición, optimización de tiempo y recursos de máquina y precisión en la recomendación a entregar al usuario del

sistema.

Es importante resaltar que la evaluación de los modelos propuestos se sustenta en la hipótesis que indica que la eficiencia de los algoritmos para los sistemas de recomendación por filtrado colaborativo toma como base la información relacionada en la matriz de de Usuarios/Ítems y la función de similitud utilizada para la creación del vecindario conformado por los vecinos más cercanos.

**Palabras Clave:**

Inteligencia artificial, sistemas de recomendación, aprendizaje automático



## Abstract

The changes in competitive environments of organizations and societies, generated by the emergence of ICTs in various areas of society; the globalization of economies; the internationalization of markets; and the scientific and technological developments that have made possible and potentiated the free mobility of goods, people and knowledge worldwide, have generated new challenges for the training of people. It is important to highlight that technological changes constitute the engine that drives the exploration and search for new options that make possible the education of people and the generation of conditions to facilitate learning processes in the so-called knowledge society, responding to training needs of communities and people with difficulties in accessing traditional training with distance education, which has evolved towards virtual education supported by E-learning training. E-learning uses digital technologies for the generation of learning, also known as learning in electronic media, based on computers, through the Internet or based on the web.

The Universidad Nacional Abierta y a Distancia UNAD focuses its training on the model established by E-Learning, emphasizing that its mission is to contribute to education for all through the open and distance modality using as a central axis what is currently known as virtual learning environments using information and communication technologies to promote and accompany autonomous learning. In this way, the Modelo Pedagógico Unadista recognizes in its e-learning action for its wide communicative and interactive potential, and for the possibility of promoting the construction of senses and meanings through the management of information mediated by different types of technologies. Currently, for the enrollment procedure, filters have been implemented that limit the courses to be selected when it is a new student who does not en-

ter by homologation agreement, but otherwise, when the student enters by homologation agreement with the SENA, the drawback is that the approved courses in the Registro y Control application are not related, and therefore, in the enrollment platform it presents the student with a complete offer, leading to the fact that, in most cases, the student enrolls courses approved by institutional agreement agreements.

Additionally, students who are in the last periods of enrollment must select the disciplinary electives and deepening courses and more support is required from the university, thus increasing the attention on site for enrollment advice, in order to support the student. to select courses according to their academic profile and the work approach in which the student is immersed. Within the enrollment procedure in periods of 16 weeks, there is the possibility of requesting changes of courses, postponements and/or cancellation of courses, which is known at the UNAD as Request for News, these procedures in the different centers of the university generate a large number of applications filed with the Registry and Control, being one of the main critical points of attention in this agency. On the other hand, the Cadena de formación en Electrónica Telecomunicaciones y Redes (ETR) does not have a system for recommending the enrollment of elective courses that facilitates the student's selection of elective courses to comply with their respective curricular mesh and thus contribute in the decrease in the indexes of novelties registered in the different centers nationwide, where they request cancellation, postponement or change of courses due to wrong enrollment for different reasons, highlighting the ignorance of prerequisites, since currently the enrollment offer does not have in counts all prerequisites for elective courses.

Due to the aforementioned, it is worth evaluating what degree of impact can be generated in students and teachers of the academic programs referred of the Universidad Nacional Abierta y a Distancia UNAD and, in turn, in the indicators of registration new enrollment in the centers, with the implementation of a system based on artificial intelligence that generates recommendations for elective courses to users to generate timely enrollments according to their progress in the training program and academic tastes, in this way a proposal that integrates technology is achieved with the aca-

ademic sector that trains in engineering sciences, in accordance with the new technological approaches, as well as generating the benefit of future professionals in related fields.

This proposed system provides the user with a list of elective courses to enroll taking into account the academic history of several students that make up the database and with this information, the system selects those that would have the greatest affinity according to the profile of the user student. . These courses correspond to those that present a value of measurement metrics with better characteristics with respect to the matrix of Users/Items that make up the system. This matrix is made up of 222 Users (Students) and 104 Items (Courses), information resulting from an initial database that contains information on 253 students with a total of 11,610 course data in their academic history in a document whose initial size was of 417 KB. As evidenced, said information is organized and selected taking into account that it contains students enrolled by institutional agreement and that generates empty grades in some of the related items in the Courses database. Therefore, the need to optimize the information in the database for the creation of the dataframe that comprises the system is evident.

Additionally, in the development of the proposal, two algorithms of recommendation systems are generated that apply the collaborative filtering technique using two methodologies focused on matrix factorization, thus obtaining that the first model focuses on non-negative matrix factorization, while the second proposed model focuses on the application of matrix factorization by SVD; The latter being the system that generates the best results in measurement metrics, optimization of time and machine resources, and precision in the recommendation to be delivered to the user of the system.

It is important to highlight that the evaluation of the proposed models is based on the hypothesis that the efficiency of the algorithms for the collaborative filtering recommendation systems is based on the information related in the Users/Items matrix and the similarity function. Used for the creation of the neighborhood made up of the closest neighbors.

**Keywords:**

Artificial intelligence, recommendation systems, machine learning

## Índice general

	Pág.
<b>Resumen</b>	<b>5</b>
<b>Abstract</b>	<b>9</b>
<b>Índice general</b>	<b>13</b>
<b>Índice de figuras</b>	<b>16</b>
<b>Índice de tablas</b>	<b>18</b>
<b>Introducción</b>	<b>20</b>
<b>Planteamiento del Problema</b>	<b>23</b>
<b>Contribuciones</b>	<b>29</b>
<b>Objetivos</b>	<b>32</b>
Objetivo General . . . . .	32
Objetivos Específicos . . . . .	32
<b>Estado del Arte</b>	<b>33</b>
Sistema de recomendación basado en contenido . . . . .	38

	14
Sistema de recomendación por filtrado colaborativo . . . . .	41
Sistema de recomendación basado en conocimiento . . . . .	44
Sistema de recomendación demográfico . . . . .	47
Sistema de recomendación híbridos . . . . .	48
<b>Metodología</b>	<b>53</b>
Características generales . . . . .	53
Ítems - Bases de datos . . . . .	54
Recomendador . . . . .	55
Librerías para el algoritmo . . . . .	57
Validación del diseño . . . . .	58
Aplicación factorización no negativa de matrices . . . . .	61
Descomposición de valores singulares (SVD) para factorización matricial . . . . .	64
<b>Desarrollo</b>	<b>66</b>
Datos del sistema de recomendación . . . . .	67
Algoritmo del sistema de recomendación . . . . .	74
Pre-Procesamiento de datos . . . . .	78
Procesamiento de datos . . . . .	81
Diseño del sistema de recomendación . . . . .	85
Diseño con factorización no negativa de matrices . . . . .	86
Diseño con SVD para factorización matricial . . . . .	96
<b>Análisis de resultados</b>	<b>107</b>

	15
Resultados . . . . .	107
Métricas de precisión de la predicción . . . . .	110
Capacidad de generar la recomendación . . . . .	112
Optimización de tiempos y recursos de máquina . . . . .	114
Discusión final . . . . .	114
<b>Conclusiones</b>	<b>117</b>
<b>Trabajos futuros</b>	<b>121</b>

## Índice de figuras

		Pág.
1.	Resumen procedimiento de selección de cursos y matrícula de estudiantes UNAD según procedimiento P-7-2. . . . .	24
2.	Características de un sistema de recomendación . . . . .	35
3.	Principales mecanismos de recomendación. . . . .	37
4.	Métodos de recomendación híbrida . . . . .	49
5.	Elementos del sistema de recomendación. . . . .	55
6.	Factorización no negativa de matrices NMF. . . . .	62
7.	Proceso de generación de Dataframe del SR. . . . .	67
8.	Modelo Entidad - Relación del SR . . . . .	75
9.	Procesamiento dataframe Historia Académica curso Electivo SISSU . . . . .	79
10.	Creación de dataframe Historia Académica por programa académico . . . . .	80
11.	Filtrado de datos de dataframe Historia Académica por programa académico . . . . .	81
12.	Procesamiento de datos del dataframe . . . . .	82
13.	Tamaño del arreglo matricial de dataframe ratings . . . . .	84
14.	Gráfica de comparación Matriz de Usuarios/Ítems y Matriz de predicción de acuerdo con el modelo diseñado . . . . .	88
15.	Comportamiento gráfico de respuesta a validación con coseno de similitud . . . . .	89



	17
16. Relación de recomendación para el ejemplo citado . . . . .	90
17. Programación para crear matrices de entrenamiento y de prueba . . . . .	90
18. Programación para evaluar las matrices de entrenamiento y de prueba . . . .	91
19. Gráfica de colores de comparación Matriz de Usuarios/Ítems y Matriz de predicción . . . . .	95
20. Datos estadísticos de la matriz de rating . . . . .	98
21. Procedimiento para crear el objeto SVD y entrenamiento del sistema . . . .	100
22. Procedimiento para selección automática de parámetros de validación cruzada	102
23. Lógica de programación para análisis de precisión con respecto a los datos de testeo en el modelo presentado . . . . .	103
24. Relación de recomendación para el ejemplo citado . . . . .	106
25. Comparación de datos obtenidos de RMSE para los tres modelos . . . . .	111

## Índice de tablas

		Pág.
1.	Programas seleccionados Cadena de formación ETR - UNAD y características	25
2.	Convenios de homologación vigentes programas seleccionados . . . . .	27
3.	Técnicas en sistemas de recomendación por filtrado colaborativo . . . . .	44
4.	Características base de datos Usuarios . . . . .	68
5.	Variables de la base de datos Usuarios . . . . .	69
6.	Características base de datos Mallas Curriculares . . . . .	70
7.	Variables de la base de datos Cursos . . . . .	71
8.	Características base de datos Historia Académica . . . . .	72
9.	Variables de la base de datos HA . . . . .	73
10.	Características de Dataframes usados . . . . .	76
11.	Relación de librerías y herramientas importadas en el algoritmo . . . . .	77
12.	Relación de características según dataframe . . . . .	83
13.	Relación algoritmos diseñados para el sistema de recomendación propuesto .	85
14.	Módulos específicos de librería Lenskit usados en el algoritmo . . . . .	93
15.	Características de configuración de implementaciones alternas de mínimos cuadrados de factorización matricial . . . . .	94
16.	Módulos específicos de librería Surprise usados en el algoritmo . . . . .	97

17.	Configuración de parámetros para la validación cruzada . . . . .	101
18.	Configuración de parámetros para <i>BaseLine</i> . . . . .	104
19.	Información de la base de datos del sistema . . . . .	108
20.	Información de la base de datos del sistema . . . . .	113

## Introducción

Hoy en día existe abundancia de información disponible en internet que muchas veces, en lugar de producir beneficios, afecta la satisfacción de los usuarios con respecto a sus intereses. Los sistemas web tienen cada vez más cantidad y variedad de información y los usuarios pueden recibir resultados ambiguos o pueden no recibir lo que desean al formular una consulta (Tekin and Van Der Schaar, 2015). Asimismo, con el propósito de hacer buen uso de esta gran cantidad de datos, se hace necesario gestionar los flujos de datos y atender la lectura y análisis de forma continua y en tiempo real.

Teniendo en cuenta que las plataformas de formación mediada por las TIC representan un gran flujo de información para el usuario, se ha evidenciado la necesidad de desarrollar aplicaciones que recomienden a los usuarios ofertas relacionadas con su perfil o con intereses previamente identificados y relacionados, llevando a lo que hoy en día se conoce como Sistemas de Recomendación (SR), que son herramientas que ayudan a focalizar al usuario ya sea por temática relacionada, por cursos previamente vistos, por búsquedas de tipo semántico, por foros en los que haya participado, por intereses de perfiles de usuario relacionados, entre otros. Los sistemas de recomendación ayudan a los usuarios a identificar la información más interesante y relevante en un grupo grande de información, evidenciando que forman parte de los reconocidos sistemas inteligentes que proporcionan sugerencias personalizadas sobre determinados temas o elementos, analizando las características de cada usuario y mediante un procesamiento de datos a través de un algoritmo, encontrar un conjunto de elementos que pueden resultar de interés al usuario (Caro Piñeres et al., 2011).

Los sistemas de recomendación han demostrado ser una herramienta valiosa pa-

ra ayudar a los usuarios en situaciones en las cuales se evidencia sobrecarga de información, cuyas tareas principales están relacionadas con el filtrado de flujos de información entrantes según las preferencias de los usuarios o para entregar elementos de interés adicionales en un contexto determinado, evidenciando actualmente que estos sistemas se han aplicado con éxito en gran variedad de dominios y, dentro de los elementos recomendables incluyen películas, libros, servicios de viajes y turismo, artículos de investigación, búsqueda o consultas y muchos más.

El sector académicos y/o educativo no es ajeno a estas situaciones y por eso aparece en la formación un concepto que enmarca un espacio virtual que facilita el aprendizaje orientado en la experiencia de capacitación a distancia y su metodología se sustenta en el uso de tecnologías digitales para la generación de aprendizajes, conocida también como aprendizaje en medios electrónicos, a través de internet, o E-Learning (Villaverde et al., 2010). La Universidad Nacional Abierta y a Distancia UNAD, en su Modelo Pedagógico Unadista (MPU), reconoce el e-learning por su amplio potencial comunicativo e interactivo, y por la posibilidad de promover la construcción de sentidos y significados mediante el manejo de la información mediada por diferentes tipos de tecnologías (Cardenas et al., 2017).

El estudiante Unadista al momento de matricular se encuentra con una amplia oferta para seleccionar los cursos por periodo académico evidenciando la ausencia de un sistema de filtrado enfocado en gustos particulares e historial académico de promedio de cursos y homologaciones aplicadas. La UNAD cuenta con convenios de homologación con algunas entidades, entre ellas el Servicio Nacional de Aprendizaje SENA, pero en el aplicativo de matrícula cuando un estudiante aplica un proceso de homologación por convenio, no se ven relacionados los cursos homologados, presentando al estudiante la oferta completa de cursos correspondientes al programa matriculado llevando a que, en la mayoría de los casos, el estudiante genere matrícula de cursos homologados por acuerdos de convenio. Adicional a esto, estudiantes de últimos periodos deben seleccionar los cursos electivos disciplinares y de profundización y se requiere de mayor acompañamiento de parte de la universidad aumentando así la atención in situ para asesoría

de matrícula.

Los avances tecnológicos con la irrupción de las TICs han facilitado y promovido el diseño de herramientas que permitan dar soluciones pertinentes y oportunas a los diferentes inconvenientes que se puedan presentar, y es por esta razón que los SR se han convertido en un aliado en el uso de información en línea y en especial en lo concerniente a la inmersión de las TICs en el desarrollo de los procesos formativos, facilitando a los usuarios de estas plataformas el acceso a información relacionada con su perfil y/o sus intereses.

## Planteamiento del Problema

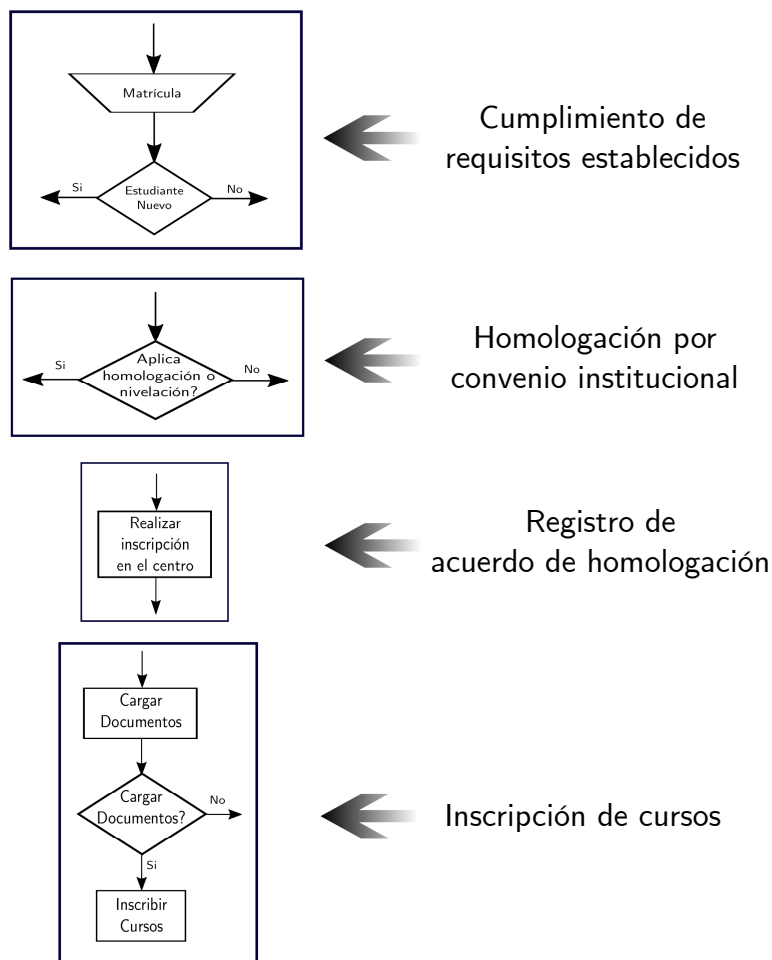
La sociedad ha dado respuesta a las necesidades de formación de las comunidades y personas con dificultades para acceder a la formación tradicional con la educación a distancia, la cual ha evolucionado hacia la educación virtual (Castillo et al., 2017). Teniendo en cuenta que la Universidad Nacional Abierta y a Distancia UNAD es la universidad pionera en América latina en implementar la educación basada en metodología E-learning, es importante señalar que esta metodología utiliza pedagógicamente escenarios basados en tecnologías digitales de información y comunicación para el desarrollo de procesos sistemáticos de formación, con entornos curriculares, didácticos, evaluativos, interacciones con medios, mediaciones y mediadores centrados en el aprendizaje y en el estudiante.

Para oficializar el ingreso a un estudiante nuevo o para el caso de estudiantes antiguos en proceso de matrícula, la universidad cuenta con un procedimiento establecido en el mapa de procesos de la entidad, el Sistema Integrado de Gestión SIG, correspondiente al Proceso de ciclo de vida del estudiante y se conoce como Procedimiento relacionado de inscripción y matrícula P-7-2 (Universidad Nacional Abierta y a Distancia - UNAD, 2020). Este procedimiento establece las condiciones a seguir para proceso de matrícula de estudiantes nuevos y antiguos y registro de homologaciones por convenio institucional, relacionando Figura 1 las actividades concernientes con la selección de cursos, proceso que corresponde al estudiante:

Una de las escuelas con que cuenta la UNAD se conoce como Escuela de Ciencias Básicas Tecnología e Ingeniería ECBTI y oferta los programas de Ingeniería Electrónica e Ingeniería de Telecomunicaciones, programas con cursos similares en algunos

Figura 1

*Resumen procedimiento de selección de cursos y matrícula de estudiantes UNAD según procedimiento P-7-2.*



*Nota.* Presentación de los procesos de verificación de requisitos para matrícula, homologación por convenio institucional, registro de acuerdo de homologación (cuando aplica) e inscripción de cursos como elementos base de procedimiento relacionado con inscripción y matrícula de estudiantes según procedimiento P-7-2 del Ciclo de vida del estudiante.

periodos de matrícula. Para Ingeniería Electrónica se tienen 170 créditos académicos que debe aprobar el estudiante, siendo 27 de ellos cursos electivos disciplinares y de línea de profundización y 4 para cursos electivos complementarios; para Ingeniería de Telecomunicaciones se tienen 158 créditos académicos, de los cuales 21 corresponden a cursos electivos disciplinares y de línea de profundización y 3 para cursos electivos com-



plementarios. En la Tabla 1 se relacionan las características mencionadas para los dos programas académicos adscritos a la escuela ECBTI.

Tabla 1

*Programas seleccionados Cadena de formación ETR - UNAD y características*

<b>Ingeniería Electrónica</b>	<b>Ingeniería de Telecomunicaciones</b>
Total: 170 créditos	Total: 158 créditos
Formación básica: 139 créditos	Formación básica: 134 créditos
Electivos: 27 créditos – 9 cursos	Electivos: 21 créditos – 7 cursos
Complementarios: 4 créditos	Complementarios: 3 créditos
Oferta cursos electivos: 22 cursos	Oferta cursos electivos: 20 cursos
4 líneas de profundización	3 líneas de profundización

*Nota.* Los datos registrados corresponden a un resumen de características de cada programa seleccionado de acuerdo con la información relacionada en las mallas curriculares de cada programa.

Como se evidencia en la Tabla 1, para el caso de Ingeniería Electrónica el estudiante debe seleccionar 9 cursos electivos pero el sistema le oferta como mínimo 22, mientras que para el caso de Ingeniería de Telecomunicaciones la oferta es mínimo de 20 cursos electivos cuando el estudiante debe seleccionar sólo 7 de ellos. Esto ha llevado a que, en algunos casos, los estudiantes de estos programas matriculen más cursos electivos de los requeridos según las mallas curriculares del programa y, a la vez, que se matriculen cursos electivos sin tener en cuenta prerrequisitos para el desarrollo correcto de las temáticas de cada uno.

Actualmente, el estudiante solicita el acompañamiento de los docentes en los diferentes centros de la universidad con el fin de recibir de ellos una asesoría sobre los cursos electivos a matricular, teniendo en cuenta su avance en la malla académica, su perfil académico de acuerdo con las calificaciones que relaciona en el historial académico y el perfil laboral e intereses laborales del estudiante. Este proceso requiere de tiem-

po de parte de los dos actores, dado que requiere de una verificación de avance en la malla académica del estudiante y la interacción con él para conocer sus intereses personales, profesionales y laborales para sustentar la sugerencia.

Uno de los convenios con que cuenta la universidad para continuidad académica de aspirantes es con el Servicio Nacional de Aprendizaje - SENA, ofertando la opción de profesionalización para egresados de dicha entidad en programadas tecnológicas avaladas por el Ministerio de Educación Nacional MEN, como también sucede con la Dirección Nacional de Escuelas de la Policía Nacional, la Escuela de Suboficiales Fuerza Aérea Andrés M. Díaz – ESUFA, la Institución Universitaria de Envigado – IUE, el Instituto Superior de Educación Rural – ISER. En la Tabla 2 se relacionan los convenios de homologación que aplican con los dos programas relacionados.

Cuando el estudiante ingresa con homologación por convenio institucional no se ven relacionados los cursos homologados en el aplicativo de matrícula lo que lleva a una presentación de oferta completa al estudiante llevando a que, en algunos casos, proceda a matricular cursos homologados por acuerdos internos. Adicional a esto, estudiantes de últimos periodos de matrícula deben seleccionar los cursos electivos disciplinares y de línea de profundización y se requiere de mayor acompañamiento de parte de la universidad aumentando así la atención in situ para asesoría de matrícula.

Por otra parte, la Cadena de formación en Electrónica Telecomunicaciones y Redes (ETR) no cuenta con un sistema de recomendación de matrícula de cursos electivos que facilite al estudiante la selección de cursos electivos para dar cumplimiento con su respectiva malla curricular y de esta forma aportar en la disminución de los índices de novedades registradas en los diferentes centros a nivel nacional, donde solicitan cancelación, aplazamiento o cambio de cursos debido a matrícula errónea por causas diferentes, resaltando el desconocimiento de prerrequisitos, ya que actualmente la oferta de matrícula no tiene en cuenta todos los prerrequisitos de los cursos electivos.

Los sistemas de recomendación están soportados en tecnologías de aprendizaje automático que permite filtrar el contenido a entregar en base a recomendaciones

Tabla 2

*Convenios de homologación vigentes programas seleccionados*

<b>Programa</b>	<b>Entidad Convenio</b>	<b>Número de Convenios</b>
Ingeniería Electrónica	Servicio Nacional de Aprendizaje - SENA	15
	Dirección Nacional de Escuelas de la Policía Nacional	2
	Escuela de Suboficiales Fuerza Aérea Andrés M. Díaz - ESU-FA	1
Ingeniería de Telecomunicaciones	Servicio Nacional de Aprendizaje - SENA	13
	Institución Universitaria de Envigado - IUE	1
	Instituto Superior de Educación Rural - ISER	1
	Dirección Nacional de Escuelas de la Policía Nacional	2
	Escuela de Suboficiales Fuerza Aérea Andrés M. Díaz - ESU-FA	1

*Nota.* Los datos registrados corresponden a un resumen de entidades con convenio de homologación para los programas seleccionados de acuerdo con la información relacionada en la página de la Universidad.

de contenido según un análisis de datos del usuario, de sus requerimientos, gustos y/o necesidades, evidenciando así que se trata de tendencias tecnológicas estratégicas con potencial de generar disrupción significativa y creación de nuevas oportunidades en ambientes digitales, como lo son inteligencia artificial, minería de datos, analítica de datos, entre otros (Collado Sánchez, 2014).

Por lo antes expuesto, en este contexto, vale la pena evaluar si es posible diseñar una herramienta, aplicando tecnologías de información, que facilite al estudiante la selección de cursos electivos en el proceso de matrícula de su proceso de formación sin necesidad de acudir al apoyo de un docente en alguna sede de la Universidad, teniendo en cuenta su avance en el programa de formación matriculado y sus gustos académicos.

## Contribuciones

El desarrollo del presente documento genera los siguientes aportes:

- *Capítulo 2 - Metodología:* Teniendo en cuenta el estudio de la literatura relacionada con sistemas de recomendación y sus aplicaciones con enfoques académicos, se relaciona la metodología utilizada para el desarrollo propuesto, los métodos implementados y sus diferentes enfoques probados en el algoritmo propuesto, buscando la forma de utilizar el algoritmo que genera una respuesta más óptima de acuerdo con el sistema a implementar.
- *Capítulo 3 - Desarrollo:* En este capítulo se relacionan los algoritmos obtenidos del desarrollo de la propuesta, documentando las etapas que lo conforman con el fin de garantizar un mayor entendimiento del sistema y una mejor interacción en el momento de su uso. De igual forma se relacionan las evidencias de funcionamiento del algoritmo propuesto con pruebas realizadas y su respectiva documentación.
- *Capítulo 4 - Análisis de resultados:* Finalmente, se presenta el análisis de los resultados obtenidos de acuerdo con la hipótesis planteada con el fin de verificar la validación del funcionamiento y aplicabilidad del sistema propuesto.

Paralelo al desarrollo de esta propuesta, se generaron productos que permitieron la socialización del trabajo realizado y la presentación de algunos de los resultados obtenidos, estos productos corresponden a:

- *Participación en eventos académicos internos que visibilicen la producción:* En el año 2019 se participa en el VI Encuentro Interzonal de Investigación - Perspectivas de la visibilidad e impacto de la comunicación científica presentando la ponencia como propuesta de desarrollo denominada "*Sistema de recomendación de matrícula*

*para estudiantes de Ingeniería Electrónica e Ingeniería de Telecomunicaciones de la UNAD soportado en tecnologías disruptivas".*

- *Participación en eventos académicos internos que visibilicen la producción:*  
En el año 2020 se participa en la Feria de Proyectos 16-01 2020 con la ponencia denominada *"Sistema de recomendación de matrícula para estudiantes de Ingeniería Electrónica e Ingeniería de Telecomunicaciones de la UNAD soportado en tecnologías disruptivas"* con presentación de avances del desarrollo.
- *Participación en eventos académicos internos que visibilicen la producción:*  
Para el año 2020 se participa en el VII Encuentro Interzonal de Investigación IV Encuentro Internacional de E-Investigación - Apropiación, transferencia de conocimiento y transformación de territorios: spinoff y otros escenarios de investigación e innovación con la ponencia denominada *"Diseño de una herramienta de software para el estudio de avance de malla de ingenierías Cadena de formación ETR UNAD"* en modalidad de póster por presentación de resultados.

Del desarrollo de la propuesta, se generaron dos algoritmos con sus respectivos manuales con proceso de registro de software ante el DNDA y corresponden a:

- *Productos de desarrollo tecnológico e innovación:* Software denominado *Estudio de avance de malla Ingenierías ETR V1*, con registro DNDA código 13-82-328 del 26 de noviembre de 2020.
- *Productos de desarrollo tecnológico e innovación:* Software denominado *SRMat - V1* en trámite ante DNDA.

De igual manera se cuenta con la generación de propuestas de artículos para sometimiento a publicación denominados:

- *Productos de generación de nuevo conocimiento:* Review of recommender systems for E-learning: a systematic approach.
- *Productos de generación de nuevo conocimiento:* Collaborative filter-based recommendation systems for the selection of elective courses in Engineering.

Los productos académicos asociados a la tesis (Ponencias, registro de software y

artículos de revistas) se reportan al grupo de investigación GIDESTEC, lo que refuerza su posición dentro del Sistema Colombiano de Ciencia, Tecnología e Innovación.

## Objetivos

### Objetivo General

Diseñar un sistema de recomendación de matrícula de cursos electivos para estudiantes de Ingeniería Electrónica e Ingeniería de Telecomunicaciones de la UNAD.

### Objetivos Específicos

- Seleccionar la técnica de recomendación a utilizar teniendo en cuenta la revisión del estado del arte de técnicas de sistemas de recomendación.
- Diseñar el algoritmo del sistema de recomendación de matrícula de cursos electivos para los programas seleccionados teniendo en cuenta la información de la base de datos y las características de la técnica de recomendación seleccionada.
- Evaluar el desempeño del sistema diseñado analizando parámetros de precisión de la recomendación.



## Estado del Arte

Día tras día la sociedad está expuesta a una cantidad de información que aumenta más rápido que la misma capacidad que se tiene actualmente para procesarla, lo que evidencia que es el momento de aplicar tecnologías en desarrollos que promuevan el procesamiento y el análisis de datos en medio de esta sobrecarga de información. Los sistemas de recomendación surgen así con el propósito de facilitar la toma de decisiones en temas/dominios en los que las posibilidades de elección son muchas y variadas, actúan sugiriendo buenos productos y/o servicios, bien sea para la compra o el consumo.

Actualmente, los sistemas de recomendación han sido aplicados en diferentes áreas, recomendadores de películas, música, libros, turismo, entre otros, y el sector académico no ha sido la excepción, en donde se evidencian investigaciones enfocadas en el desarrollo de herramientas de recomendación para la creación de cursos, recursos académicos y actividades de aprendizaje, así como también se encuentran desarrollos con recomendadores que sugieren a los instructores las modificaciones más apropiadas para mejorar la eficiencia de los sistemas educativos web. Estos sistemas utilizan información de los usuarios, sus preferencias, previamente identificadas, para generar predicciones respecto a los elementos que generen mayor afinidad con la información del usuario, por medio de la aplicación de algoritmos en diferentes lenguajes de programación.

El objetivo fundamental de un sistema de recomendación es brindar al usuario información relacionada con su perfil, sus preferencias y necesidades, previo filtrado de la información con el fin de seleccionar aquella que se relacione directamente con el usuario para generar la recomendación final (Vera et al., 2017). Teniendo en cuenta las técnicas de recomendación aplicadas en los algoritmos de sistemas de recomendación

se pueden generar como resultado sugerencias al usuario con información que no se había contemplado anteriormente, generando una mejor satisfacción al usuario que usa el sistema (Burke, 2017). Según establece Wang, los sistemas de recomendación tienen como tarea principal seleccionar ciertos objetos, de acuerdo con los requerimientos del usuario, teniendo en cuenta información previamente recolectada y almacenada para ser analizada según los requerimientos del usuario (Wang et al., 2019).

Teniendo en cuenta los diferentes desarrollos de sistemas de recomendación existentes, el punto de partida de la mayoría de estos sistemas está relacionado directamente con la ponderación que otros usuarios han asignado previamente a la información que conforma la recomendación, como es el caso de los sistemas colaborativos o sistemas de filtrado colaborativo (Bobadilla et al., 2009). También se tienen sistemas en los cuales se usa información distinta a las ponderaciones mencionadas, como pueden ser los perfiles de preferencias de los usuarios para la toma de decisiones, las características particulares de los productos, información de los usuarios de carácter demográfico, entre otras. Y los mismos desarrollos investigativos han involucrado sistemas de recomendación que usan diferentes combinaciones de las técnicas de recomendación existentes, presentando a los usuarios recomendaciones con mayor nivel de confianza por la aproximación en las sugerencias generadas (Ansari et al., 2017; Wang et al., 2019).

Como se ha evidenciado, la base de todo sistema de recomendación está en la información recolectada que está relacionada con los intereses y/o preferencias de los usuarios, información que debe ser analizada como fundamental para el desarrollo del algoritmo. Teniendo en cuenta la forma de recolección de esta información, estos datos pueden ser clasificados como datos explícitos cuando es el usuario quien previamente asigna una ponderación directa sobre el producto (Salehi, 2013), y datos implícitos, cuando se obtiene dicha ponderación aproximada partiendo de la interacción del usuario con el sistema (Salehi et al., 2012).

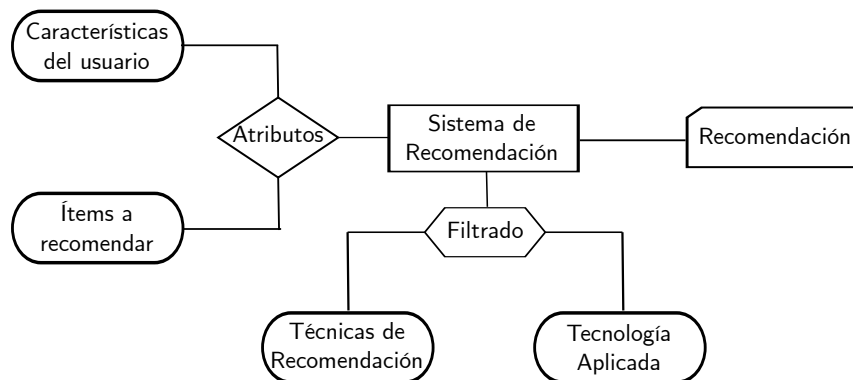
La tendencia de los nuevos desarrollos investigativos en el ámbito de los sistemas de recomendación tiene como objetivo principal que el sistema llegue a conocer

al usuario sin que éste le tenga que dar información explícita sobre sus gustos y preferencias, por lo que estos desarrollos se enfatizan en el análisis de diferentes parámetros que entran en juego en la interacción del usuario con el dispositivo, encontrando hoy en día algoritmos de desarrollos de sistemas de recomendación que, con aspectos como el tiempo de permanencia en el contenido, el número de clicks, los comentarios sobre un contenido o recomendar el contenido a otros usuarios pueden evidenciar el interés de los usuarios por determinado contenido.

El proceso de recomendación tiene en cuenta no sólo las características y preferencias del usuario sino también las características de las diferentes técnicas de recomendación usadas en el algoritmo implementado. Por lo tanto, el filtrado de la información recolectada representa la clave del funcionamiento de estos sistemas, evidenciando así que los sistemas de recomendación generan una sugerencia al usuario en base a información personalizada de cada uno con relación a información recolectada de otros usuarios. (ver Figura 2)

Figura 2

*Características de un sistema de recomendación*



*Nota.* Se relaciona el diagrama general de un sistema de recomendación, resaltando los atributos de Usuario/Ítem y la técnica de recomendación.

La estructura base de un sistema de recomendación o el *pipeline* se sustenta en cuatro etapas:

- *Recolección de información:* El primer paso o primera etapa de estos sistemas se basa en recolectar los datos de los usuarios, como sus preferencias, características,

gustos, entre otros, es decir, la información relevante e importante que aporte al algoritmo.

- *Almacenamiento de la información:* Una vez recolectada la información que aporta al funcionamiento del sistema, debe almacenarse en un formato que permita el acceso a esta y, a su vez, modificarla en caso que se requiera. Por lo general el formato usado es en tablas de excel o formato csv.
- *Analizar la información:* Esta etapa hace referencia a la lógica de programación usada en el algoritmo y tiene relación con el tipo de recomendación a aplicar. En algunos casos, se requiere ajustar el tipo de información que se tiene para poder usarla en el algoritmo propuesto, teniendo en cuenta que se pueden utilizar ecuaciones o análisis de correlación.
- *Filtrar la información:* En esta etapa se aplican técnicas que relacionen las características de los usuarios, con características específicas de acuerdo a lo que el sistema desea recomendar. El filtrado puede aplicarse por medio de funciones matemáticas, información suministrada por el usuario, ranking, entre otras.

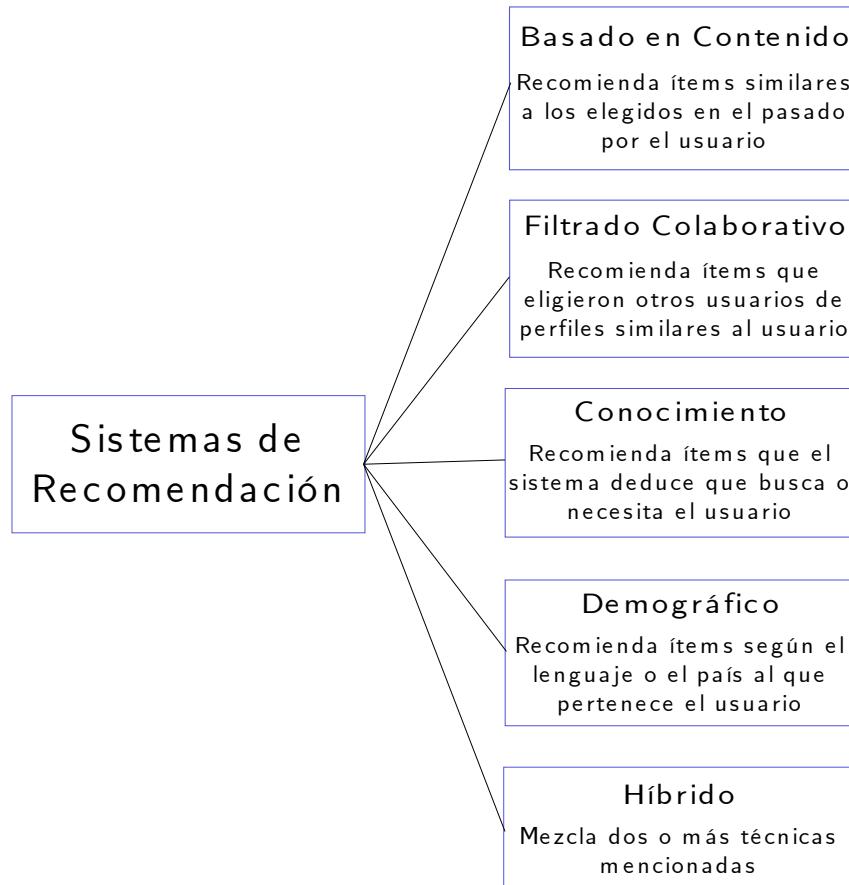
Actualmente, los sistemas de recomendación han implementado mejoras en sus desarrollos mejorando sus algoritmos reduciendo su problemática al predecir la valoración que el usuario podría dar a una serie ítems que todavía no ha puntuado, teniendo como base las puntuaciones que otros usuarios asignaron en el pasado, como sucede con los sistemas colaborativos o sistemas de filtrado colaborativo, aunque también puede obtenerse usando otra clase de información distinta a las valoraciones mencionadas, como pueden ser las características específicas de los productos, los perfiles de preferencias de los usuarios a la hora de tomar decisiones, información demográfica de los usuarios, etc., y por supuesto, usar combinaciones de diversas aproximaciones entre las existentes.

Roin Burke, en su artículo denominado *Hybrid Web Recommender Systems* (Burke, 2007), relaciona cuatro clases diferentes de técnicas de recomendación basadas en la fuente de conocimiento que corresponden a: sistemas recomendación colaborativos, basados en contenido, demográfico, y basado en conocimiento. Actualmente, varios de-

sarrollos investigativos relacionados con los sistemas de recomendación, han llevado a complementar estas técnicas de recomendación mencionadas generando sistemas híbridos, mejorando las características propias de cada técnica y reduciendo las desventajas que presenta cada una.

Figura 3

*Principales mecanismos de recomendación.*



*Nota.* Se relacionan los cinco mecanismos de recomendación de mayor aplicación en desarrollos existentes, especialmente aquellos con enfoque académico.

En los mecanismos de recomendación relacionados en la Figura 3, tanto simples como híbridos, se destaca que todos utilizan algoritmos para llevar a cabo las recomendaciones, los algoritmos pueden ser redes bayesianas, algoritmos basados en reglas, árboles de decisión, clustering, redes neuronales, etc. Además, existe una rama dentro de la inteligencia artificial conocida como aprendizaje automático, cuyo objetivo es desarrollar técnicas que permitan garantizar el funcionamiento esperado de los sistemas de

recomendación.

A continuación, se relacionan las características principales de cada técnica y algunas de las desventajas que pueden presentar.

### **Sistema de recomendación basados en contenido**

El sistema de recomendación basado en contenido tiene como característica principal la generación de recomendaciones teniendo como base la información que corresponde a las características asociadas con los productos y las ponderaciones o valoraciones que un usuario les ha dado previamente. Otra de las características que representan este tipo de sistemas es que tratan la recomendación como un problema de clasificación específico del usuario e implementan en su algoritmo una clasificación de los gustos del usuario según las características específicas del producto (Ruiz-Iniesta et al., 2010). Por tanto, este tipo de recomendación se basa en la información recolectada y clasificada de las necesidades que han relacionado o descrito los usuarios y sugiere la recomendación sobre la base de las características previamente clasificadas de cada uno de los ítems. Finalmente, el proceso de filtrado garantiza una coincidencia entre los atributos del perfil del usuario con los atributos de los ítems a recomendar (Hdioud et al., 2012).

En estos sistemas, de forma automática, se recomiendan elementos que el usuario ya había adquirido o consumido en el pasado basándose en la información que el usuario ya había registrado previamente en su perfil, de tal manera que cuando un usuario encuentra interesante un elemento, el sistema le sugiere contenidos similares dependiendo de los valores que se asignan a las palabras que describen el contenido del elemento o en base a las características del elemento dentro del perfil del usuario, es decir, se accede a su perfil de usuario para recomendarle otros elementos que pueden ser de su agrado.

En la etapa de filtrado de la información, los sistemas de recomendación basados

en contenido tienen el producto como base de la predicción, en lugar de tener al usuario. Lo que implica que utilizan las características del artículo como su marca, precio, calificaciones, tamaño, categoría, etc., para generar las recomendaciones. Dependiendo de las características de los datos se pueden construir modelos de usuarios basados en sus preferencias. Este es un problema de clasificación o clustering dependiendo del tipo de estructura y dominio, lo que lleva a diseñar algoritmos que permitan diferenciar los tipos de usuarios y construir un modelo que permita agrupar los usuarios de acuerdo con sus preferencias.

Como inconveniente, esta técnica de recomendación requiere de suficiente información para proponer una recomendación fiable y no todos los usuarios presentan disponibilidad para puntuar o asignar un valor a la información que se les ofrece, por tanto, no siempre se garantiza que el sistema ofrezca información relevante y pertinente de acuerdo con el perfil del usuario. Los sistemas de recomendación que usan esta técnica sufren algunos problemas que también son presentes en los sistemas que utilizan técnicas basadas en un filtro colaborativo, como el problema de un nuevo usuario y el de dispersión de ratios. Pese a esto, la principal desventaja de esta radica en que los sistemas ofrecen ítems similares a los provistos y no siempre se garantiza que el sistema ofrezca información relevante y pertinente de acuerdo con el perfil del usuario (Albatayneh et al., 2018).

Si se carece de suficiente información, se recomienda al usuario ítems relacionados con los recursos que él mismo recomienda, de acuerdo con su historial, por tanto, en estos sistemas de recomendación no se pueden realizar recomendaciones basados en serendipia (Kotkov et al., 2016). En contraste a sus desventajas, se tiene que este sistema recomienda recursos respecto a su contenido, es decir, de manera objetiva sin tener en cuenta calificaciones subjetivas de otros usuarios o perfiles, respecto a particularidades de cada contenido (Antony Rosewelt and Arokia Renjit, 2020). Este sistema no predice la valoración que pueda dar el usuario a un ítem, sino que trabaja directamente con la puntuación que el usuario le asigna al mismo, evidenciando de esta manera la preferencia directa del usuario con los ítems relacionados, por tanto, los ítems se rela-

cionan por su descripción textual generando correspondencia entre los términos asociados con la descripción, o por su representación en atributos, como características adicionales del ítem, representado de la siguiente forma:

$$\text{Utilidad } (u, p) = \text{Puntuación (Perfil}(u) \cdot \text{Contenido } (p)) \quad (1)$$

Los algoritmos que se elaboran para este tipo de sistemas tienen en sus líneas de código los referentes de la implementación de técnicas de clasificación, de acuerdo a las relacionadas a continuación:

- *Modelos en Espacios Vectoriales*: Se obtiene un vector asociado a cada usuario. Se obtiene similitud del usuario con un vector de contenidos de cada ítem. Se recomienda los N productos con mayor similitud (Dierk, 1972; Tzung-Pei Hong and Jyh-Bin Chen, 1999; Symeonidis et al., 2007).

Modelo Booleano básico

$$\cos(u, v) = \frac{\sum_{i=1}^* u_i v_i}{\sqrt{\sum_{i=1}^N (u_i)^2} \cdot \sqrt{\sum_{i=1}^N (v_i)^2}} \quad (2)$$

En donde  $u_i$  y  $v_i$  representan dos usuarios diferentes pero con características en común a los cuales se les analiza la similitud entre ellos.

Modelo Booleano con Ponderación Basada en TF-IDF

$$\cos(u, v, w) = \frac{\sum_{i=1}^N u_i v_i w_i}{\sqrt{\sum_{i=1}^N (u_i)^2} \cdot \sqrt{\sum_{i=1}^N (v_i)^2}} \quad (3)$$

Para este caso, la variable  $w_i$  representa la ponderación dada a las características relacionadas por cada usuario  $u_i$  y  $v_i$ .

- *Redes Bayesianas*: Se basa en crear un modelo probabilístico que relaciona los metadatos del contenido mediante un grafo. Una vez establecido el grafo se realiza inferencias basadas en probabilidades condicionadas.
- *Modelo probabilista*: Modelo que representa una secuencia de variables en el tiempo.
- *Redes de Eventos*: Modelo probabilista que relaciona las Redes Probabilísticas de Tiempo Discreto y las Redes Bayesianas de Nodos Temporales en donde cada va-



lor de la variable representa el tiempo en el que cierto evento ocurre (Jensen and Nielsen, 2001; de Campos et al., 2010).

- *Redes Neuronales*: Se obtiene un vector de características de los metadatos del ítem. La red predice la calificación de acuerdo a estas características y es entrenada para cada usuario. Los modelos basados en redes neuronales se fundamentan en la relación de semejanza que permite evaluar si la información de entrada es representada, según el parámetro de vigilancia, de la mejor manera por la categoría ganadora (Christakou et al., 2007; Tarazona et al., 2013).

### **Sistema de recomendación por filtrado colaborativo**

El filtrado colaborativo representa la técnica más utilizada en los algoritmos de sistemas de recomendación y su funcionamiento se relaciona con la generación de recomendaciones al usuario por medio de la información sobre los perfiles de calificación para diferentes usuarios únicamente. En estos sistemas se crean vecindarios teniendo como referencia el historial de calificación del usuario con respecto al historial de pares y en relación a los mismos ítems ponderados para recomendar ítems al usuario de este vecindario (Goldberg et al., 1992).

Un sistema de recomendación basado en filtrado colaborativo tiene como punto de partida la calificación que un usuario ha dado previamente a los ítems utilizados o que requiere analizar de un usuario específico, generando de esta forma un indicativo de referencia respecto a la necesidad de uso o experiencia favorable o no de la información que se relacione para su ponderación. Estas ponderaciones se recolectan y filtran, según los procesos que se realizan para los sistemas de recomendación, con el fin de generar un amplio campo de referencia para la recomendación específica, permitiendo así que otros usuarios del sistema sean beneficiados de esta información y se tome como base para la creación de vecindarios y/o universos. Este procedimiento evidencia una ventaja sobre los sistemas de recomendación basado en contenido, ya que tienen en cuenta

ítems que no estaban contemplados inicialmente por el usuario en la selección o relación de preferencias, pero que potencialmente pueden ser de su gusto por afinidad con el vecindario. Pero a su vez, también se presenta una desventaja y se evidencia cuando se tiene un usuario nuevo y el sistema no conoce las preferencias de este, sobre todo cuando no se cuenta con información relacionada a ponderaciones previas o interacciones del usuario con ítems relacionados en el sistema (Wei et al., 2017).

Se sugiere que el algoritmo a implementar en base a esta técnica de recomendación tenga dos etapas: primero buscar la forma de seleccionar los usuarios con los mismos gustos o preferencias según los patrones de evaluación o ponderación para clasificación de los ítems; y de esta forma, se procede a la segunda etapa que busca utilizar estas valoraciones para estimar el resultado final y generar la recomendación al usuario (Miller et al., 2004). En la literatura se pueden encontrar técnicas basadas en la correlación entre usuarios o productos (Shardanand and Maes, 2015) y técnicas más complejas basadas en el aprendizaje automático (Marlin, 2004).

Los algoritmos a implementar usando esta técnica de recomendación se dividen en (Zhuang et al., 2018):

- *Basados en modelos*: En esta técnica se realizan las recomendaciones teniendo en cuenta la construcción de un modelo basado en las ponderaciones asignadas previamente por los usuarios. Estos modelos se crean a partir de diferentes algoritmos y enfoques, teniendo prelación a aquellos basados en el aprendizaje automático, entre las que se destacan las redes bayesianas, el encasillamiento, los enfoques basados en reglas, características latentes, entre otros. Estos sistemas predicen o calculan la posible elección del usuario activo teniendo en cuenta la información recolectada de los usuarios y sus respectivas ponderaciones (Nicholas and Francis, 2019).
- *Basados en memoria*: Esta técnica toma como referencia para su recomendación la matriz de ponderaciones de datos completos recolectados de la información requerida y ponderada por los usuarios en relación a los ítems presentados y de esta forma generar estimaciones, en algunos casos utilizan relaciones estadísticas de se-

mejanza y de esta forma determinan el grupo de usuarios que tienen una mayor similitud con el usuario activo (Cai et al., 2020).

El filtrado colaborativo utiliza 4 métodos con los cuales puede hallar la métrica de similitud para el diseño del algoritmo que genera la recomendación y se relacionan a continuación:

- *Cálculo de K-Vecinos*: Se obtiene una relación de usuarios de mayor afinidad entre si teniendo como base los resultados de la métrica de similitud utilizada e implementada en ese algoritmo, esto con el fin de utilizar estos usuarios como referencia en la recomendación. A estos usuarios se les conoce como K-Vecinos (Krzywicki et al., 2015).
- *Cálculo de similitud entre usuarios*: Para realizar este cálculo se usan métricas de similaridad ya existentes, como lo son la diferencia cuadrática media, la correlación de Pearson o el coseno de similitud, entre otras técnicas. El valor obtenido de este cálculo debe oscilar entre 0 y 1 (Krzywicki et al., 2015). A continuación se encuentran las ecuaciones correspondientes para obtener los modelos matemáticos según sea la métrica a usar:

- *Correlación de Pearson*:

$$\text{sim}(x, y) = \frac{\sum_{i \in B_{xy}} (r_{x,i} - \bar{r}_x) \cdot (r_{y,i} - \bar{r}_y)}{\sqrt{\sum_{i \in B_{xy}} (r_{x,i} - \bar{r}_x)^2 \cdot \sum_{i \in B_{xy}} (r_{y,i} - \bar{r}_y)^2}} \quad (4)$$

La relación matemática del numerador de la ecuación representa la covariancia entre dos usuarios, es decir, hace referencia al grado de variación conjunta de dos usuarios respecto a sus medias; y los datos del denominador representan la desviación estándar de los datos correspondientes a los dos usuarios, es decir, la variación o la dispersión de los datos numéricos que representan cada usuario.

- *Coseno de similitud*:

$$\text{sim}(x, y) = \frac{\sum_{i \in B_{xy}} r_{x,i} \cdot r_{y,i}}{\sqrt{\sum_{i \in B_{x,y}} r_{x,i}^2} \sqrt{\sum_{i \in B_{x,y}} r_{y,i}^2}} \in [0, 1] \quad (5)$$

En este caso, hace referencia al valor de la medida que considera la simili-

tud entre pares de características o de usuarios, analizando la similaridad evaluando en características determinadas.

- *Factorización de matriz*: Este método permite la descomposición de la matriz de datos en matrices más pequeñas con el fin de operarlas de una forma más sencilla y poder determinar de esta forma la relación que existe entre un usuario y los ítems identificados.
- *Cálculo de predicciones por ítem*: En este paso se asignan valores a los ítems que el usuario no ha votado aún, tomando como punto de partida base las valoraciones o ponderaciones asignadas por los K-Vecinos que se hallaron previamente en el algoritmo.

Teniendo en cuenta estas características de los sistemas de recomendación basados en filtrado colaborativo, en la Tabla 3 se relaciona una tabla resumen de su clasificación y las técnicas más usadas en cada una:

Tabla 3

*Técnicas en sistemas de recomendación por filtrado colaborativo*

<b>Basados en memoria</b>	<b>Basados en modelo</b>
K-Vecinos	Algoritmos Genéricos
JMSD	Redes Neuronales
MSD	Lógica Difusa
SVD	SVD
Correlación	
Coseno de similitud	

*Nota.* Los datos son producto de revisión de estado del arte realizado por el autor.

## **Sistema de recomendación basado en conocimiento**

El sistema de recomendación basado en conocimiento centra su funcionamiento en la generación de sugerencias de productos o ítems al usuario activo teniendo en

cuenta inferencias sobre las preferencias de un usuario según la información recolectada previamente y que representa el perfil del usuario. En algunos casos, esta información será funcional explícita sobre cómo ciertas características de un producto logran un alto grado de satisfacción respecto a las necesidades de los usuarios (Tarus et al., 2018). Esta información previamente recolectada sobre los usuarios, sus perfiles y preferencias se utiliza para realizar un análisis que permita determinar, dentro de todas las posibilidades, cuál ítem cumple con los requerimientos identificados en base a las necesidades relacionados con el usuario, muchas veces sin tener en cuenta las ponderaciones o valoraciones que el usuario pueda otorgar a cada ítem (Kiyoki et al., 1988).

Como se ha evidenciado, esta técnica de recomendación tiene en cuenta aquella información que sirve para representar directa o indirectamente el nivel de satisfacción de un usuario respecto a un ítem específico y genera una relación de inferencia teniendo en cuenta las preferencias relacionadas por el usuario y las necesidades que le pueda representar un ítem determinado, para finalmente generar la recomendación al usuario (Burke, 2017). Para su correcto funcionamiento, este sistema de recomendación necesita una buena cantidad de información, en algunas ocasiones, es información recolectada partiendo de registros previos de los usuarios, y en este caso, aunque se requiere de información previa como punto de partida para el funcionamiento del algoritmo del sistema, no se generan recomendaciones relacionadas con lo que otros usuarios han relacionado en algún momento como preferencias personales, todo lo contrario, el resultado de la recomendación se relaciona directamente con las necesidades del usuario que requiere la recomendación, teniendo como resultado recomendaciones con alto nivel de similitud comparadas con la información recolectada del perfil de los usuarios y, en muy raras ocasiones, se recomiendan ítems diferentes (Blanco-Fernandez et al., 2008).

Estos sistemas han sido considerados como sistemas heterogéneos tomando como base la interacción del sistema y su algoritmo con el usuario (Viappiani et al., 2008), a continuación se relacionan algunas de estas interacciones:

- Sistemas que le facilitan al usuario la reorganización de las recomendaciones según

sus preferencias (Chen et al., 2009).

- Sistemas que posibilitan la interacción del usuario con el propósito de generar un listado de los ítems o productos de acuerdo con sus preferencias (Rodríguez et al., 2010).
- Estos sistemas solicitan al usuario que relacione o liste aquellos productos que tengan relación o mayor grado de similitud con lo que está buscando, y le regresan al usuario como resultado una lista de ítems identificados o relacionados como similares según la recomendación o sugerencia para el usuario (Martinez et al., 2008).
- En la interacción con el usuario le requieren únicamente ponderaciones y/o puntuaciones sobre los ítems que relaciona y realiza el proceso de inferencia teniendo en cuenta esta información (Blanco-Fernández et al., 2011).
- Estos sistemas le permiten al usuario establecer relaciones entre las características de los productos que previamente ha señalado y sus preferencias (Bridge and Ferguson, 2002).

Teniendo en cuenta los desarrollos adelantados sobre estos sistemas de recomendación y la literatura existente respecto a los mismos, se ha evidenciado que existen cuatro enfoques que relacionan el conocimiento en los algoritmos de estos sistemas de recomendación, en primer lugar se tiene un enfoque en donde se utiliza el conocimiento que se obtiene de las experiencias de los usuarios teniendo como base una evaluación del problema y de las experiencias pasadas para darle solución y se denominan razonamiento basado en casos (Lorenzi and Ricci, 2003). En el segundo enfoque se tiene el conocimiento mediante ontologías y es una forma de representar diferentes relaciones y propiedades de las entidades del dominio al que hace referencia, por medio del cual se puede obtener una métrica de similitud entre los ítem y los usuarios del sistema (Blanco-Fernandez et al., 2006) o, teniendo en cuenta la descripción de los ítems del sistema y los diferentes contextos que previamente han registrado los usuarios (Zhen et al., 2010). Como tercer enfoque se tiene la implementación de un algoritmo de búsqueda incremental que busca eliminar ítems teniendo en cuenta restricciones añadidas al algoritmo partiendo de las preferencias de los usuarios (Felfernig et al., 2006). Y el

cuarto enfoque se relaciona con las relaciones de preferencia del usuario teniendo en cuenta si un usuario prefiere una alternativa A en lugar de otra alternativa B y se pueden relacionar utilizando indiferencia, preferencia débil y preferencia estricta (Zanker et al., 2010).

### **Sistema de recomendación demográfico**

En el caso de los sistemas de recomendación demográficos la recomendación para el usuario se fundamenta en el perfil demográfico del usuario, es decir, el perfil con características específicas de edad, nivel educativo, ciudad de residencia, entre otras, y esta información de perfil se relaciona con usuarios con características de perfil similares (Burke, 2002).

En términos de algoritmos de los sistemas de recomendación, existe gran relación entre estos sistemas de recomendación y los sistemas basados en contenido, pero difieren en la forma de calcular las similitudes de los usuarios en sus perfiles, ya que los sistemas demográficos utilizan la información suministrada por el usuario en su perfil y no se requiere de ninguna valoración previa a los ítems que relaciona el usuario en su perfil, evidenciando de esta manera que la base de los sistemas de recomendación demográficos es la información que suministra el usuario como información de su perfil y sus características (Ahn, 2008).

Analizando la comparación de estos dos sistemas, también se tiene como ventaja de los sistemas demográficos que no tienen requisito para inicio información histórica que se conoce como arranque en frío y, en contraste, su desventaja radica en la necesidad de contar con gran cantidad de información de carácter personal del usuario, y teniendo en cuenta el manejo de información personal de los usuarios, se ha evidenciado resistencia de parte de estos para la entrega de esta información en la mayoría de los casos (Rich, 1979).

De esta forma se evidencia que los sistemas de recomendación demográficos tienen su inicio en la información que suministra el usuario y la clasificación de ellos se realiza teniendo en cuenta las características personales de cada usuario y las sugerencias o recomendaciones se generan a partir del grupo demográfico en el cual se encuentra relacionado el usuario en base a una ficha de clasificación que se encuentra en el algoritmo del sistema y en la mayoría de los casos se tienen en cuenta los estereotipos de los usuarios (Pazzani, 1999). También se puede implementar que el sistema organice los distintos estereotipos de una forma jerárquica, de tal manera que se tiene un conocimiento a mayor nivel de esta información, permitiendo realizar razonamientos de inferencia. Directamente en la recomendación, estos estereotipos se combinan con toda la información que relaciona el usuario con el fin de generar un resumen de su perfil, y de esta manera usar esta información simplificada para filtrar los ítems que se presentan al usuario, así como la información que se presenta sobre cada producto.

Aunque en los inicios de los sistemas de recomendación se utilizaban mucho los sistemas de recomendación demográficos, actualmente se aplican a dominios muy concretos, debido a la dificultad de disponer de los datos necesarios para su funcionamiento, teniendo en cuenta las políticas de seguridad de la información y la poca receptividad de parte de los usuarios para suministrar su información personal.

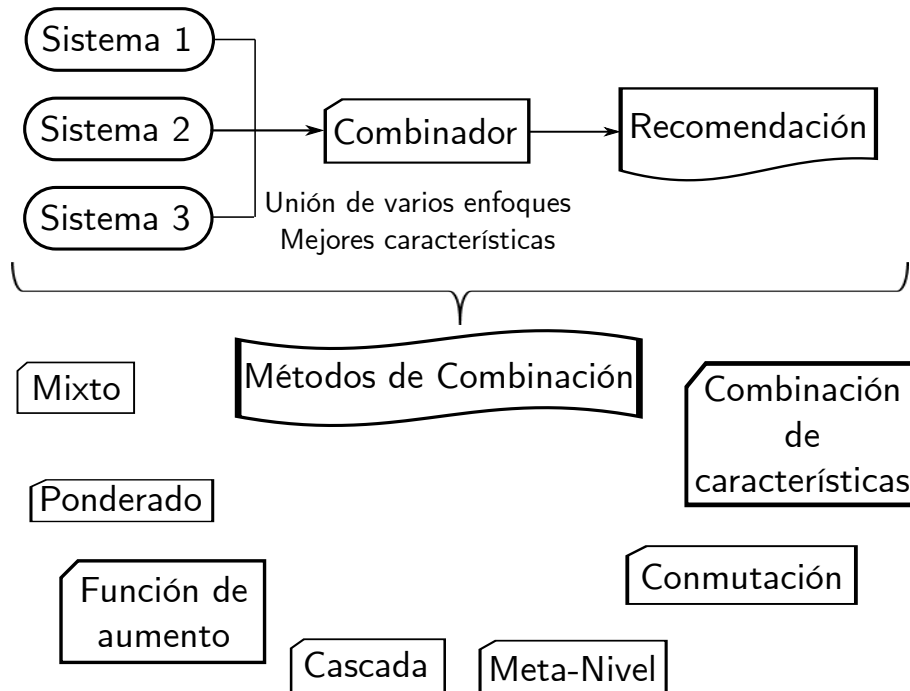
### **Sistema de recomendación híbridos**

En los sistemas de recomendación híbridos se evidencia la unión de dos o varias técnicas de recomendación en busca de complementarse entre sí utilizando las mejores características de cada enfoque y generar una mejor recomendación al usuario (Rodríguez, 2017).

Teniendo en cuenta la revisión del estado de arte de las diferentes fuentes bibliográficas relacionadas se concluye que los sistemas de recomendación híbridos más usa-



Figura 4

*Métodos de recomendación híbrida*

*Nota.* Se identifica en primera instancia la estructura base de un sistema de recomendación híbrida que representa la unión de varios enfoques resaltando las mejores características de cada uno y en la parte inferior se identifica la forma de combinación que se puede aplicar.

dos tienden a combinar técnicas de filtrado colaborativo con otras técnicas de recomendación con el fin de obtener un resultado más fiel (Muñoz, 2014), tomando de esta manera a la recomendación por filtrado colaborativo como el enfoque de mayor prioridad y eficiencia con respecto a los demás. Adicionalmente, se evidencia que, en la mayoría de los desarrollos implementados como recomendaciones híbridas, se ha combinado el filtrado colaborativo con sistemas basados en contenido, esto buscando la forma de reducir las desventajas de las dos técnicas, especialmente cuando se trata del arranque en frío por número de usuarios, recordando que la hibridación busca combinar las fortalezas de las técnicas integradas para generar recomendaciones válidas (Zhuhadar et al., 2009; Bertani et al., 2020).

También se concluye que en el momento de seleccionar las técnicas o enfoques

que se van a combinar en el sistema híbrido, es importante tener en cuenta que hay diferentes métodos con los cuales se puede combinar dichas técnicas (Jannach et al., 2010), entre los cuales sobresalen tres tipos de diseños en un sistema de recomendación híbrido (Rodríguez Marín et al., 2016):

- *Monolítico*: En donde se combinan las puntuaciones o votos dados por el usuario con el fin de producir una única recomendación.
- *Paralelizado*: Está compuesto de dos enfoques de recomendación que producen de forma independiente las recomendaciones y al final se combinan en un único conjunto.
- *Segmentado*: En el diseño segmentado se busca la incorporación de varias técnicas de recomendación de forma secuencial.

Cada sistema de recomendación tiene su propia metodología de hacer sugerencias. Cada una de ellas obtiene información importante de usuarios e ítems para realizar una recomendación desde un enfoque diferente. Sin embargo, aún dejan a un lado factores de gran importancia a la hora de hacer una predicción. La idea de los sistemas de recomendación híbridos se basa en la combinación de técnicas, que permitan que lo que se recomiende pueda estar más cercano a lo que un usuario realmente desea.

Para la implementación de los algoritmos de sistemas de recomendación se utiliza Inteligencia Artificial, que corresponde al campo científico de la informática que se centra en la creación de programas y mecanismos que pueden mostrar comportamientos considerados inteligentes. En otras palabras, la IA es el concepto según el cual “las máquinas piensan como seres humanos” (Muthukrishnan et al., 2020). Normalmente, un sistema de este tipo es capaz de analizar datos en grandes cantidades, identificar patrones y tendencias y, por lo tanto, formular predicciones de forma automática, con rapidez y precisión, lo importante es que permite que las experiencias cotidianas sean más inteligentes integrando análisis predictivos y otras técnicas de IA en aplicaciones que se utilizan de forma constante. Esto ha llevado a identificar técnicas que permiten el uso de la IA en diferentes procesos:

- *Aprendizaje automático*: Es la ciencia que se encarga de hacer que las computadoras realicen acciones sin necesidad de programación explícita. La idea principal aquí es que se les puede proporcionar datos a los algoritmos de Aprendizaje automático y luego usarlos para saber cómo hacer predicciones o guiar decisiones. Se centra en desarrollar sistemas que aprenden, o mejoran el rendimiento, en función de los datos que consumen. Algunos ejemplos de algoritmos de Aprendizaje automático incluyen los diagramas de decisiones, algoritmos de agrupamiento, algoritmos genéticos, redes Bayesianas, entre otros.
- *Aprendizaje profundo*: Técnica de Aprendizaje automático que utiliza redes neuronales (el concepto de que las neuronas se pueden simular mediante unidades computacionales) para realizar tareas de clasificación. Las redes de aprendizaje profundo aprenden mediante la detección de estructuras complejas en los datos que reciben (Perconti and Plebe, 2020). Cuando se crean modelos computacionales compuestos por varias capas de procesamiento, las redes pueden crear varios niveles de abstracción que representen los datos. Algunos ejemplos de aplicaciones prácticas del Aprendizaje profundo relacionan la identificación de vehículos, peatones y placas de matrícula de vehículos autónomos, reconocimiento de imagen, traducción y procesamiento de lenguaje natural, entre otros.
- *Descubrimiento de datos inteligentes*: Es el próximo paso en soluciones de Inteligencia Empresarial, la idea consiste en permitir la automatización total del ciclo de la IE: la incorporación y preparación de datos, el análisis predictivo y los patrones y la identificación de hipótesis. Este es un ejemplo interesante de la recuperación de datos inteligentes en acción. La información que ninguna herramienta de IE había descubierto.
- *Análisis predictivo*: El concepto principal de análisis predictivo (o modelado) significa que se puede utilizar un número de variables combinadas con resultados para generar un modelo que proporcione una puntuación que representa la probabilidad de un evento. El análisis predictivo emplea datos históricos para predecir eventos futuros. Los casos de uso en los negocios son amplios: modelos de crédito, modelos

de segmentación de clientes (agrupamiento), modelos de probabilidad de compra y modelos de migración de clientes, entre otros.

Para poder ofrecer a los clientes un servicio o producto, los sistemas de recomendación utilizan algoritmos con aprendizaje de máquina para predecir el proceso y encontrar ítems más adecuados y que tengan mayor relación con las características y perfiles de los usuarios. En ese orden, la IA por medio de machine learning y de la técnica deep learning permite a los desarrolladores crear algoritmos que replican la lógica que se sigue en la vida antes de tomar una decisión y que lleva a preguntar a un amigo o conocido, con el que se comparten aficiones, o a un experto de confianza o en el área. La gran diferencia es que en lugar de preguntar a dos o tres personas, la inteligencia artificial permite hacer una trazabilidad de consultas que equivalen a hablar con ciudades completas partiendo de la cantidad de información que se tenga almacenada en los registros.

El hecho de que hoy la IA sea más tangible que nunca se debe precisamente a que los usuarios generan una huella digital cada vez más completa. Cada interacción en las redes sociales, en la web o en los sistemas de uso común aportan datos que permiten conocer gustos, tendencias, preferencias, entre otras características, demostrando que estos sistemas y sus algoritmos dotan a las máquinas de la capacidad de aprender y realizar tareas complejas sin haber sido programadas para ello de forma explícita por medio de un diseño a base de motores que permiten tomar decisiones basadas en algún razonamiento concreto.

## Metodología

### Características generales

La metodología es la estrategia de investigación que se elige para responder a las preguntas de investigación y depende tanto de éstas como del marco teórico de la investigación (Hernández Sampieri and Fernández Collado, 2014). Se trata de optar por una estrategia de investigación general, ya sea de índole cuantitativa, cualitativa o mixta, así como el nivel de complejidad y detalle deseado.

En segunda instancia, se selecciona la forma como se va a reunir la información y las técnicas de análisis de la misma, siendo estas últimas, las herramientas más específicas de la investigación y, por ende, de mayor relevancia.

La metodología propuesta para el desarrollo de los objetivos está dividida en fases. Para las fases iniciales del proyecto en donde se realiza la búsqueda del estado del arte y el desarrollo del módulo de recomendación, de esta forma se utilizan dos marcos metodológicos, siendo el primero tomado desde la ingeniería de sistemas complejos (Faulconbridge and Ryan, 2005) que brinda herramientas para la administración de proyectos técnicos complejos en ingeniería; y adicionalmente se trabajan conceptos extraídos de ASUM, por sus siglas en inglés, Analytics Solutions Unified Method, el cual está orientado al desarrollo de soluciones con base en analítica de datos (Angée et al., 2018).

Estas fases están enfocadas en el logro de los objetivos por medio de:

a. *Identificación del sistema de recomendación*: planteamiento inicial de los re-

querimientos del sistema y las características a tener en cuenta para la recomendación según necesidad.

b. *Selección de la tecnología disruptiva:* Selección de tecnología acorde al sistema identificado que se desea utilizar de acuerdo con la disponibilidad de las herramientas necesarias para su desarrollo.

c. *Diseño básico y en detalle de la aplicación:* Diseño de la aplicación que soporta el sistema de recomendación diseñado.

d. *Validación del diseño:* Aplicación de procedimientos y métricas para la medición de hiper parámetros de los modelos propuestos.

Para el diseño del sistema de recomendación se tiene en cuenta en primera instancia la estructura de estos sistemas en relación con los elementos que lo comprenden. Esta estructura generalizada comprende los siguientes elementos:

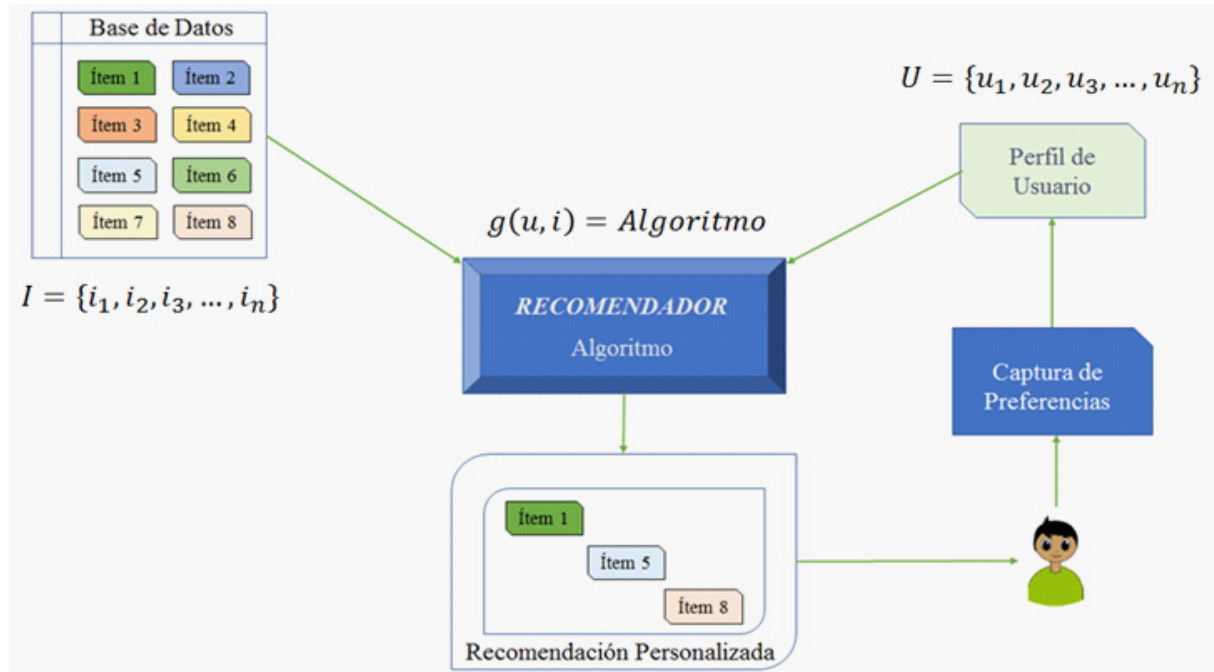
En la Figura 5 se relacionan los elementos del sistema de recomendación, evidenciando una Base de datos con listado de ítems, en algunos casos esos ítems pueden tener relación entre si por medio de características similares. Estos ítems son recomendados al usuario según sus preferencias de acuerdo con el perfil de usuario creado en base a sus preferencias.

## **Ítems – Base de datos**

Como se ha mencionado, las bases de datos forman parte del eje central de un sistema de recomendación, dado que son los elementos que contienen la información tanto de los usuarios como de los ítems que conforman el propósito del sistema de recomendación. En estos sistemas se usan bases de datos que se encuentran en archivos cuyo formato sea de fácil acceso y aplicabilidad, por lo general, el manejo de datos en este tipo de sistemas se realiza con archivos de excel.

Figura 5

*Elementos del sistema de recomendación.*



*Nota.* En términos generales, los sistemas de recomendación están conformados por bases de datos con la información del Usuario y el algoritmo que se encarga de filtrar la información y procesarla de tal manera que entregue al usuario, de acuerdo con su perfil, la recomendación esperada.

Es importante garantizar que la información registrada en las bases de datos que correspondan sea completa y permita la relación entre los diferentes ítems y los usuarios, de tal forma que se presente una congruencia entre los elementos que las comprenden.

## Recomendador

El sistema de recomendación propuesto se encuentra en lenguaje de programación Python que es un lenguaje de programación de código abierto, orientado a objetos, muy simple y fácil de entender, su sintaxis es sencilla que cuenta con una vasta

biblioteca de herramientas, que hacen de Python un lenguaje de programación único (Van Rossum and Drake, 2009). Este lenguaje de programación está adquiriendo muchísima popularidad en el mundo de Machine Learning. En parte, esto se debe a la disponibilidad de una gran cantidad de librerías para visualización, cálculo numérico, análisis de datos, aprendizaje automático y deep learning.

Las características del lenguaje de programación Python se resumen a continuación:

- *Interpretado*: Cuando se escribe un código en realidad lo que se hace es hablar un lenguaje más fácil de comprender para el programador y que luego será traducido a lenguaje de máquina que es lo que puede entender el procesador. Los lenguajes Interpretados son aquellos en los que el código del programador es traducido mediante un intérprete a medida que es necesario. Entre los más comunes se encuentra Python, Ruby, Javascript, etc., lo que lleva a invertir menos tiempo en desarrollo y prueba de una aplicación y el código fuente puede ser ejecutado en cualquier software que disponga de intérprete (Windows, Linux, Mac, Android, Web (Van Rossum and Drake, 2009).
- *Multiparadigma*: Python es un lenguaje que soporta más de un paradigma, suponiendo paradigma como modelo de desarrollo (y cada lenguaje tiene el suyo). Python admite como paradigmas el imperativo (se describe paso a paso un conjunto de instrucciones que deben ejecutarse para variar el estado del programa y solucionar el problema), funcional (permitir operar con datos de entrada y salida con la posibilidad que el usuario ingrese datos que serán procesados para generar otros datos de salida.) y orientado a objetos (Los objetos manipulan los objetos de entrada para la obtención de resultados de salida específicos donde cada objeto ofrece una función específica y permite la agrupación de bibliotecas o librerías) (Van Rossum and Drake, 2009).



## Librerías para el algoritmo

Para el funcionamiento de los sistemas de recomendación se requieren algoritmos que optimizan el análisis de los datos para la generación de las recomendaciones a los usuarios (De Nart and Tasso, 2014). Para ello es necesario trabajar con algunas librerías específicas que se requieren para que el código elaborado cumpla con las características deseadas, evidenciando en el estudio bibliográfico que actualmente se da mayor prelación a las librerías disponibles en código abierto (Khan and Al-Badi, 2020). Sin embargo, es importante resaltar que con los diferentes desarrollos y aplicaciones con enfoques específicos, es común encontrar modelos de software de sistemas de recomendación como servicios (SaaS, por sus siglas en inglés) (Loukis et al., 2019).

A continuación, se relacionan algunas de las librerías relacionadas con sistemas de recomendación y sus algoritmos:

- *Pandas*: Corresponde a un paquete de Python que facilita el uso de estructuras de datos similares a los dataframes que se usan en R (Reback et al., 2020). Se aclara que Pandas no es una librería directamente orientada al desarrollo de algoritmos de sistemas de recomendación, pero su uso se ha vuelto muy común para el preprocesamiento de datos, por lo cual hace que Pandas sea de gran utilidad en los sistemas de recomendación basados en Python. Como ventaja se tiene que permite la representación de los datos de forma tabular, facilitando así el desarrollo de acciones relacionadas con la selección y el filtrado de los datos, valores o etiquetas, para así lograr fusionarlos y unirlos, entre otras posibles funciones.
- *Scikit-learn*: Esta librería de Python tiene funciones específicas para aplicaciones de Machine Learning y Analítica de Datos y su funcionamiento se basa en las librerías NumPy, SciPy y Matplotlib. La principal ventaja es que Scikit-learn está relacionada con su facilidad de uso y la gran cantidad de técnicas de aprendizaje automático que implementa, lo cual lleva a implementar líneas de programación enfocadas a realizar aprendizaje de tipo supervisado y no supervisado (Pedrego-

sa et al., 2011). Cuenta con módulos de descomposición matricial y algoritmos de similitudes, que son funciones utilizadas en la construcción de los sistemas de recomendación.

- *Lenskit*: Esta es una librería de código abierto realizada por el grupo de investigación de GroupLens (Ekstrand et al., 2011). Esta librería está compuesta por un conjunto de herramientas específicas para usar en algoritmos de sistemas de recomendación con características específicas para implementaciones modulares de varios algoritmos, específicamente los algoritmos de sistemas de recomendación implementados con filtrado colaborativo. La primera versión desarrollada fue para implementar en java y posteriormente fue migrada para usos en Python (Ekstrand, 2018).
- *MyMediaLite*: Es una librería de código abierto para utilizar en lenguaje C# y se caracteriza por implementar varios algoritmos de filtrado colaborativo, como los relacionados con ítem y usuario (generador de kNN), algoritmos no personalizados (Random, Popularidad) o de factorización de matrices (WRMF y BPRMF), entre otros (Gantner et al., 2011).
- *Surprise*: Es un scikit de Python que facilita la creación y el análisis de datos para sistemas de recomendación que tratan con datos de calificación explícitos. Esta herramienta proporciona varios algoritmos de predicción que se encuentran listos para usar, entre los que se encuentran algoritmos de línea base, métodos de vecindad, basados en factorización matricial (SVD, PMF, SVD ++, NMF), entre otros, así como la integración de varias medidas de similitud, entre las que sobresalen el coseno de similitud, MSD y Pearson (Hug, 2020a).

## Validación del diseño

La analítica de datos para este diseño se centra en la recopilación y el análisis de los datos de la muestra con el fin conocer la trayectoria de aprendizaje de cada uno y

proceder, con el sistema de recomendación, a proporcionar orientación sobre los cursos a matricular por programa y optimizar el procedimiento de matrícula para el estudiante.

Para el diseño del algoritmo que representa este sistema de recomendación propuesto se tiene en cuenta el procedimiento de aplicación de la factorización matricial que corresponde a una clase de algoritmos que aplican la técnica de recomendación por filtrado colaborativo, por tanto, el desarrollo propuesto utiliza para la recomendación el enfoque de filtrado colaborativo para presentar las sugerencias y/o recomendaciones al usuario.

El funcionamiento de estos algoritmos está relacionado directamente con la acción de descomponer la matriz de interacción ítem-usuario en un producto de dos matrices rectangulares de menor dimensión (Lu et al., 2015). Este método caracteriza los datos relacionados en ítem-usuario por vectores de factores inferidos por medio de patrones definidos en la Matriz de identificación, de esta manera, cuando se presenta una correspondencia entre los diferentes factores de un ítem con los factores de un usuario se procede a generar la recomendación.

Los algoritmos que utilizan modelos obtenidos por medio de factorización matricial mapean de forma conjunta los ítems y los usuarios en un espacio conocido como factores latentes de una determinada dimensión. En estos casos, cada ítem  $i$  se asocia con un vector  $q_i \in \mathbb{R}^f$  y cada usuario  $u$  se asocia con un vector  $p_u \in \mathbb{R}^f$ , en donde  $f$  hace relación a la dimensión de factores latentes (Koren et al., 2009).

De acuerdo con esta información, cuando se tiene un ítem  $i$ , los elementos que conforman el vector  $q_i$  permite la medición del grado en el cual ese ítem se relaciona con esas características o factores determinados. Mientras que para cualquier usuario  $u$ , los elementos que componen el vector  $p_u$  relacionan la medición del grado con el cual el usuario se relaciona o presenta interés por los ítems que se encuentran en mayor grado de correspondencia con la matriz factorial (Koren et al., 2009). De esta manera, se obtiene un producto escalar de los dos vectores ( $q_i$  y  $p_u$ ) que genera el valor relacionado

con el interés que un usuario tendría respecto a las características de un ítem específico y se determina de acuerdo con la siguiente ecuación:

$$r'_{ui} = q_i^T \cdot p_u \quad (6)$$

En donde  $q_i$  y  $p_u \in \mathbb{R}^f$ , en donde se presenta el mayor desafío de estos algoritmos, ya que se relaciona con el mapeo que se debe realizar con cada ítem por cada usuario en vectores factoriales, generando una matriz compacta agregando mayor eficiencia en el sistema de recomendación, garantizando un número de atributos menor en comparación del número total de ítems posibles. Adicional a esto, otra ventaja que se encuentra con este proceso es que genera una red de vecinos más efectivo, ya que los usuarios se encuentran conectados entre si a través del mismo espacio de relación de características.

Como se evidencia, el desafío en los sistemas de recomendación se relaciona directamente con la forma en que se plantea el tratamiento de los datos obtenidos con el fin de generar recomendaciones precisas para los usuarios. La literatura ha concluido que se encuentran, entre otros, tres (3) grandes dificultades en el tratamiento de estos datos y se relacionan en primera instancia con la gran cantidad de datos que se pueden obtener conllevando a que el algoritmo diseñado presente interacciones en su programación que garanticen que el sistema genera una respuesta rápida de su ejecución; como segunda dificultad se encuentra la carencia de información completa haciendo énfasis en las ponderaciones asignadas por los usuarios a los ítems relacionados, teniendo en cuenta que en muchas ocasiones los sistemas de recomendación dependen de este tipo de información emitida por el usuario; y como tercera dificultad y no menos importante, se tiene la naturaleza dinámica de los datos, es decir, la variación constante de la información a medida que se registran nuevos usuarios y estos a la vez van registrando nuevos ítems, evidenciando de esta manera que es importante garantizar que el algoritmo se actualice con rapidez y precisión (Zhou et al., 2015).

Como respuesta de solución a estas dificultades, se han utilizado procedimientos matemáticos aplicados como métodos de reducción enfocados específicamente en la de

dimensión de las bases de datos utilizadas o en la forma de generar un agrupamiento de estos datos, utilizando lo que se conoce como el modelo basado en agrupamiento, para lo cual se plantea el uso de diferentes técnicas de agrupamiento entre las que se encuentran:

- Agrupamiento de K-medias: Agrupamiento que se enfoca en generar una partición de un conjunto de  $n$  observaciones en  $k$  grupos en donde se garantice que cada observación  $n$  pertenece al grupo  $k$  en el cual el valor medio es más cercano, razón por la cual es de gran uso en procesos de minería de datos (Xue et al., 2005).
- Árbol de expansión mínimo: Corresponde a un procedimiento cuyo inicio se da desde un vértice y encuentra todos los nodos accesibles y las relaciones en conjunto que permiten la conexión de esos nodos con el menor peso o tamaño posible (Demir et al., 2007).
- Partición alrededor de medoides (PAM): Corresponde a un algoritmo de agrupamiento directamente relacionado con los algoritmos k-means y medoidshift y son algoritmos que dividen el conjuntos de datos por medio de particiones y de esta forma minimizar la distancia entre puntos que se adicionan a un grupo y otro punto asignado como el centro de ese grupo (Chakraborty, 2009).
- Modelo basado en factorización matricial como descomposición en valor singular (SVD): El objetivo de esta técnica de factorización de matrices, es dividir la matriz de usuarios/ítems en dos matrices (Funk, 2006).
- Factorización matricial no negativa (NMF): Corresponde a la aproximación de una matriz cuyos valores son positivos o nulos aplicando la multiplicación de dos submatrices de elementos también no-negativos de dimensión previamente determinada (Zhang et al., 2019).

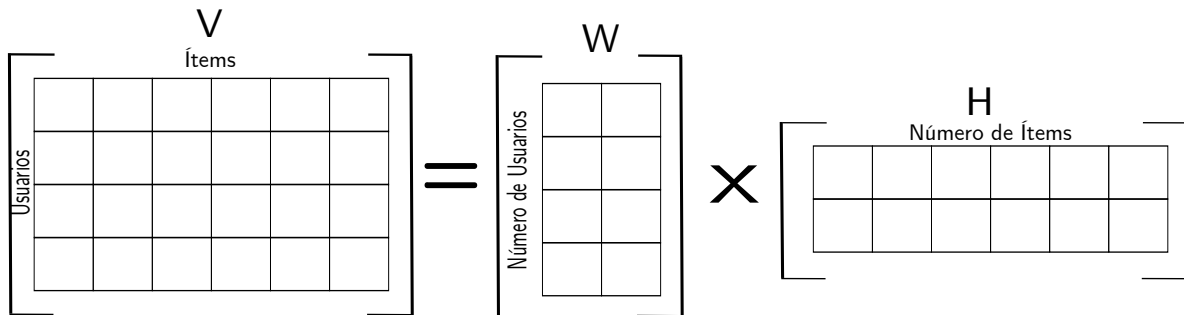
#### *Aplicación factorización no negativa de matrices*

Algunos de los desarrollos implementados tienen como eje central la aplicación de la factorización matricial no negativa (NMF o NNMF) o conocida también como

técnica de aproximación matricial no negativa, en donde teniendo una matriz  $V$  se procede a factorizarla en dos matrices por lo general y se conocen en la teoría como matrices  $W$  y  $H$  con la particularidad que ninguna de las tres matrices comprenden elementos negativos, como se evidencia en la Figura 6.

Figura 6

*Factorización no negativa de matrices NMF.*



*Nota.* Se relaciona el comportamiento gráfico de la aproximación matricial no negativa evidenciando que la matriz  $V$  se factoriza en dos matrices denominadas  $W$  y  $H$ , resaltando que una de sus propiedades indica que las tres matrices ( $V$ ,  $W$  y  $H$ ) no tienen valores negativos.

El objetivo de garantizar la no negatividad de estas matrices consiste en que las matrices resultantes de este desarrollo van a ser más fáciles de inspeccionar generando aproximaciones numéricas.

Formalmente el problema de factorización se define como: sea  $V \in \mathbb{R}^{m \times n}$  donde  $m$  corresponde a la cantidad de usuarios y  $n$  es la cantidad de ítems a evaluar.

Para esta multiplicación de matrices se computan los vectores columnas de  $V$  como combinaciones lineales de los vectores columnas en  $W$  usando los coeficientes proporcionados por las columnas que comprenden  $H$ .

En estos casos, cada una de las características que conforman el vector columna en la matriz de características  $W$  se comporta como un arquetipo que comprende un conjunto de palabras en donde cada palabra determina el rango de la característica principal y se tiene que cuanto mayor sea el valor de la celda de una palabra enton-

ces mayor será el rango de la palabra en esa misma característica (Lopez-Nores et al., 2012). De igual forma, cuando una columna en la matriz de coeficientes  $H$  está representando un documento original con un valor que representa el rango del documento para una característica determinada. De esta forma se puede reconstruir un documento o vector columna de la matriz de entrada del sistema de recomendación utilizando así una combinación lineal de todas las características relacionadas como vectores columnas en  $W$  en donde cada una de las características allí relacionadas se pondera teniendo en cuenta el valor de la celda de característica de la columna del documento en  $H$  según corresponda.

Cuando se implementa una NMF se tiene como complemento la propiedad inherente de agrupamiento (Taslaman and Nilsson, 2012) cuya función es agrupar de forma automática las columnas de datos de entrada conocida como

$$\mathbf{V} = (v_1, \dots, v_n) \quad (7)$$

Teniendo en cuenta la ecuación definida para la factorización de las variables del sistema, se busca la forma de descomponer dicha matriz en la multiplicación de dos matrices de menor dimensión, específicamente aplicando el procedimiento que se relaciona a continuación:

$$V \cong WH^T \quad (8)$$

En donde se tiene que  $W \in \mathbb{R}^{m \times p}$  y  $H \in \mathbb{R}^{n \times p}$

En estos casos, para encontrar las matrices del sistema se recurre a un problema de optimización como se evidencia en la ecuación 2.4:

$$\min_{W, H} \|V - WH\|_F \quad (9)$$

que está sujeto a las condiciones  $W \geq 0$  y  $H \geq 0$

### *Descomposición de valores singulares (SVD) para factorización matricial*

Técnicamente hablando, la descomposición en valores singulares, SVD por sus siglas en inglés, hace referencia a una factorización que se realiza a una matriz de tipo real o compleja con muchas aplicaciones en estadística y otras disciplinas.

El objetivo de las técnicas de factorización de matrices, consiste en dividir la matriz de usuarios/ítems en dos matrices, que corresponden a una matriz para los usuarios y las características latentes  $f$  del problema en operación y una segunda matriz que relaciona los ítems del problema con las características latentes. Es decir, se tiene una matriz  $V \in \mathbb{R}^{m \times n}$  que corresponde a la matriz de ponderaciones de un problema determinado en donde  $m$  representa la cantidad total de usuarios y la variable  $n$  corresponde a la cantidad total de ítems en determinado sistema de recomendación.

De esta forma, se garantiza que cada ítem  $i$  tiene asociado un vector que cumple la condición  $q_i \in \mathbb{R}^f$  y a la vez garantiza que cada usuario  $u$  se relaciona con un vector identificado con la condición  $p_u \in \mathbb{R}^f$ . Cumpliendo de esta manera que el producto punto resultante de la operación  $q_i^T p_u$  representa de forma aproximada la interacción entre el usuario  $u$  y el ítem  $i$ . En estos casos, los usuarios e ítems del sistema pertenecen a un espacio con características de dimensiones representados por  $k$ , por tanto, el vector  $p_u$  representa el nivel de relevancia de una característica para un usuario mientras que el vector  $q_i$  representa el nivel en que dicha característica se relaciona con un ítem del sistema.

De esta manera se tiene que la predicción de la ponderación que el usuario  $u$  le da al ítem  $i$  se conoce mediante la representación dada por  $\hat{r}_{ui}$  y se representa matemáticamente de la siguiente forma:

$$\hat{r}_{ui} = \mu + b_i + b_u + q_i^T p_u \quad (10)$$

En donde se tiene que la variable  $\mu$  representa el valor numérico del promedio global de todas las ponderaciones de la matriz de puntuaciones, las desviaciones observada o bias del ítem y usuario se representan por medio de las variables  $b_i$  y  $b_u$  respec-



tivamente. Dando cumplimiento a estas indicaciones, la forma de determinar los parámetros mencionados se realiza mediante un problema de optimización:

$$\min_{b_*, q_*, p_*} \sum_{(u,i) \in X} (r_{ui} - \hat{r}_{ui})^2 + \lambda (b_i^2 + b_u^2 + \|q_i\|^2 + \|p_u\|^2) \quad (11)$$

En donde  $\lambda$  representa lo que se conoce como el factor de regularización y se puede calcular usando validación cruzada lo cual sirve para prevenir el efecto de sobreajuste en el momento de la regularización de los parámetros aprendidos.

En términos generales, lo que busca con el SVD en los algoritmos de sistemas de recomendación es tomar una matriz de ponderaciones o calificaciones y reducir su dimensión para todos los usuarios e ítems en tamaños o dimensiones más pequeños.

## Desarrollo

De acuerdo con las técnicas de recomendación señaladas, sus características, los requerimientos del sistema de recomendación a diseñar y la información que se tiene para el diseño de los mismos, se identifica que el sistema de recomendación a implementar, con la información que se tiene actualmente, corresponde a un sistema de recomendación por filtrado colaborativo, teniendo en cuenta las fortalezas de esta técnica de recomendación para generar recomendaciones válidas y el almacenamiento constante de la información a medida que más usuarios acceden al uso de este recurso.

De igual forma, la compatibilidad de esta técnica de recomendación con la Inteligencia Artificial por medio de machine learning, permitiendo a sus desarrolladores la creación de algoritmos que replican la lógica que se sigue en la vida antes de tomar una decisión y que lleva a preguntar a un amigo o un conocido, con el que se comparten aficiones, o a un experto de confianza o en el área.

La gran diferencia es que en lugar de preguntar a dos o tres personas, la inteligencia artificial aplicada en algoritmos específicos de sistemas de recomendación permite realizar una trazabilidad en las consultas que equivalen a hablar con ciudades completas partiendo de la cantidad de información que se tenga almacenada en los registros.

A continuación, se presenta en detalle el desarrollo metodológico que incluye la descripción de los datos utilizados para la creación del modelo, su estructura y las características que los representan. Así como el procesamiento que se realizó sobre estos para extraer contenido para el sistema de recomendación, teniendo en cuenta las diferentes técnicas y métodos utilizados para la optimización del sistema propuesto.

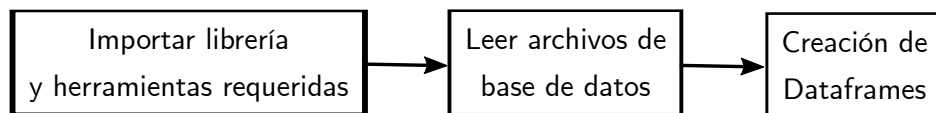
## Datos del sistema de recomendación

La librería Pandas permite utilizar Dataframes o DFs, que corresponden a estructuras de tipo tabular de dos dimensiones cuyo tamaño es modificable y por lo general son estructuras potencialmente heterogéneas, el objetivo de convertir los datos en este tipo de estructuras es que facilita la aplicación de operaciones de tipo aritmético tanto en las filas como en las columnas, lo cual ha evidenciado un punto importante de esta herramienta para el manejo adecuado de los datos que se tienen en los sistemas de recomendación (Reback et al., 2020).

Por esta razón, el manejo de las bases de datos que se tienen para el algoritmo se someten a un proceso interno de creación del Dataframe de cada archivo que contiene información base del sistema y de esta manera, el algoritmo en su estructura inicial representa en código lo que indica la Figura 7

Figura 7

*Proceso de generación de Dataframe del SR.*



*Nota.* Para la generación del dataframe del sistema de recomendación se deben importar las librerías y herramientas necesarias para el correcto funcionamiento del algoritmo, la lectura de los diferentes archivos que comprenden la base de datos y finalmente se crea el dataframe, de acuerdo con la lógica de programación estipulada en el algoritmo.

La base de datos para este sistema de recomendación está conformada por tres (3) archivos o documentos de hoja de cálculo de Microsoft Excel, con los cuales se van a crear los respectivos Dataframe para el sistema de recomendación y cada uno corresponde a un tipo de información diferente y que alimenta las características del sistema, estos archivos corresponden a:

1. *Usuarios*: Esta categoría integra los datos de estudiantes de Ingeniería Electrónica e Ingeniería de Telecomunicaciones de algunas sedes de la universidad, relacionando la mayoría de datos de estudiantes del CEAD Yopal y su historial académico en el desarrollo de su proceso formativo. En la Tabla 4 se presenta el tipo de información que contiene esta base de datos y la distribución que contiene este archivo que se encuentra en hoja de cálculo de Microsoft Excel.

Tabla 4

*Características base de datos Usuarios*

<b>Identificador</b>	<b>ProgramaActual</b>	<b>Acuerdo</b>	<b>Centro</b>
Número de C.C.	INGENIERÍA ELECTRÓNICA (Resolución 13155)	Código de acuerdo SENA si aplica	Centro al cual está matriculado
Número de C.C.	INGENIERÍA DE TELECOMUNICACIONES (Resolución 14518)	Código de acuerdo SENA si aplica	Centro al cual está matriculado

*Nota.* Se relacionan las características que se presentan en la base de datos de Usuarios en sus cuatro columnas.

De esta base de datos se resalta que se tiene como variable Identificador el documento de identidad del estudiante, ya que para la UNAD el código del estudiante es su mismo número de documento de identificación.

De igual manera se aclara que no todos los estudiantes que conforman la base de datos tienen acuerdo de homologación por convenio SENA.

La relación del centro del estudiante en la base de datos se tiene en cuenta para efectos de análisis demográficos para futuras mejoras el sistema de recomendación.

En la Tabla 5 se relacionan las variables que se utilizan en el Dataframe correspondiente para Usuarios y la descripción de cada una.

2. *Mallas curriculares*: Esta categoría integra las mallas curriculares de los dos programas mencionados, Ingeniería Electrónica e Ingeniería de Telecomunicaciones, in-

Tabla 5

*Variables de la base de datos Usuarios*

<b>Variable</b>	<b>Tipo de Dato</b>	<b>Dataframe</b>	<b>Descripción</b>
Identificador	Int	df_Usuarios	Número de documento de identidad del estudiante
ProgramaActualString		df_Usuarios	Programa académico matriculado por el estudiante: INGENIERÍA ELECTRÓNICA (Resolución 13155) o INGENIERÍA DE TELECOMUNICACIONES (Resolución 14518)
Acuerdo	String	df_Usuarios	Código de acuerdo de homologación por convenio con el SENA, si aplica para el estudiante
Centro	String	df_Usuarios	Nombre del centro de la UNAD en el cual se encuentra matriculado el estudiante

*Nota.* Se relacionan las características de las variables que comprenden el Dataframe de Usuarios.

cluyendo las opciones de cursos Electivos, Complementarios y Obligatorios. En la Tabla 6 se presenta el tipo de información que contiene esta base de datos y la distribución que contiene este archivo que se encuentra en hoja de cálculo de Microsoft Excel.

Tabla 6

*Características base de datos Mallas Curriculares*

<b>Nombre Curso</b>	<b>Cód Cur- so</b>	<b>Metodología</b>	<b>Créditos</b>	<b>Programa</b>	<b>Categoría</b>	<b>Sub cate- goría</b>
Sistemas Dinámi- cos	243005	Metodológico	3	IE, IT	Obligatorio	
Telecontrol	203052	Metodológico	3	IE, IT	Electivo	Disciplinar
Metrología	2030549	Metodológico	3	IE, IT	Electivo	Profundización
Protocolo	80007	Teórico	1	IE, IT	Electivo	Complementario

*Nota.* Se relacionan las características de las variables que comprenden el base de datos de los cursos que comprenden las mallas curriculares de los programas seleccionados.

De acuerdo con la información relacionada en este Dataframe, se aclara que se tienen en cuenta los cursos académicos ofertados para los planes nuevos o registro calificado actual de los programas de Ingeniería de Telecomunicaciones e Ingeniería Electrónica, así como los relacionados en los acuerdos internos No. 795 de julio 30 de 2020 y 972 de julio 30 de 2020 que establece las equivalencias de algunos cursos obligatorios y electivos entre planes antiguos y planes nuevos de los dos programas de formación en mención. En la Tabla 7 se relacionan las variables que se utilizan en el Dataframe correspondiente para Mallas Curriculares y la descripción de cada una.

3. *Historia Académica:* Esta categoría integra toda la historia académica de los estudiantes relacionados en la base de datos de Usuarios y que están relacionados con los dos programas Ingeniería Electrónica e Ingeniería de Telecomunicaciones. En la Tabla 8 se presenta el tipo de información que contiene esta base de datos y la distribu-

Tabla 7

*Variables de la base de datos Cursos*

<b>Variable</b>	<b>Tipo de Dato</b>	<b>Dataframe</b>	<b>Descripción</b>
NombreCurso	String	df_Cursos	Nombre del curso relacionado en la malla curricular
CódCurso	Int	df_Cursos y df_HA	Código de identificación del curso según malla curricular
Metodología	String	df_Cursos	Tipo de curso según malla curricular, puede ser curso Teórico o Metodológico
Créditos	Int	df_Cursos	Cantidad de créditos académicos del curso
Programa	String	df_Cursos	Programa académico al que hace referencia el curso: IE o IT
Categoría	String	df_Cursos	Clasificación del curso según malla curricular: Obligatorio o Electivo
Subcategoría	String	df_Cursos	Los cursos electivos se dividen en tres (3) subcategorías: Profundización, Disciplinarios y Complementarios

*Nota.* Se relacionan las características de las variables que comprenden el Dataframe de Cursos y la información que se encuentra en este archivo.

ción que contiene este archivo que se encuentra en hoja de cálculo de Microsoft Excel.

Tabla 8

*Características base de datos Historia Académica*

<b>Identificador</b>	<b>Cód</b>	<b>Nota75</b>	<b>Nota25</b>	<b>Nota</b>	<b>Estado</b>	<b>Periodo</b>
	<b>Curso</b>			<b>Final</b>		
CC	243005	4	3.9	4	Aprobado	2018 II PE- RIODO 16-04
CC	203052	1.7	0	1.3	Reprobado	2019 II PE- RIODO 16-04

*Nota.* Se relacionan las características de las variables que se encuentran en la base de datos de Historia Académica.

Respecto a esta base de datos de Historia Académica se aclara que es obtenida por medio de un proceso de Scrapping con los datos de los estudiantes que se usaron para este sistema y su información individual de avance en los procesos de formación registrada en el aplicativo de Registro y Control de la universidad con fecha de marzo de 2020.

En la Tabla 9 se relacionan las variables que se utilizan en el Dataframe correspondiente para Mallas Curriculares y la descripción de cada una.

Estos tres (3) archivos que conforman la base de datos del SR propuesto tienen variables y características particulares que se encuentran registradas en la Tabla 5 relacionando así las variables de cada archivo y el tipo de dato recomendado para su asignación, con su respectiva descripción que es de gran utilidad para la selección de características en la etapa de procesamiento de datos.

Teniendo en cuenta que el sistema de recomendación diseñado aplica para los estudiantes de Ingeniería Electrónica e Ingeniería de Telecomunicaciones matriculados en la UNAD, se crea la base de datos a partir del registro académico de 253 estudiantes distribuidos entre los diferentes periodos académicos de cada programa según su malla curricular, homologaciones por convenio con el SENA y con características diferentes



Tabla 9

*Variables de la base de datos HA*

<b>Variable</b>	<b>Tipo de Dato</b>	<b>Dataframe</b>	<b>Descripción</b>
Identificador	Int	df_HA	Número de documento de identidad del estudiante
CódCurso	Int	df_HA	Código de identificación del curso según malla curricular
Nota75	Float	df_HA	Valor numérico obtenido en los momentos de evaluación Inicial e Intermedio, valor entre 0.0 y 5.0
Nota25	Float	df_HA	Valor numérico obtenido en el momento de evaluación Final, valor entre 0.0 y 5.0
NotaFinal	Float	df_HA	Valor numérico definitivo obtenido en cada curso, valor entre 0.0 y 5.0
Estado	String	df_Cursos	Valoración cualitativa del curso: Aprobado cuando el valor numérico en NotaFinal se encuentra entre 3.0 y 5.0 y Reprobado cuando el valor numérico en NotaFinal se encuentra entre 0.0 y 2.9
Periodo	String	df_Cursos	Periodo académico y año en el cual fue matriculado el curso

*Nota.* Se relaciona la descripción de las variables que se encuentran en el Historial Académico, señalando a su vez el DF en donde se utiliza cada elemento.

de avance en la malla curricular. Teniendo en cuenta que hay cursos que no se comparten en los dos programas seleccionados, es necesario separar las bases de datos de los dos programas, con el fin de evitar mayor cantidad de cursos con ponderación de vacíos en la matriz de Usuarios/Ítems, por lo tanto, ahora se tienen dos dataframes, uno para Electrónica y otro para Telecomunicaciones. De ahora en adelante el desarrollo se centra en el DF de Electrónica que contiene la mayor cantidad de datos que corresponde a 238 Usuarios.

En el algoritmo diseñado para el sistema de recomendación se importan estas tres (3) bases de datos con el fin de procesar los dataframes correspondientes y se denominan de la siguiente manera:

- `df_HA` = Dataframe de Historia Académica de varios estudiantes.
- `df_Cursos` = Dataframe de Listado de los cursos correspondientes a cada programa.
- `df_Electronica` = Dataframe de Listado de estudiantes pertenecientes al programa de Ingeniería Electrónica.

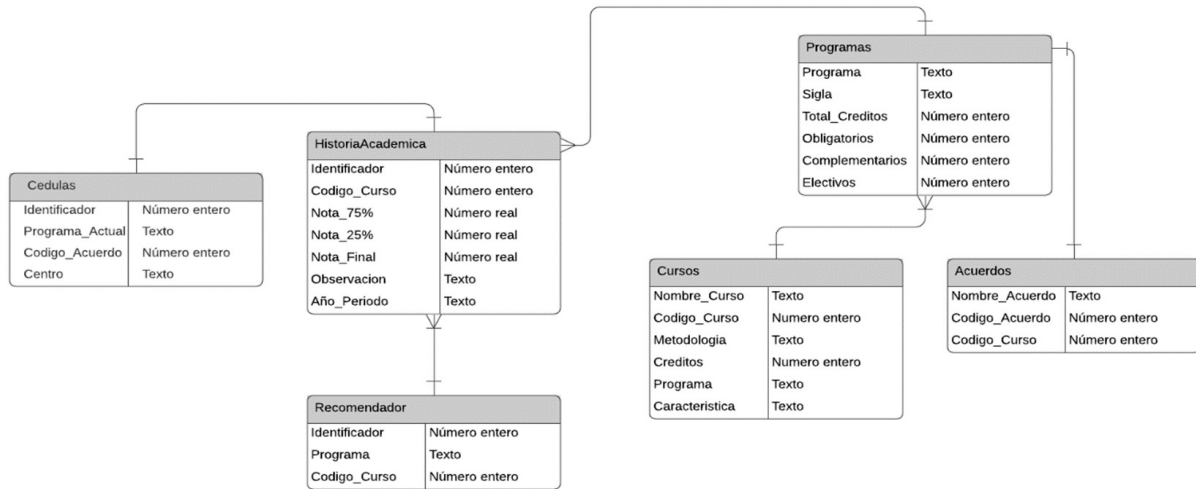
Estas bases de datos están relacionadas entre sí para la generación de recomendación de cursos al usuario, esta relación se representa mediante un modelo de entidad relación representado en la Figura 8.

En la Tabla 10 se relacionan las características de cantidad de datos y tamaño de los archivos de los dataframes usados para el algoritmo del sistema de recomendación propuesto.

### **Algoritmo del sistema de recomendación**

El sistema de recomendación propuesto se encuentra en lenguaje de programación Python que es un lenguaje de programación de código abierto, orientado a objetos, muy simple y de fácil entendimiento, su sintaxis es sencilla y cuenta con una vasta bi-

Figura 8

*Modelo Entidad - Relación del SR*

*Nota.* Se relaciona el modelo de entidad relación del sistema propuesto teniendo en cuenta las bases de datos que conforman el sistema y la información relacionada en cada una de ellas.

biblioteca de librerías y herramientas, que hacen de Python un lenguaje de programación muy utilizado para la obtención de este tipo de algoritmos (Van Rossum and Drake, 2009).

Este algoritmo se desarrolla en Jupyter Notebook que es una aplicación cliente-servidor lanzada en 2015 por la organización sin ánimo de lucro Proyecto Jupyter y una de sus grandes ventajas es que el programa se ejecuta desde la aplicación web como cliente y que funciona en cualquier navegador estándar. Para poder utilizar esta herramienta, se tiene como requisito previo la instalación y ejecución en el sistema el servidor Jupyter Notebook. Otra ventaja que presenta esta herramienta es que los documentos creados en ella se pueden exportar, entre otros formatos, a HTML, PDF, Markdown o Python y también pueden ser compartidos con otros usuarios por correo electrónico, utilizando Dropbox o GitHub o mediante el visor integrado de Jupyter Notebook (Kluyver et al., 2016).

En primera instancia, como en todo algoritmo, el primer objetivo es cargar las librerías necesarias para trabajar de acuerdo con los requerimientos de funcionalidad

Tabla 10

*Características de Dataframes usados*

<b>Dataframe</b>	<b>Número de datos DF</b>	<b>Columnas del DF</b>	<b>Tamaño del archivo</b>
df_HA	11610	7	417 KB
df_Cursos	238	4	28 KB
df_Electronica	234	7	26 KB

*Nota.* Se relaciona el número de datos que conforman cada Dataframe y el tamaño del archivo que representa cada uno.

y operación del sistema de recomendación y teniendo en cuenta las librerías relacionadas en el capítulo 2, se han utilizado para este algoritmo propuesto algunas librerías desarrolladas para uso en Python como pandas, numpy, sklearn.decomposition, sklearn.metrics.pairwise, scipy.sparse, lenskit, lenskit.algorithms, lenskit.batch, lenskit.metrics.predict, surprise y surprise.model-selection, ejecutando como primera acción del código del algoritmo la importación de las herramientas específicas requeridas de cada una de estas librerías señaladas anteriormente, tal como se relacionan en la Tabla 11

Estas librerías, junto con las herramientas señaladas, fueron usadas para la obtención de dos (2) algoritmos para el sistema de recomendación. Estos algoritmos implementan métodos diferentes para el agrupamiento de los datos, obtención de las matrices de operación y muestreo del sistema y la verificación del funcionamiento del sistema de recomendación propuesto por medio de las características mencionadas en el estado del arte para el filtrado colaborativo. La diferencia de los algoritmos propuestos radica en el método de agrupamiento utilizado y los modelos matemáticos aplicados para dicho proceso, obteniendo finalmente resultados similares para la recomendación al estudiante pero con tiempos de operación diferentes en la ejecución de cada código.

Los algoritmos propuestos corresponden a filtrado colaborativo con factorización no negativa de matrices y factorización con SVD, cuya finalidad radica en la veri-

Tabla 11

*Relación de librerías y herramientas importadas en el algoritmo*

<b>Librería</b>	<b>Herramienta</b>
pandas	
numpy	
sys	
sklearn.decomposition	NMF
sklearn.metrics.pairwise	cosine_similarity
scipy.sparse	csr_matrix
lenskit	crossfold
lenskit	util
lenskit	batch
lenskit.algorithms	mf-common,Recommender
lenskit.algorithms	item_knn
lenskit.batch	predict
lenskit.metrics.predict	rmse
surprise	Dataset
surprise	SVD
surprise	SVDpp
surprise	BaselineOnly
surprise	accuracy
surprise.model-selection	GridSearchCV
surprise.model-selection	RandomizedSearchCV
surprise.model-selection	train-test-split

*Nota.* Se relacionan las librerías y herramientas que se requieren para el correcto funcionamiento del algoritmo.

ficación de cual de los dos métodos de entrenamiento y aplicación de la metodología de filtrado genera mejores resultados en términos de precisión en la predicción y optimización de tiempos y recursos de máquina.

### **Pre-Procesamiento de datos**

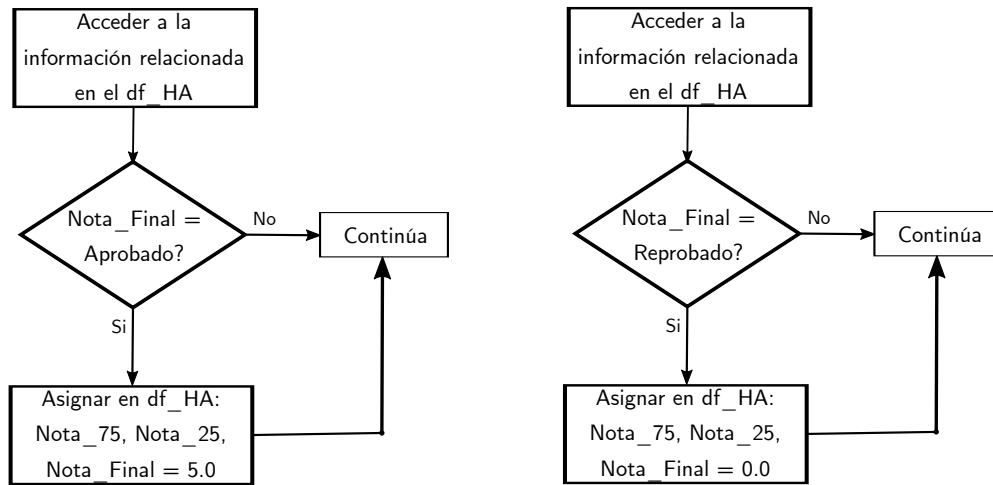
La base de datos de Historia Académica es sometida a un preprocesamiento en el mismo algoritmo con el fin de encontrar en el dataframe de la historia académica, aquellos cursos que no tienen calificación con valor numérico sino que relacionan una calificación de tipo cualitativo, como sucede con el caso particular del curso Electivo SISSU, que según el reporte de Registro y Control se conoce como Prestación del Servicio Social Unadista. Dicha calificación definitiva es procesada directamente por el algoritmo y realiza el cambio por su respectivo valor numérico asignado, obteniendo un valor numérico equivalente a 5 cuando la ponderación cualitativa es Aprobado y, en caso contrario, asigna una valoración numérica equivalente a 0 cuando la ponderación cualitativa es Reprobado.

Este proceso se realiza con condicional en la programación y la secuencia que desarrolla se representa en la Figura 9.

En este caso, los datos numéricos obtenidos por los estudiantes como calificación de los momentos de evaluación correspondientes a Nota-75, Nota-25 y Nota-Final se convierten a datos de tipo flotante para poder ser tratados más adelante.

La siguiente acción a tener presente en el tratamiento de la base de datos corresponde a la selección del programa académico del estudiante, teniendo en cuenta que el sistema de recomendación aplica para dos programas académicos con características específicas en cuanto a los cursos que incluye en su respectiva malla curricular. Por esta razón es necesario filtrar la base de datos de Usuarios especificando el programa académico matriculado por el estudiante en uso del sistema de recomendación y se crea un

Figura 9

*Procesamiento dataframe Historia Académica curso Electivo SISSU*

*Nota.* Procesamiento dataframe Historia Académica curso Electivo SISSU. El curso Electivo SISSU tiene una característica particular que su calificación no es cuantitativa sino cualitativa, por tanto, es necesario hacer una preprocesamiento desde el algoritmo en el dataframe de Historia Académica convirtiendo el valor cualitativo en su forma cuantitativa.

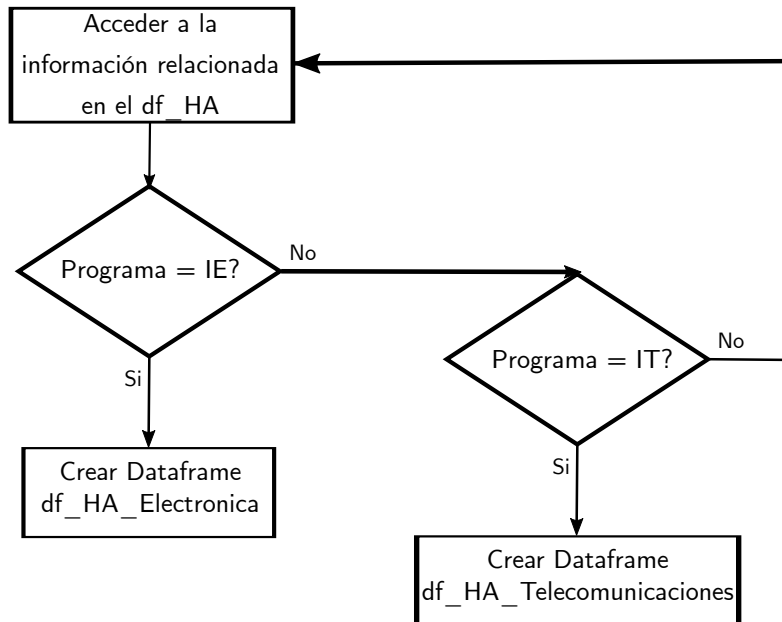
dataframe en el mismo algoritmo discretizando esta información.

Este proceso se realiza comparando los datos de usuarios que relacionan en la casilla Programa de la Tabla 7, identificando a los estudiantes de Ingeniería Electrónica con las siglas IE y los estudiantes de Ingeniería de Telecomunicaciones con las siglas IT. Con esta clasificación se crea el dataframe de historia académica exclusivo de cada programa, teniendo ahora `df_HA_Electronica` para los estudiantes de Ingeniería Electrónica y de igual manera se tiene `df_HA_Telecomunicaciones` para los estudiantes de Ingeniería de Telecomunicaciones, tal como se evidencia en la Figura 10.

En esta etapa del algoritmo primero se filtran los cursos que corresponden a cada programa académico, es decir, para el dataframe de Electrónica se filtran únicamente cursos que estén en la malla curricular de dicho programa o que se encuentren en acuerdos de homologación y/o acuerdos de equivalencia, lo mismo sucede con el dataframe de Telecomunicaciones, la Figura 11 nos señala este procedimiento presentando

Figura 10

*Creación de dataframe Historia Académica por programa académico*



*Nota.* Creación de dataframe Historia Académica por programa académico. El dataframe de Historia Académica relaciona información de los dos programas que forman parte del SR propuesto y para garantizar una mejor precisión de la recomendación a dar al usuario es necesario separar el dataframe en uno particular para cada programa.

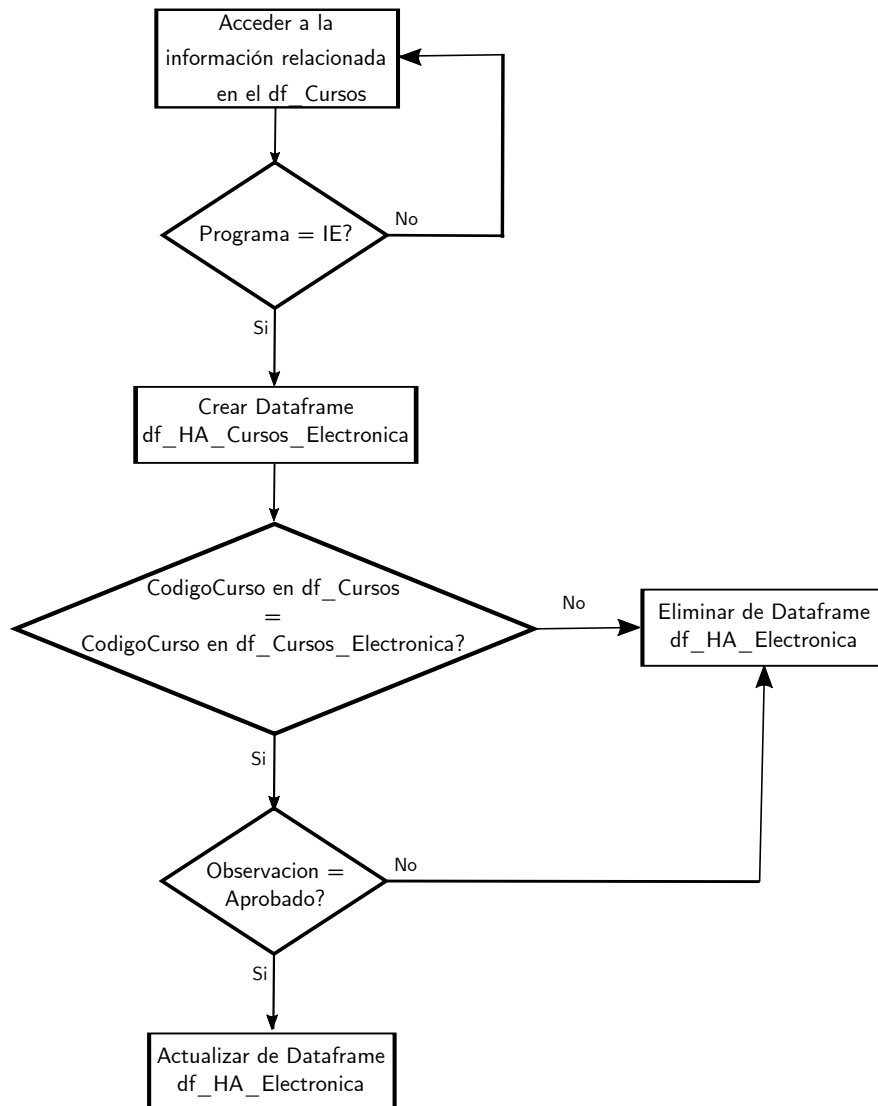
el caso del dataframe de Electrónica, aclarando que el paso a paso es el mismo para el dataframe de Telecomunicaciones.

En esta misma etapa también se realiza el filtrado de los cursos que registran APROBADO en la casilla de Observación con el fin de trabajar únicamente con los cursos que representan valor numérico superior a 3.0 y que en el momento de crear las matrices de entrenamiento y de prueba del sistema de recomendación no vayan a generar puntos críticos al momento de recomendar el curso al estudiante usuario del sistema.



Figura 11

*Filtrado de datos de dataframe Historia Académica por programa académico*



*Nota.* Filtrado de datos de dataframe Historia Académica por programa académico.

Teniendo en cuenta que se separa el dataframe para los usuarios de cada programa, es necesario realizar el proceso de filtrado de los cursos que corresponden a cada programa, tomando en cuenta la información registrada en el dataframe de Cursos.

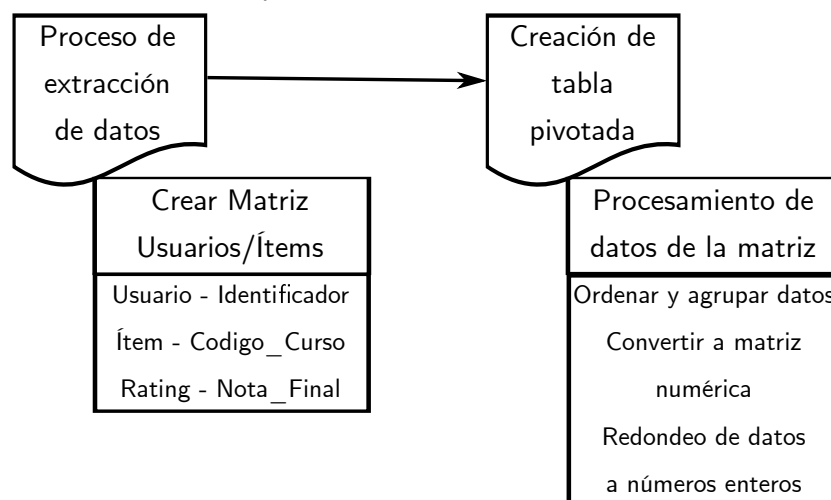
## Procesamiento de datos

El procesamiento de datos se realiza previamente al proceso de implementación del sistema de recomendación con el fin de crear un formato de datos que sea propicio

para la creación del modelo de filtrado colaborativo propuesto para el algoritmo del sistema de recomendación en su versión inicial. Este procesamiento contiene etapas bien definidas para cualquier preparación de datos al crear modelos de aprendizaje automático, que incluyen la selección de características de las matrices de entrenamiento y de prueba del sistema y el proceso de normalización de datos en cada una de ellas, tal como se relaciona en la Figura 12.

Figura 12

*Procesamiento de datos del dataframe*



*Nota.* Procesamiento de datos del dataframe. Se relaciona el proceso para el procesamiento de los datos teniendo en cuenta que es necesario la creación de una tabla pivotada que permita ordenar y agrupar los datos en una matriz de tipo numérica.

Para el proceso de extracción de la información o los datos para crear la matriz de usuarios/ítems, se tienen en cuenta las siguientes características tomadas de las bases de datos previamente relacionadas, como se identifican en la Tabla 12.

Con este dataframe se crea una tabla pivotada y se convierte, por programación, a una matriz numérica. En este paso del proceso es importante entender que el objetivo del sistema de recomendación propuesto es recomendar al usuario cursos electivos en los cuales le ha ido bien a otro estudiante con perfil similar al usuario, por tanto, se espera que el resultado a obtener en dicho curso sea similar en el nuevo estudiante. Teniendo en cuenta esta aclaración y con el fin de facilitar la comparación entre los di-

Tabla 12

*Relación de características según dataframe*

<b>Característica</b>	<b>Categoría Dataframe</b>
Usuario	Identificador
Item	Codigo-Curso
Rating	Nota-Final

*Nota.* Se presenta la relación de las categorías del Dataframe con las características de la matriz Usuarios/Ítems

ferentes estudiantes, los datos relacionados en la casilla de Nota-Final de cada curso se redondea a números de tipo entero.

De esta forma se crea un nuevo dataframe que relaciona la información indicada en la Tabla 12, trayendo estos datos desde el dataframe del programa, según corresponda, utilizando la función `copy()` de Python, que es un módulo de tipo estándar que facilita la creación de copias de distintos objetos, generalmente colecciones mutables (como las listas y los diccionarios) e instancias de clases, también mutables (Van Rossum and Drake, 2009).

Adicional a este proceso, se crea la tabla pivotada teniendo en cuenta la función `pivot_table` que es una función de Pandas que permite la acción de ordenar, agrupar, calcular y manejar datos de forma similar a la forma como se realizan estas acciones por medio del uso de hojas de cálculo con Excel (Reback et al., 2020). En esta misma línea de código se tiene que el algoritmo agrega el valor equivalente a 0 para que aquellas casillas del arreglo que se encuentren sin dato o, en su defecto, registren `NaN` que indica que el espacio está libre de dato, esto con el fin de no tener espacios vacíos y la función que permite esta acción se denomina `fillna()` que permite sustituir los valores nulos de una estructura Pandas por otro valor según ciertos criterios establecidos por el programador (Reback et al., 2020).

Con los datos que se tienen en el dataframe creado se identifica que hay varios usuarios que, aunque han matriculado en varios periodos académicos diferente número

de cursos, no han aprobado muchos cursos, lo cual genera inconvenientes en el momento de crear los arreglos matriciales para el tratamiento de los datos, por lo cual fue necesario la creación de una condicional en el algoritmo que permitiera la selección únicamente de los usuarios que tuvieran aprobados al menos 10 o más cursos en su historial académico usando la creación de una máscara. Este proceso no es necesario a medida que se van usando librerías especializadas en la realización de algoritmos para sistemas de recomendación especialmente los que utilizan la técnica de filtrado colaborativo, como es el caso de la librería **Surprise**.

En la Figura 13 se evidencia el tamaño del arreglo matricial obtenido con la ejecución del algoritmo propuesta para el sistema de recomendación, que corresponde a 104 ítems y 222 usuarios.

Figura 13

*Tamaño del arreglo matricial de dataframe ratings*

```
El tamaño de la matriz es: (222, 104)
array([[4.5, 0. , 4.8, ..., 0. , 0. , 0. ],
       [4.9, 4.9, 5. , ..., 4.5, 0. , 0. ],
       [0. , 0. , 4.4, ..., 0. , 0. , 0. ],
       ...,
       [3.9, 4.7, 5. , ..., 4.3, 0. , 0. ],
       [3.7, 3.5, 0. , ..., 0. , 0. , 0. ],
       [4.6, 3.5, 4. , ..., 4.4, 4.1, 0. ]])
```

*Nota.* Tamaño del arreglo matricial de dataframe ratings. Una vez realizado el procesamiento de los datos del dataframe, se reorganiza la matriz del mismo y se obtiene que el dataframe consta de 104 ítems para 222 usuarios relacionados.

De acuerdo con los datos obtenidos en la matriz de Usuarios/Ítems para el sistema, se observa que se analizan 104 ítems que hacen relación a los cursos que forman parte de la malla curricular del programa en mención, para el caso en particular Ingeniería Electrónica, adicionando los cursos que se encuentran avalados en el Acuerdo No. 795 de julio 30 de 2020 en donde se reconocen equivalencias de cursos entre plan antiguo y plan nuevo de dicho programa en la UNAD, así como es importante recordar que se tienen en cuenta únicamente los cursos donde el estudiante tiene una calificación

Tabla 13

*Relación algoritmos diseñados para el sistema de recomendación propuesto*

<b>Nombre del archivo</b>	<b>No. Datos</b>	<b>Procedimiento de factorización matricial usado</b>	<b>Librería Especializada</b>
RS_ELearning_Lenskit	104 ítems y 222 Usuarios	Factorización no negativa de matrices	Sklearn y Lenskit
RS_ELearning_Surprise	104 ítems y 222 Usuarios	SVD para factorización matricial	Surprise

*Nota.* Se presentan las características de los dos algoritmos diseñados.

aprobatoria, es decir, con Nota\_Final igual o superior a 3.0.

### **Diseño del sistema de recomendación**

El sistema de recomendación propuesto trabaja con la técnica de filtrado colaborativo aplicando el procedimiento de aplicación de la factorización matricial. Para el diseño del algoritmo se realizaron dos (2) propuestas obteniendo resultados similares en estos desarrollos, la diferencia radica en procedimiento o método aplicado en cada algoritmo, tal como se relaciona en la Tabla 13.

A continuación se explican de forma detallada los desarrollos relacionadas en la Tabla 13.

### *Diseño con factorización no negativa de matrices*

Para este caso, el primer paso es determinar el valor del ratio de dispersión de los datos que se tienen en el dataframe previamente procesado, para lo cual se usan las funciones de Pandas que facilita el desarrollo matemático, específicamente las funciones `np.sum` y `np.prod`, que para el caso en cuestión y con los datos obtenidos, teniendo en cuenta que se toman usuarios con más de 10 cursos aprobados, se obtiene que el ratio de dispersión equivale al 54.82 %.

En el trabajo de ciencia de datos con matrices se encuentran eventualidades en las cuales algunas de estas matrices tienen en su mayoría datos equivalentes a cero, a estas matrices se les conoce como matrices dispersas, porque son matrices de gran tamaño pero con la mayor parte de sus elementos iguales a cero. Cuando se trata de ciencia de datos, se considera ineficiente almacenar en la memoria de los dispositivos procesadores de datos elementos iguales a cero, por lo cual es necesario trabajar con matrices dispersas teniendo en cuenta que se almacenan únicamente los valores que no son cero y alguna información adicional que se considere relevante para el tratamiento de los datos.

Para el caso en cuestión, el tamaño de la matriz original es de 112 bytes y se somete a un proceso de dispersión de matriz utilizando el módulo `Scipy` que facilita la optimización de datos en matrices manejando interpolación y funciones especiales de uso en ciencia de datos (Jones et al., 2001). Con este procedimiento, la matriz dispersa ocupa 48 bytes, con un tamaño equivalente al 42.85 %.

Seguido de este proceso, se busca inicializar el modelo que permita realizar la optimización de las matrices no negativas, para lo cual es importante entender que a mayor número de características el modelo particulariza más, pero no encuentra una generalización del mismo. Para este caso, se busca la forma de encontrar dos matrices no negativas ( $W$ ,  $H$ ) cuyo producto se aproxime a la matriz no negativa  $X$ , tal como lo establece la teoría de Factorización matricial no negativa (NMF) (Lopez-Nores et al.,

2012). Esta factorización se puede utilizar, por ejemplo, para la reducción de dimensionalidad, la separación de fuentes o la extracción de temas. La función objetivo que permite realizar esta acción está determinada por la ecuación

$$\begin{aligned}
 & 0,5 * \|X - WH\|_{\text{Fro}}^2 + \text{alpha} * l\_ratio * \|\text{vec}(W)\|_1 \\
 & \quad + \text{alpha} * l1\_ratio * \|\text{vec}(H)\|_1 \\
 & \quad + 0,5 * \text{alpha} * (1 - l\_ratio) * \|W\|_{\text{Fro}}^2 \\
 & \quad + 0,5 * \text{alpha} * (1 - l1\_ratio) * \|H\|_{\text{Fro}}^2
 \end{aligned} \tag{12}$$

En donde se tienen los siguientes parámetros con su respectiva configuración (Pedregosa et al., 2011):

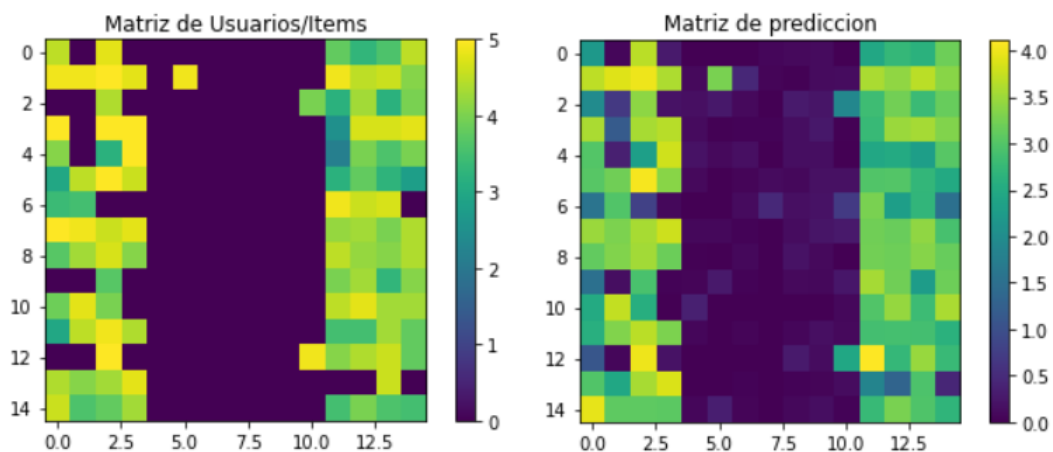
- *n\_components*: Corresponde al número de componentes o características que se usarán en la matriz a crear, para el caso en particular se configura con un valor equivalente a 30.
- *init*: Representa el método utilizado para inicializar el procedimiento de NMF y está configurada en el caso particular que se inicia de manera aleatoria.
- *random\_state*: Éste parámetro se utiliza para la inicialización del procedimiento NMF, en especial cuando *init* se configura de manera aleatoria y en el caso particular se configura equivalente a 0.
- *alpha*: Variable de tipo flotante que representa una constante que multiplica los términos de regularización para el procedimiento de NMF y para el caso en particular se configura con un valor equivalente a 1.
- *l1\_ratio*: Éste parámetro de mezcla de la regularización se configura de tal manera que cumpla con la condición que  $0 \leq l1\_ratio \leq 1$ . Para  $l1\_ratio = 0$ , también se tiene en cuenta la penalización representada por L2 y también es conocida como Norma Frobenius, cumpliendo esta condición en el caso en particular se usa con un valor equivalente a 0.1.
- *max\_iter*: Éste parámetro representa el número de iteraciones a utilizar en el procedimiento de NMF y para el caso en particular se configura con un valor de 2000 iteraciones.

De esta manera se obtiene una matriz de predicción de ratings con la cual se

espera generar la recomendación de cursos a los usuarios del sistema. Python tiene la particularidad de plotear las matrices para entender la similitud de ellas de forma gráfica y como es de esperarse, van a tener un grado de similitud entre ellas, tal como se evidencia en la Figura 14.

Figura 14

*Gráfica de comparación Matriz de Usuarios/Ítems y Matriz de predicción de acuerdo con el modelo diseñado*



*Nota.* Gráfica de comparación Matriz de Usuarios/Ítems y Matriz de predicción. Se relacionan las matrices obtenidas tanto del dataframe de Usuarios/Ítems (izquierda) y de la predicción que realiza el algoritmo propuesto en el modelo 1 (derecha).

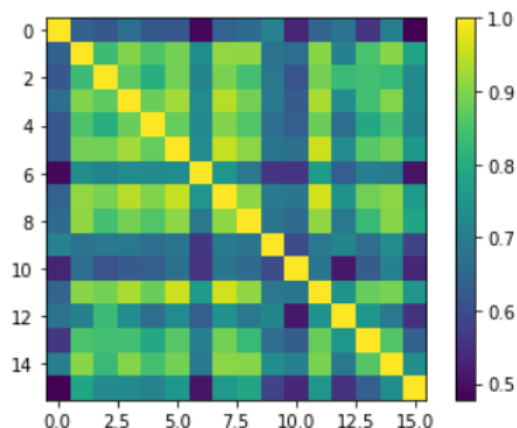
Como se observa, entre la Matriz de Usuarios/Ítems y la Matriz de predicción hay bastante similitud, de lo cual se puede concluir que las bases obtenidas con el procedimiento de NMF representan partes localizadas de la matriz original, y se destaca en este caso que la representación de datos mediante este procedimiento NMF genera bases y matrices de codificación que serán dispersas, evaluando este resultado con la técnica de coseno de similitud con la función de Python `cosine\_similarity` que permite la comparación entre la distancia/similitud por pares de las muestras en las matrices donde se aplica, como se puede observar en la Figura 15.

Una vez verificado el resultado obtenido y el nivel de similitud, es necesario convertir la matriz de predicción en un dataframe para poder procesar los datos y de es-



Figura 15

*Comportamiento gráfico de respuesta a validación con coseno de similitud*



*Nota.* Comportamiento gráfico de respuesta a validación con coseno de similitud. Se relaciona el diagrama que representa los resultados obtenidos con el análisis del coseno de similitud entre las dos matrices obtenidas en el modelo 1, evaluando la distancia/similitud entre los datos obtenidos.

ta forma proceder a verificar las predicciones de manera particular, es decir, para cada usuario o estudiante teniendo en cuenta el Identificador que diferencia a cada usuario en particular, que en este caso hace referencia a cada estudiante y una vez se tiene la fila de predicciones del estudiante analizado, se procede a extraer los cursos electivos del programa, como lo son cursos electivos disciplinares y electivos de profundización, que son el objetivo de la recomendación y representa el valor de cercanía a la recomendación de cada curso según el historial del estudiante y del análisis de la predicción obtenida.

De esta manera, para el estudiante con Identificador “Estudiante 63” se relacionan 51 cursos electivos y cada uno con un valor de similitud diferente de acuerdo con la predicción, ordenados de izquierda a derecha, con la particularidad que los de mayor similitud y se estarían recomendando al estudiante se encuentran ubicados en las columnas de la derecha, es decir, el orden es ascendente de izquierda a derecha en cuanto a la recomendación, tal como se evidencia en la Figura 16.

El siguiente paso es entrenar el modelo obtenido con la matriz de predicción pa-

Figura 16

*Relación de recomendación para el ejemplo citado*

Codigo_Curso	2150521	302090	243011	301121	203053	208055	201101	299012	203043	10200
Estudiante 63	0.0	0.0	0.0	0.0	0.000802	0.003266	0.008497	0.014553	0.022338	0.02390
1 rows × 11 columns										
...	103380	203044	299011	299013	203054	203049	203055	203050	208007	203052
...	0.410365	0.508379	0.589006	0.869952	2.32018	2.639262	2.669351	2.76233	2.791404	3.59318

*Nota.* Relación de recomendación para el ejemplo citado. El algoritmo finaliza con el proceso de entrega de las recomendaciones para el usuario y se entrega el código del curso recomendado, evidenciando que se sugieren 51 cursos obteniendo el de mayor afinidad al costado derecho de la relación de cursos.

ra lo cual se crean dos conjuntos de datos que se conocen en los sistemas de recomendación como la matriz de entrenamiento o Train y la matriz de prueba o Test, teniendo en cuenta la programación relacionada en la Figura 17.

Figura 17

*Programación para crear matrices de entrenamiento y de prueba*

```
def split_RatingMatrix(X, NumTest = 0.1):
    XF = X.copy()
    np.random.seed(4)
    m = XF.shape[0]
    n = XF.shape[1]
    XM_test = np.ones((m,n))
    XM_train = np.ones((m,n))
    j = 0
    for row in XF:
        NonZeroCells = np.sum(row!=0)
        # Garantizar que al menos se elimine un valor por fila
        CellToEliminate = max(np.round(NonZeroCells*NumTest),1)
        while CellToEliminate>0:
            NumAle = np.random.randint(0,n)
            if XF[j,NumAle] != 0:
                XM_train[j,NumAle] = 0
                XF[j,NumAle]=0
                CellToEliminate-=1
            j+=1
    return XM_train, XM_test
```

*Nota.* Programación para crear matrices de entrenamiento y de prueba. Se relaciona el segmento del algoritmo que se encarga de la creación de las matrices de entrenamiento y de prueba para evaluar el modelo.

Este proceso se realiza tomando varios datos de configuración en las característi-

cas del procedimiento de NMF, teniendo que el mejor comportamiento se obtiene cuando se trabaja con 60 componentes, un factor de penalización  $\alpha = 0.1$  y un factor de aprendizaje  $= 0.1$ , como se evidencia en Figura 18:

Figura 18

*Programación para evaluar las matrices de entrenamiento y de prueba*

```

from sklearn.metrics import mean_squared_error
UserItemTrain_Mask,UserItemTest_Mask = split_RatingMatrix(UserItem,0.1)
UserItemTrain = UserItem*UserItemTrain_Mask
UserItemTest = UserItem*UserItemTest_Mask
# Creamos el objeto NMF con p=20, un factor de penalizacion alpha=0.1 y un factor de aprendizaje = 0.1
model = NMF(n_components=60, init='random', random_state=0, alpha=0.1, l1_ratio=0.1,max_iter=2000)
W = model.fit_transform(UserItemTrain)
H = model.components_
print(model.reconstruction_err_)
all_user_predicted_ratings = np.dot(W,H)
#all_user_predicted_ratings = (all_user_predicted_ratings/(np.max(all_user_predicted_ratings)))*5
#ErrorRMSE_Train = np.linalg.norm(UserItemTrain-all_user_predicted_ratings)
ErrorRMSE_Train = mean_squared_error(UserItemTrain,all_user_predicted_ratings)
print("El error cuadrático medio es {}".format(ErrorRMSE_Train))
ErrorRMSE_Test = mean_squared_error(UserItemTest,all_user_predicted_ratings)
print("El error cuadrático medio es {}".format(ErrorRMSE_Test))

```

*Nota.* Programación para evaluar las matrices de entrenamiento y de prueba. Se relaciona el segmento del algoritmo que evalúa las matrices creadas para entrenamiento y prueba teniendo en cuenta los mejores datos configurados para los hiper parámetros de las funciones respectivas.

Como se evidencia en la Figura 18, se halla el error cuadrático medio para las dos matrices de entrenamiento y prueba del modelo propuesto, de donde se obtiene que el error cuadrático medio RMSE de la matriz de entrenamiento (Train) es 0.2131 y el error cuadrático medio RMSE de la matriz de prueba (Test) es 0.7172.

Como desventaja, se tiene que el tiempo de ejecución de este algoritmo es alto teniendo en cuenta las diferentes iteraciones de pruebas que debe realizar para obtener los datos de las matrices de predicción, de entrenamiento y de prueba. Adicional, es un algoritmo que requiere de características avanzadas del sistema en el cual se va a poner a prueba su funcionamiento, es decir, lo que se refiere a las características de máquina.

Este modelo se optimiza utilizando una librería especializada de Python conocida como `Lenskit` que cuenta con un conjunto de herramientas específicas para ser utilizadas en algoritmos propios de sistemas de recomendación, especialmente en algoritmos que implementan la técnica de filtrado colaborativo (Ekstrand, 2018), como es el caso en cuestión.

Como en todo algoritmo, es necesario importar las librerías que se necesitan para el correcto funcionamiento de la lógica de programación propuesta, que cuenta con diferentes herramientas que se encargan de optimizar funciones específicas, a continuación se puede observar en la Tabla 14 la forma como se relacionan los módulos específicos que se tienen en cuenta para este diseño.

Teniendo en cuenta que la matriz está conformada en su mayoría por ceros es importante mejorar la forma de almacenamiento que se tiene en dicha matriz. Para lo cual, en primer lugar se procede a calcular la dispersión de la matriz de usuarios/ítems y se utiliza el módulo de la librería `Lenskit` conocido como `Lenskit.metrics.predict` de donde se importa la herramienta `rmse` que permite al algoritmo desarrollar acciones con modelos que implementan el modelo de factorización de matriz sesgada de tipo estándar, que se conoce como `Lenskit.algorithms.als.BiasedMF` y esta función tiene como característica principal que aprende los parámetros del modelo mediante el descenso de gradiente en lugar del algoritmo de mínimos cuadrados alternos (Ekstrand, 2020).

La ventaja que tiene esta implementación de mejora y optimización del algoritmos es que usa `Lenskit.algorithms.als.Bias` para calcular los sesgos. Una vez realizado este proceso, extrae las matrices resultantes y se basa en `MFPredictor` para implementar la lógica de predicción, por medio del módulo conocido como `Lenskit.algorithms.als.BiasedMF` (Ekstrand, 2020). En el momento de implementar este módulo es necesario tener en cuenta la configuración de los datos que se requieren para su correcto funcionamiento y a su vez, para garantizar el mejor modelo de lógica de predicción del sistema y para el caso en desarrollo se tiene la relación que se presenta en la Tabla 15 (Zhou et al., 2008).

Esta librería tiene una función particular para la evaluación de los resultados obtenidos y es que se basa en el pos procesamiento de la salida de recomendadores y predictores, teniendo en cuenta que para este tipo de operaciones es importante y fundamental incluir los valores de las calificaciones asignadas en el marco de predicción

Tabla 14

*Módulos específicos de librería Lenskit usados en el algoritmo*

<b>Librería</b>	<b>Módulo</b>	<b>Características</b>	<b>Referencia</b>
Lenskit	crossfold as xf	Facilita la medición del comportamiento de los modelos de entrenamiento y prueba	(Ekstrand, 2020)
Lenskit.algorithms	mf_common, Recommender, als, item_knn as knn	Funciones que proporcionan implementaciones de filtrado colaborativo k-NN clásicas basadas en ítems y usuarios	(Ekstrand, 2020)
Lenskit	util, batch	Funciones que permiten generar muchas recomendaciones o predicciones en el mismo tiempo, útiles para evaluaciones y experimentos	(Ekstrand, 2020)
Lenskit.metrics	rmse	Módulo que contiene métricas de precisión útiles para procesos de predicción	(Ekstrand, 2020)

*Nota.* Se relacionan los módulos de la librería Lenskit que se requieren para el correcto funcionamiento del algoritmo propuesto.

Tabla 15

*Características de configuración de implementaciones alternas de mínimos cuadrados de factorización matricial*

<b>Parámetro</b>	<b>Características</b>	<b>Dato configurado</b>
Features	Variable de tipo entero que representa la cantidad de funciones para entrenar	50
Iterations	Variable de tipo entero que representa el número de iteraciones para entrenar	10000
Method	Variable de tipo caracter que representa el parámetro del paso de optimización a utilizar por medio del método de descomposición	Lu
Bias	Variable de tipo booleano que representa el ajuste a un sesgo con amortiguación	False
Reg	Variable de tipo flotante que representa el factor de regularización; también puede ser una tupla (ureg, ireg) para especificar términos de regularización de elementos y usuarios separados	(0.1,0.1)

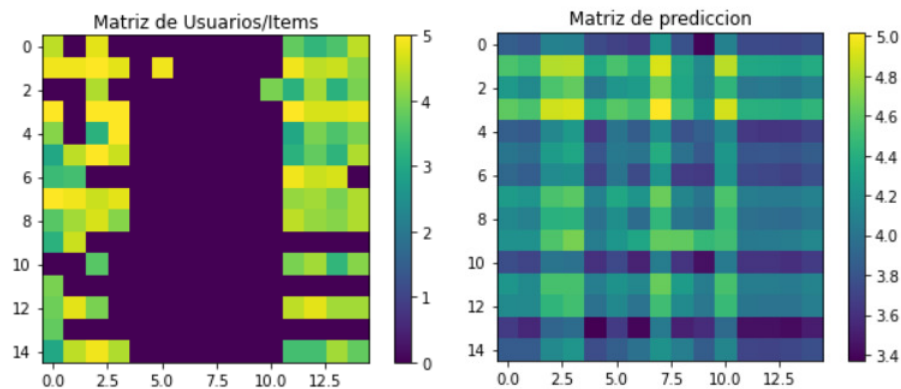
*Nota.* Se presentan las características de configuración de parámetros para la implementación de mínimos cuadrados de factorización matricial en el sistema.

cuando sus pares de usuario-ítem de entrada contienen datos de calificación, para lo cual usan las funciones métricas de predicción RMSE o error cuadrático medio y MAE o error absoluto medio, que para el caso en particular el valor obtenido para el cálculo de RMSE corresponde a 0.472.

Como complemento a este desarrollo, es necesario aclarar que la librería Lenskit no genera valor de las operaciones de predicción cuando el modelo central no puede predecir, es decir, un recomendador de vecino más cercano no puntuará un elemento si no puede encontrar ningún vecino adecuado de acuerdo con las características e información relacionada para cada ítem de la base de datos. Esto genera que los datos de la predicción, aunque se tenga una métrica dentro de los rangos de valor recomendados para estos sistemas, no sean muy cercanos en comparación con el modelo que se obtiene de la Matriz de Usuarios/Ítems con respecto a la Matriz de predicción, como se observa en la Figura 19, porque el modelo de predicción no se está probando en el mismo conjunto de elementos que conforman la base de datos.

Figura 19

*Gráfica de colores de comparación Matriz de Usuarios/Ítems y Matriz de predicción*



*Nota.* Gráfica de comparación Matriz de Usuarios/Ítems y Matriz de predicción. Se relacionan las matrices obtenidas tanto del dataframe de Usuarios/Ítems (izquierda) y de la predicción que realiza el algoritmo propuesto en el modelo 1 (derecha).

Cuando se presentan estas situaciones, Lenskit tiene funciones que le permiten evaluar las N listas de recomendaciones principales de acuerdo con la funcionalidad de tres métricas conocidas como Métricas de clasificación que procesa la lista de recomen-

daciones como una clasificación de elementos relevantes calculando la precisión de la recomendación y la recuperación de recomendaciones, Métricas de lista clasificada que procesan la lista de recomendaciones como una lista clasificada de elementos que pueden ser relevantes o no cuando se calcula la clasificación recíproca del primer elemento relevante en una lista de recomendaciones y Métricas de utilidad que estima una puntuación de utilidad para una lista clasificada de recomendaciones realizando el cálculo de la ganancia acumulada descontada normalizada (Ekstrand, 2020).

### *Diseño con SVD para factorización matricial*

En el segundo desarrollo del algoritmo propuesto, se trabaja con una librería especializada de Python conocida como **Surprise** que corresponde a un scikit dedicado a sistemas de recomendación específicamente, realizando acciones de validación cruzada aplicando la técnica de SVD para factorización matricial (Hug, 2020b).

Al igual que sucede en el caso anterior y en cualquier algoritmo, el primer paso corresponde a importar las librerías que se requieren para la correcta ejecución del programa y se relacionan los respectivos módulos, de esta manera adicional a la importación de Pandas y Numpy, es necesario importar la librería **Surprise** de donde se tienen en cuenta los módulos relacionados en la Tabla 16.

A continuación, se procede a extraer la información o los datos para crear la matriz de usuarios/ítems que estaría conformado por Usuario (Identificador), Ítem (Codigo\_Curso) y Rating (Nota\_Final). Con este dataframe se crea una tabla pivotada y es convertida a una matriz de tipo numérico, en donde es importante aclarar que para facilitar la comparación entre los diferentes estudiantes que conforman el universo, las notas finales que corresponden al Rating, se redondean a números de tipo entero.

Al igual que el algoritmo usado en el modelo anterior, se tiene que el tamaño de la matriz es de 104 ítems y 222 usuarios. Python tiene una función que permite conocer los valores estadísticos para la matriz de rating y para el caso en particular, dichos



Tabla 16

*Módulos específicos de librería Surprise usados en el algoritmo*

<b>Librería</b>	<b>Módulo</b>	<b>Características</b>	<b>Referencia</b>
Surprise	Dataset	Módulo utilizado para definir la clase de datos a usar y subclases que se utilizan para administrar y organizar	(Hug, 2020b)
Surprise	Reader	Función usada para analizar un archivo que contiene los ratings del sistema	(Hug, 2020b)
Surprise	SVD, SVDpp, BaselineOnly	Funciones que se requieren para la configuración y desarrollo de los algoritmos basados en factorización de matrices con técnica SVD	(Hug, 2020b)
Surprise	accuracy	Módulo que proporciona herramientas para calcular métricas de precisión en un conjunto de predicciones como rmse, mse, mae o fcp	(Hug, 2020b)
Surprise.model	GridSearchCV, RandomizedSearchCV, train_test_split	Módulo que contiene herramientas para ejecutar procedimientos de validación cruzada	(Hug, 2020b)

*Nota.* Se presentan los módulos de la librería Surprise que se requieren para el correcto funcionamiento del algoritmo y se especifican sus principales características.

datos se encuentran relacionados en la Figura 20, en donde se evidencian los valores correspondientes de la Media de rating, la Mediana de rating, la Suma de rating, los percentiles de rating (25 %, 50 % y 75 %) y los valores de MAX y MIN de rating.

Figura 20

*Datos estadísticos de la matriz de rating*

	<b>user</b>	<b>item</b>	<b>rating</b>
<b>count</b>	8.334000e+03	8.334000e+03	8334.000000
<b>mean</b>	6.235131e+08	3.024550e+05	4.093964
<b>std</b>	5.138312e+08	4.520558e+05	0.567975
<b>min</b>	2.717811e+06	4.000200e+04	3.000000
<b>25%</b>	7.979267e+07	1.004120e+05	3.600000
<b>50%</b>	1.030665e+09	2.030420e+05	4.100000
<b>75%</b>	1.100400e+09	2.430060e+05	4.600000
<b>max</b>	1.144164e+09	2.150521e+06	5.000000

*Nota.* Datos estadísticos de la matriz de rating. Se relacionan los datos de la matriz de rating correspondientes a la media y mediana de rating, la suma del rating, los percentiles correspondientes y valores máximos y mínimos en comparación con las matrices de usuarios e ítems.

Teniendo en cuenta que este algoritmo utiliza la técnica de SVD para factorización matricial, es importante recordar que este procedimiento consiste en dividir la matriz de usuarios/ítems en dos matrices, una para los usuarios y las características latentes del problema en cuestión y la segunda corresponde a ítems y características latentes. Según lo indica la teoría, en este caso se trabaja en base a la siguiente representación matemática:

$$\hat{r}_{ui} = \mu + b_i + b_u + q_i^T p_u \quad (13)$$

Es importante tener presente que la variable  $\mu$  corresponde al promedio global de todas las ponderaciones de la matriz, y que las desviaciones observadas o bias del ítem y usuario están representadas por las variables  $b_i$  y  $b_u$  respectivamente. Por lo tanto, y dando cumplimiento con lo que indica la teoría, se determinan estos parámetros me-

diante un problema de optimización según lo indica la siguiente identidad:

$$\min_{b_*, q_*, p_*} \sum_{(u,i) \in X} (r_{ui} - \hat{r}_{ui})^2 + \lambda (b_i^2 + b_u^2 + \|q_i\|^2 + \|p_u\|^2) \quad (14)$$

En donde se tiene que la variable  $\lambda$  corresponde al factor de regularización y se calcula por medio de la validación cruzada. lo que ayuda a prevenir que se presenta el efecto overfitting cuando se realiza la regularización de los parámetros aprendidos.

Para cumplir con esto que indica la teoría se utiliza la librería **Surprise**, para lo cual, como en cualquier algoritmo, el primer paso es convertir el dataframe que se tiene para ratings en la clase de Dataset necesaria para trabajar con esta librería, para lo cual es importante indicar la escala de la calificación y corresponde a un valor comprendido entre 1 y 5, utilizando la línea de código correspondiente a `reader = Reader(rating_scale=(1, 5))`. Finalmente, se programa el sistema para que proceda a crear la estructura de datos de la clase "surprise.dataset.DatasetAutoFolds", según lo establece la teoría existente para el uso de esta librería, teniendo que este nuevo dataset debe cumplir con la condición que las columnas deben corresponder al identificador del usuario, la identificación del ítem y las calificaciones (en ese orden).

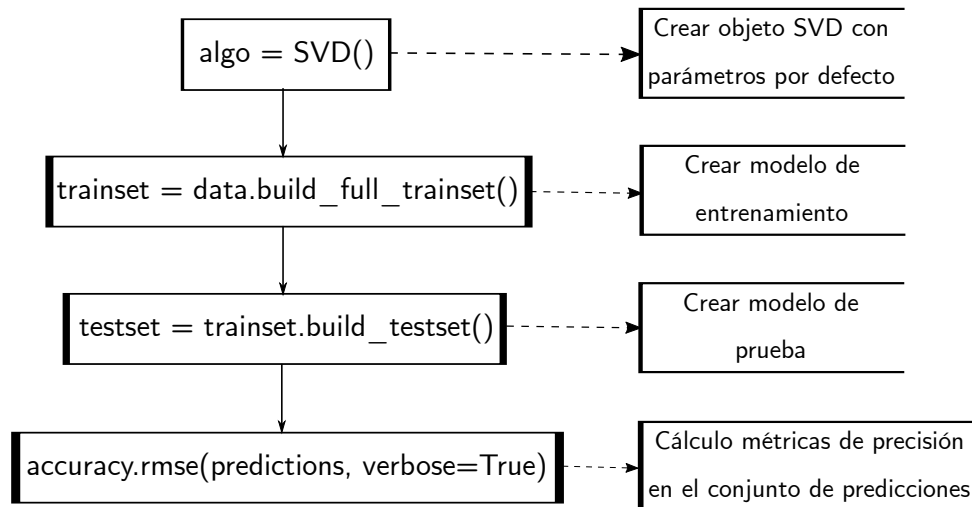
Continuando con el proceso, el siguiente paso corresponde a la creación del objeto SVD para proceder así, a entrenar el sistema, para lo cual se usa el módulo SVD que es la función equivalente a la factorización matricial probabilística (Salakhutdinov and Mnih, 2008). Para este proceso, se procede como lo indica la Figura 21.

De esta manera, en el proceso de validación del modelo por medio de RMSE, se obtiene que el valor obtenido corresponde a 0.3992 y se cumple la condición que RMSE debe ser bajo ya que en este proceso los datos y el algoritmo está sesgado. De igual manera se puede concluir que el algoritmo fue entrenado con la totalidad de los datos y adicionalmente ningún hiper parámetro del modelo fue seleccionado.

El siguiente paso es realizar la validación cruzada para definir los hiper parámetros de configuración del objeto SVD y de esta forma proceder a generar una precisión del modelo pero esta vez sin bias. En ese orden de ideas, primero se divide el conjun-

Figura 21

*Procedimiento para crear el objeto SVD y entrenamiento del sistema*



*Nota.* Procedimiento para crear el objeto SVD y entrenamiento del sistema. Se relaciona el diagrama de flujo que representa el procedimiento de creación del objeto SVD, los modelos de entrenamiento y de prueba y el cálculo de métricas de precisión para el conjunto de predicciones.

to de datos en dos modelos distribuidos de la siguiente manera: 80 % para el modelo de entrenamiento y 20 % para el modelo de prueba. Una vez se tienen estos nuevos modelos y que cumplan con estas condiciones, con el conjunto de entrenamiento se realiza el procedimiento de validación cruzada utilizando los módulos `RandomizedSearchGrid` y `k-fold`, siendo `k=5`.

Para realizar la validación cruzada se evalúan una serie de parámetros en unos rangos determinados con el fin de obtener los mejores valores de los hiper parámetros de configuración del objeto SVD, tal como se relacionan en la Tabla 17

Los parámetros que se están midiendo en el rendimiento corresponden a RMSE (Error cuadrático medio) y MAE (Error medio absoluto). Finalmente, se toman 1000 muestras de forma aleatoria. La variable `n_jobs=-1` indica al sistema paralelizar el proceso con todos los núcleos del computador a usar, como se observa en la Figura 22.

La función `RandomizedSearchCV` se encarga de calcular métricas de precisión

Tabla 17

*Configuración de parámetros para la validación cruzada*

<b>Parámetro</b>	<b>Características</b>	<b>Rango de datos</b>
<code>n_factors</code>	Variable de tipo entero que representa el número de factores a utilizar y se recomienda un valor que no supere 100	Entre 40 y 80
<code>n_epochs</code>	Variable de tipo entero que representa el número de iteraciones del procedimiento SVD	Entre 1000 y 3000
<code>lr_all (learning rate)</code>	Variable de tipo flotante que representa la tasa de aprendizaje de todos los parámetros	Entre 0.001 y 0.01
<code>reg_all (lambda)</code>	Variable de tipo flotante que representa el término de regularización para todos los parámetros	Entre 0.1 y 1

*Nota.* Se presentan las características de los hiper parámetros que se requieren para el procedimiento de validación cruzada para el segundo algoritmo y los rangos de valores necesarios para su correcta configuración.

Figura 22

*Procedimiento para selección automática de parámetros de validación cruzada*

```

param_grid = {'n_factors':list(range(40,80,1)), 'n_epochs': [1000,2000,3000], 'lr_all': np.arange(0.001, 0.01, 0.001).tolist(),
              'reg_all': np.arange(0.1, 1, 0.1).tolist()}
gs = RandomizedSearchCV(SVD, param_grid,n_iter=1000,measures=['rmse', 'mae'], cv=5,joblib_verbose=1,n_jobs=-1)
gs.fit(data)

algo = gs.best_estimator['rmse']

```

*Nota.* Procedimiento de selección automática de parámetros de validación cruzada. Se relaciona el segmento del algoritmo que se representa las funciones que determinan de forma automática las parámetros de configuración para la validación cruzada.

para un algoritmo en varias combinaciones de parámetros, a través de un procedimiento de validación cruzada (Hug, 2020b). Esta función se diferencia de `GridSearchCV` que utiliza un enfoque combinatorio exhaustivo, por lo tanto, la función `RandomizedSearchCV` muestra de forma aleatoria el espacio de los parámetros de acuerdo con los rangos establecidos por programación. Este proceso es útil cuando se requiere encontrar el mejor conjunto de parámetros para un algoritmo de predicción, especialmente si se usa un enfoque de grueso a fino.

De la ejecución de la lógica de programación relacionada en la Figura 22 se evidencia que el mejor resultado obtenido para el mejor modelo encontrado, de acuerdo a la métrica `rmse` (error cuadrático medio), corresponde a un modelo cuyos parámetros de configuración corresponden a `parameters\_dic= n\_factors = 69, n\_epochs = 1000, lr\_all = 0.001, reg\_all = 0.1`

Tal como lo establece la teoría, el siguiente paso corresponde a la validación de estos hiper parámetros con el entrenamiento del modelo con estos parámetros y de esta forma analizar la precisión con respecto a los datos de testeo, para lo cual se sigue el mismo procedimiento establecido en la Figura 21. De este procedimiento se aclara que el objetivo es ajustar los mejores parámetros del algoritmo, para lo cual se tienen en cuenta dos conjuntos A y B, en donde A se usa para el ajuste de parámetros usando la búsqueda de cuadrícula, mientras que B se usa para la estimación sin sesgo, como se relaciona en la Figura 23.

Figura 23

*Lógica de programación para análisis de precisión con respecto a los datos de testeo en el modelo presentado*

```
# Compute biased accuracy on A
predictions = algo.test(trainset.build_testset())
print('Biased accuracy on A,', end=' ')
accuracy.rmse(predictions)

# Compute unbiased accuracy on B
testset = data.construct_testset(B_raw_ratings) # testset is now the set B
predictions = algo.test(testset)
print('Unbiased accuracy on B,', end=' ')
accuracy.rmse(predictions)
```

*Nota.* Lógica de programación de análisis de precisión en datos de testeo. Se relaciona el segmento del algoritmo propuesto que representa la forma de hallar los datos de precisión del sistema evaluados en la matriz de prueba.

De la ejecución de estas líneas de código y como resultado de la verificación del modelo con los mejores parámetros obtenidos para el sistema, se tiene que la precisión sesgada en el conjunto A arroja un resultado correspondiente de RMSE equivalente a 0.4316, mientras que la precisión sin sesgo en el conjunto B da como resultado correspondiente de RMSE un valor equivalente a 0.4696, concluyendo que los datos obtenidos están muy cercanos entre si y no están muy lejanos al valore RMSE obtenido con el objeto SVD creado sin hiper parámetros configurados previamente.

Una de las características que tiene el algoritmo implementado es que predice estimaciones de referencia para cada usuario y cada ítem, lo que se conoce como recomendación **BaseLine** y su principal función es la predicción de la estimación de la línea de base para un usuario y artículo determinados y la estimación de referencia para una calificación desconocida  $r_{ui}$  se denota como  $b_{ui}$  y tiene en cuenta los efectos del usuario y el ítem, como se relaciona en la ecuación 3.6:

$$\hat{r}_{ui} = b_{ui} = \mu + b_u + b_i \quad (15)$$

En donde se tiene que  $\mu$  es la calificación promedio general, los parámetros  $b_u$  y  $b_i$  indican la desviación observada para el usuario  $u$  y el ítem  $i$  respectivamente. En el caso particular donde el usuario  $u$  sea desconocido entonces se supone que el sesgo  $b_u$  es

Tabla 18

*Configuración de parámetros para BaseLine*

Parámetro	Características	Valor configura- do
method	Representa el método a usar en el algoritmo, puede configurarse por defecto como <i>als</i> o <i>sgd</i> y dependiente de esta selección, se seleccionan los datos y rangos de los demás parámetros	<i>als</i>
n\_epochs	Variable de tipo entero que representa el número de iteraciones del procedimiento ALS	100
reg_u	Variable de tipo flotante que representa el valor de regularización para usuarios	0.01
reg_i	Variable de tipo flotante que representa el regularización para ítems	0.01

*Nota.* Se presentan las características y valores configurados de los parámetros requeridos para aplicar *BaseLine* en el algoritmo 2.

ceros. Lo mismo se aplica al ítem  $i$  con  $b_i$  (Koren, 2010).

En la configuración de la función que permite trabajar *BaseLine*, es decir la función especializada *BaselineOnly*, se tiene la configuración del hiper parámetro conocido como *bsl\_options* que es un diccionario de opciones para el cálculo de estimaciones de línea base, en el cual se configuran los parámetros relacionados en la Tabla 18.

Una vez configurados los parámetros de línea base, se realiza de nuevo el reentrenamiento de los dos conjuntos, A y B, siendo A el conjunto de entrenamiento y el conjunto en el cual se calcula por código la precisión sesgada, mientras que B es el conjunto de prueba y se calcula con este conjunto la precisión sin sesgo. De esta manera se



tiene como resultado de la verificación del modelo en línea base y con los parámetros indicados en la Tabla 18, se tiene que la precisión sesgada en el conjunto A arroja un resultado correspondiente de RMSE equivalente a 0.4559, mientras que la precisión sin sesgo en el conjunto B da como resultado correspondiente de RMSE un valor equivalente a 0.4790, concluyendo que los datos obtenidos están muy cercanos entre si y no están muy lejanos al valore RMSE obtenido con el objeto SVD creado con los mejores valores para la configuración de los hiper parámetros.

De acuerdo con los valores obtenidos, el siguiente paso es realizar las recomendaciones de cursos electivos, para lo cual es necesario hallar el valor correspondiente al error cuadrático medio RMSE teniendo en cuenta la precisión sesgada en los datos realizando el proceso de validación cruzada utilizando los mejores parámetros hallados para la función SVD y que corresponden a `n_factors = 69`, `n_epochs = 1000`, `reg_all = 0.1` y `lr_all = 0.001` y se realiza el entrenamiento en todos los datos del sistema, realizando también la prueba de los elementos que no están calificados, obteniendo un valor de la precisión sesgada en los datos determinada por la métrica RMSE equivalente a 0,3559.

Finalmente, se realiza la predicción para todos los usuarios que conforman la base de datos del sistema de recomendación y se ordenan las predicciones para cada usuario y recupera las  $k$  más altas, según el caso de la programación, el valor asignado como prueba para  $k$  es equivalente a 10.

Siguiendo el caso de ejemplo utilizado en el primer modelo desarrollado y presentado, se tiene que para el estudiante con Identificador .Estudiante 63"los 10 cursos con predicción más alta se relacionan en la Figura 24.

De esta manera se entregaría la relación de cursos recomendados al estudiante, teniendo en cuenta su perfil y la información de los diferentes usuarios que comprenden la base de datos del sistema y que conforman el universo del sistema y teniendo en cuenta la afinidad con el perfil particular.

Figura 24

*Relación de recomendación para el ejemplo citado*

Estudiante 60	[700004, 80012, 203045, 80003, 80002, 208006, 301402, 712001, 2150506, 1020
Estudiante 61	[700004, 80012, 203045, 208006, 208055, 243011, 203043, 2150506, 80002, 800
Estudiante 62	[700004, 434206, 80012, 80003, 80020, 208055, 80002, 208018, 221120, 215050
Estudiante 63	[80012, 80003, 203045, 80020, 208055, 221120, 712001, 208006, 102015, 20303
Estudiante 64	[700004, 80012, 203045, 80020, 80003, 208006, 80002, 208055, 2150506, 71200
Estudiante 65	[700004, 80012, 203045, 80020, 208006, 301402, 80003, 208055, 2150506, 1020
Estudiante 66	[700004, 80012, 203045, 80003, 80020, 80002, 208055, 208006, 2150506, 10201

*Nota.* Relación de recomendación para el ejemplo citado. Se relacionan los 10 códigos de cursos de mayor valor en precisión con respecto a la predicción como sugerencia al usuario analizado, aclarando que el dato del número de cursos recomendados es configurable por código según lo desee el usuario.

## Análisis de resultados

### Resultados

Los resultados que se obtienen a partir del desarrollo del trabajo realizado responden a la metodología de validación descrita en el capítulo anterior, en donde se presentan los dos modelos propuestos para el sistema de recomendación con su respectiva evaluación comparativa teniendo en cuenta las métricas de escalabilidad en tiempo y capacidad de uso de recursos de máquina para el procesamiento, así como también la precisión y el RMSE de cada modelo propuesto.

Las pruebas de ejecución y procesamiento de cada modelo se ejecutan en Jupyter Notebook: 0.7.2 con la versión de Python 3.5.2 en un ambiente Windows 10 cuyo hardware cuenta con un procesador Intel Core i7 a 1.5GHz, 8 núcleos y 8 GB de Ram DDR4.

La base de datos del sistema se divide en tres archivos de Excel en formato `xlsx` y cada uno cuenta con una estructura, organización y contenido diferente, pero con información que los relaciona entre si. Esta particularidad lleva a que se cuenta con una base de datos diversa en cuanto a información y escenarios para la validación de los resultados que cada prototipo propuesto procesa con la aplicación de diferentes métodos o técnicas.

Se tienen dos modelos propuestos, cada uno con resultados diferentes pero con valores obtenidos de sus métricas dentro de los rangos aceptables para los sistemas de recomendación. La información que comprende la base de datos del sistema de recomendación propuesto y del desarrollo presentado en el capítulo anterior presenta unas

Tabla 19

*Información de la base de datos del sistema*

<b>DataFrame</b>	<b>Valores</b>	<b>Características</b>
HistoriaAcademica	11610/7	11610 datos relacionados en 7 columnas
Electronica	238/4	238 datos de Usuarios relacionados en 4 columnas
Cursos	234/7	234 datos de Ítems relacionados en 7 columnas

*Nota.* Se relacionan los dataframes usados y las características de cada uno.

características que se relacionan en la Tabla 19.

Adicional, se tiene que el dataframe utilizado correspondiente a la Matriz de Usuarios/Ítems utilizada, tiene un total de 222 Usuarios y 104 Ítems y se relacionan únicamente estudiantes del programa de Ingeniería Electrónica Resolución 13155.

Se tiene la particularidad, que en algunos casos, no todos los usuarios han matriculado los mismos cursos que otros, es decir, al tener un total de 104 ítems genera una amplia relación de cursos que no todos los 222 usuarios han matriculado, como se refleja en su respectivo historial académico. Esta situación presenta que algunos ítems van a tener calificación nula y que fue procesada en el algoritmo, de tal manera que la dispersión del desarrollo y los resultados obtenidos no fuera tan elevada, utilizando funciones especiales de las librerías usadas en los tres modelos propuestos.

Cada archivo de datos es usado en los dos modelos de algoritmos propuestos, iniciando con el método de generación de los dataframe para cada archivo, llevando a cabo el paso del procesamiento de los datos que comprenden el sistema y su adaptación a un formato que reconoce la plataforma de programación del algoritmo, que para el caso particular es lenguaje de programación Python.

En el preprocesamiento de los datos, es necesario organizar alguna información relacionada de las calificaciones de un ítem en particular, el curso Electivo SISSU, cuya calificación obtenida de la base de datos de HistoriaAcademica es de carácter cualitativo, mientras que en el sistema se requieren valoraciones de tipo cuantitativo. Razón

por la cual, en los dos modelos propuestos se relaciona el condicional que convierte el ítem de `Nota_Final` con resultado Aprobado en un valor equivalente a 5.0. Adicional a este proceso, en esta etapa también se tienen en cuenta únicamente aquellos cursos que sean aprobados, excluyendo del dataframe los cursos que no vayan a generar información importante y relevante al sistema, por esta razón el archivo inicial del dataframe de `Usuarios/Ítems`, que tenía un tamaño equivalente a 112bytes, queda convertido en un archivo con un tamaño correspondiente a 45bytes.

Se proponen dos modelos diferentes de algoritmos que, aunque aplican la técnica de recomendación por filtrado colaborativo, utilizan métodos diferentes para la creación de las matrices de entrenamiento y prueba, utilizando funciones y características diferentes. Para cada modelo es necesario la selección de hiper parámetros específicos que garantizaban el correcto funcionamiento de los métodos seleccionados y la obtención del mejor modelo que daba respuestas precisas como resultado de los desarrollos propuestos. Este proceso de selección de los mejores parámetros para los modelos propuestos se ejecutan de forma paralela pero requiere de condiciones de máquina específicas, presentando un alto consumo de recursos de máquina y el tiempo que se requiere para su ejecución es considerablemente alto y depende del tamaño de la Matriz de `Usuarios/Ítems` con que cuente el algoritmo.

El entrenamiento del algoritmo se considera un proceso relativamente rápido, en comparación del proceso de selección de los mejores parámetros de cada modelo. Para la evaluación de cada modelo propuesto se utiliza la evaluación por medio de métricas, que aunque son procesos que requieren de varios desarrollos y cálculos matemáticos, las librerías propuestas en cada modelo tienen funciones específicas que se encargan de ejecutar dichos desarrollos, optimizando tanto el tiempo de ejecución como la precisión de los resultados obtenidos.

Partiendo de los datos obtenidos en la ejecución de cada modelo, se tiene que el algoritmo de factorización matricial SVD, o segundo modelo, evidencia generar un mejor resultado en comparación con los resultados obtenidos en el otro escenario de prue-

ba de los modelos propuestos. Evidenciando así, que tal como lo indica la teoría, los algoritmos de sistemas de recomendación que aplican factorización matricial SVD tienen un rendimiento superior sobre algoritmos basados en memoria para filtro colaborativo (Cremonesi et al., 2010), complementando con los resultados obtenidos del algoritmo **Baseline only** que generan un mejor resultado en cuanto a la precisión de la predicción generada.

Para la selección del mejor modelo se tiene en cuenta una comparación de los resultados obtenidos de cada uno de forma independiente evaluando con las métricas de precisión de la predicción, la capacidad de generar la recomendación y la optimización de tiempos y recursos de máquina para el procesamiento del algoritmo al momento de generar la recomendación.

#### *Métricas de precisión de la predicción*

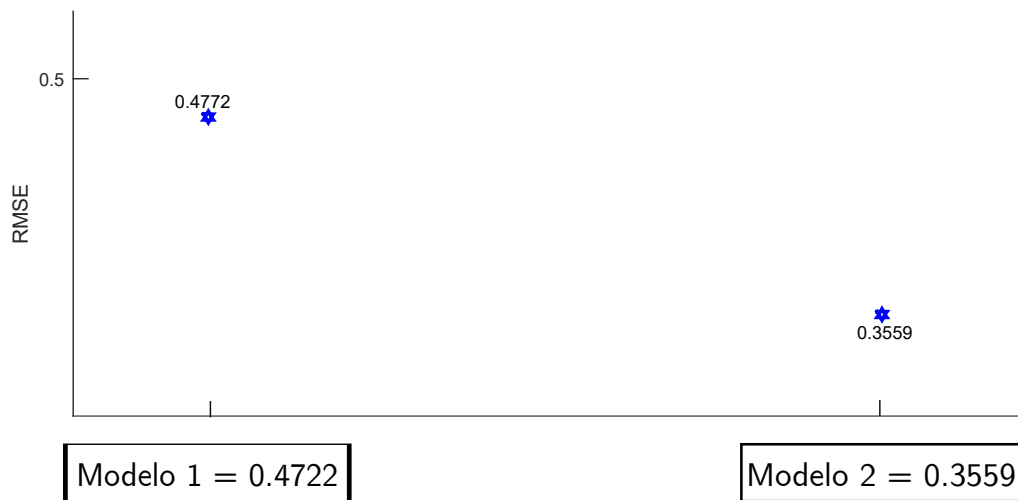
El primer método utilizado para la comparación de los modelos para seleccionar el mejor de ellos, es calculando la distancia existente entre la predicción realizada y la predicción, lo cual permite determinar la precisión en la predicción de cada modelo y representa la desviación estándar de las diferencias entre los valores estimados y los valores reales (Koren, 2010). Para esto, se utiliza el error cuadrático medio que corresponde a la media del error que el sistema comete al realizar las predicciones de las valoraciones de preferencia, al cuadrado, evidenciando así que esta medida penaliza errores de predicción aún mayores, por tanto, se concluye que, cuanto menor es el dicho error, mejor es la eficacia del sistema propuesto con sus respectivas características e hiper parámetros.

Como se evidencia en la Figura 25 se tiene, de acuerdo con los resultados obtenidos que, el modelo 2 que aplica factorización matricial con SVD es el método que mejores resultados entrega respecto a esta métrica analizada, haciendo énfasis que este modelo trabaja con la configuración de hiper parámetros que mejor resultado arroja en la selección de los mismos por algoritmo.

Figura 25

Comparación de datos obtenidos de RMSE para los tres modelos

Comparación de precisión en los dos modelos propuestos



*Nota.* Comparación de datos obtenidos de RMSE para los dos modelos. Se relaciona la representación gráfica de la comparación de los datos obtenidos en RMSE de los tres modelos propuestos resaltando que el modelo de menor valor corresponde al modelo 2.

Se evidencia también que, aunque los datos obtenidos en el primer modelo no distan mucho de la mejor precisión en la predicción, son valores que generan diferencial para la recomendación, teniendo en cuenta que en estos modelos es necesario realizar ajustes respecto a los datos de usuarios que no tenían ítems evaluados y su calificación era vacía inicialmente, lo cual ratifica que una de las ventajas que tiene la recomendación por filtrado colaborativo aplicando factorización matricial por SVD respecto de las otras metodologías a aplicar en esta técnica de recomendación se fundamenta en que no es necesario determinar una característica de vecinos óptimos respecto a todos los ítems evaluados., evitando así dispersión entre la matriz Usuarios/Ítems vs la matriz de predicción del sistema.

También es importante resaltar en este caso que, a medida que aumenta el número de usuarios y de ítems calificados, el valor del RMSE tiende a disminuir, dado que se tiene mayor número de datos para la selección de los hiper parámetros que determinan las características principales de las funciones específicas que realizan la validación

de los datos, las matrices de entrenamiento y prueba, y finalmente para realizar la verificación del funcionamiento del algoritmo en todos los datos que comprenden el dataframe del sistema.

### *Capacidad de generar la recomendación*

En los sistemas de recomendación se reconoce al coverage como la capacidad del sistema y su algoritmo de generar una recomendación válida y corresponde al porcentaje de elementos para los cuales el sistema y su algoritmo puede generar una predicción (Wei et al., 2017).

Al momento de evaluar la capacidad de recomendación de los modelos propuestos, se tiene que el modelo que implementa SVD es capaz de predecir los valores de calificación de los usuarios de acuerdo a los ítems que comprenden la matriz, partiendo de la información que se tiene y las funciones especiales que se usan para esta metodología. De igual manera es importante resaltar que en esta metodología no es necesario contar con un número amplio de datos de usuarios para determinar las posibles predicciones para cada ítem .

Respecto otro modelo se tiene que es necesario realizar un procesamiento de los datos que conforman el dataframe del sistema en la matriz Usuarios/Ítems utilizando únicamente los usuarios que cuentan con al menos 10 ítems calificados, con el fin de disminuir la dispersión en la recomendación por la gran cantidad de ítems que se tienen sin calificación. Adicional, aunque se optimiza el sistema utilizando una librería con funciones específica, también requiere de información completa o con poca carencia de ítems calificados, evidenciando así que no cumple con la totalidad de la capacidad de generar una recomendación que sea válida, teniendo en cuenta el comportamiento de los k-vecinos que conforman la base de datos.

Es importante resaltar que se solicita a un docente de Ingeniería Electrónica para que de forma manual, como se realiza actualmente, evalúe el historial académico de



Tabla 20

*Información de la base de datos del sistema*

Identificador	Cursos sugeridos por docente	Cursos recomen-	Cursos recomen-
		dados por SRMat V1 Modelo 1	dados por SRMat V1 Modelo 2
Estudiante 63	203055 - 203052 - 203053	203045 - 208006 - 208007	203052 - 208006 - 203055
Estudiante 33	700004 - 2150520 - 2150521	700004 - 208006 - 203052	700004 - 2150520 - 2150521
Estudiante 159	203054 - 203052 - 203049	203045 - 208006 - 203054	208006 - 203054 - 203049

*Nota.* Se presentan la relación de cursos electivos recomendados por los dos algoritmos diseñados y la recomendación que realiza un docente adscrito al programa de Ingeniería Electrónica para 3 estudiantes que conforman la base de datos.

tres (3) estudiantes que conforman la base de datos del sistema y sugiera la recomendación de tres (3) cursos electivos para cada uno de ellos, esto con el fin de obtener una valoración comparativa entre la sugerencia de un docente y la recomendación que genera el sistema. En la Tabla 20 se relaciona la información obtenida para los estudiantes en mención.

Como se evidencia en la Tabla 20, la predicción dada por el sistema tiene un alto porcentaje de precisión en la recomendación de los tres estudiantes relacionados en comparación de la sugerencia que le daría un docente que acompañe y/o asesore al estudiante en el proceso de selección de cursos para su respectiva matrícula atendiendo al registro académico del estudiante.

En la segunda columna se relacionan los cursos recomendados por el docente, en la tercera columna los que sugiere el primer modelo propuesto y en la última columna se relacionan los cursos recomendados por el segundo modelo propuesto. Como se observa, es este último modelo el que genera una predicción similar a la recomendación

dada por el docente, evidenciando que el sistema trabaja sobre predicciones de acuerdo con el historial académico del estudiante en relación con los demás estudiantes que componen la base de datos del sistema.

### *Optimización de tiempos y recursos de máquina*

El proceso de verificación y comparación de los tiempos de ejecución de cada algoritmo propuesto y el uso de recursos de máquina se tiene en cuenta con el comportamiento del sistema en la ejecución de cada modelo.

Es importante resaltar que en el caso de SVD, si se presenta una parte de la programación que se encarga de encontrar los mejores hiper parámetros que se requieren para la configuración de las funciones específicas del algoritmo, evidenciando un alto consumo de recursos de máquina y de un tiempo considerable para su ejecución. Pero una vez se tienen los datos de los hiper parámetros, no es necesario ejecutar este apartado del algoritmo, evidenciando una optimización del sistema en cuanto a consumo de recursos de máquina del equipo y tiempo de ejecución del algoritmo.

Como conclusión relacionada con la optimización de tiempos y recursos de máquina, se tiene que el algoritmo que mejor responde antes estas características y del cual se obtienen a su vez las mejores recomendaciones hace referencia al algoritmo que implementa la factorización matricial SVD, una vez se tiene a mejor optimización del sistema con los hiper parámetros que fueron determinados como de mejor desempeño.

### **Discusión final**

Teniendo en cuenta los resultados obtenidos en los desarrollos propuestos y los análisis presentados, a continuación se relaciona información que representa el análisis final de los algoritmos y la selección de la propuesta que mejor se adapta a las necesidades del entorno:

- Respecto a las bases de datos se resalta que, aunque para el caso se trabaja con datos correspondientes a 238 estudiantes, información consignada en la primera base de datos denominada Usuarios y el historial académico de cada uno que comprende la segunda base de datos utilizada en el sistema, toda esta información se encuentra consignada en un archivo de excel independiente. En el mismo formato se tiene la tercera base de datos que relaciona la información de los cursos que corresponden a los dos programas académicos relacionados para el sistema de recomendación de acuerdo con la información establecida en las respectivas mallas curriculares de plan nuevo de dichos programas, teniendo en cuenta también lo consignado en los acuerdos de equivalencia actualizados en julio de 2020.
- Esta información es utilizada en el algoritmo para crear dataframes de las tres bases de datos, comprobando de esta manera que la representación de datos en forma matricial con aplicación de dispersión por medio de la programación con Python garantizan una reducción en el uso de memoria en un 42.8 % y a la vez se logra una reducción del 20 % en el tiempo de entrenamiento.
- Como condición particular para este algoritmo, es necesario realizar preprocesamiento de los datos, actividad que permite la optimización de la base de datos del historial académico, evidenciado de esta manera la flexibilidad que tiene el algoritmo de organizar la información que se tiene en los dataframes llevando a mejorar la respuesta del sistema y del equipo en donde se ejecuta la lógica de programación propuesta.
- Trabajar con dos propuestas permite la evaluación de los resultados obtenidos con los datos que arrojan las métricas de cada uno. Esto a su vez garantiza que aplicando diferentes métodos se pueda obtener la mejor combinación de parámetros de cada modelo, evaluando resultados por medio del RMSE, teniendo en cuenta que se busca el modelo que garantice el menor resultado. Esto permite que el sistema genere una mejor respuesta en cuanto se refiere al entrenamiento de los modelos, y una vez se cuente con dichos valores de hiper parámetros con mejor respuesta, llevar al sistema a que optimice recursos de máquina y tiempos de respuesta.

- *Evaluando la precisión obtenida en cada modelo con respecto al tamaño de la base de datos de entrada para los algoritmos, se evidencia que el modelo 2 que aplica factorización matricial SVD entrega como resultado de evaluación de métricas valores más bajos en comparación con el otro escenario, lo que indica un mejor rendimiento de este algoritmo. De igual forma se comprueba que el tamaño de la base de datos de entrada no afecta drásticamente la precisión de los SR aquí utilizados para el caso de la evaluación por RMSE.*
- *En términos de optimización de recursos de máquina y evaluación de tiempos de respuesta se tiene que, una vez obtenidos los mejores valores de respuesta para los hiper parámetros de configuración en cada modelo, la mejor respuesta también se tiene con el modelo que aplica factorización matricial SVD se obtienen un tiempo de respuesta mejor, consiguiendo además mejores resultados en la precisión de las predicciones generadas.*

## Conclusiones

El desarrollo de este sistema de recomendación surge de la necesidad de liberar a los docentes de la Cadena de formación ETR de la UNAD en términos del apoyo y acompañamiento constante a los estudiantes de planes de nuevos de los programas de Ingeniería Electrónica e Ingeniería de Telecomunicaciones al momento de realizar la selección de cursos electivos en el proceso de matrícula en cada periodo académico, utilizando técnicas de inteligencia artificial que optimizaran el acompañamiento al estudiante. Por tanto, con este desarrollo se llega al diseño de una herramienta que permite generar una recomendación al usuario para la selección de cursos electivos y de esta manera, continuar con su avance en el curso de la malla curricular del programa.

Teniendo en cuenta lo que establece la inteligencia de negocios respecto a los procesos ETL, se evidencia que este desarrollo aplica el proceso de extracción de datos y a su vez, aplicando la transformación de esos datos según la necesidad del sistema, garantizando así el análisis de la misma, aplicando acciones de procesamiento y optimización de la información. Es importante resaltar que el proceso de ETL que se aplica en este caso, genera resultados como una relación de sugerencias para los usuarios, evidenciado de esta manera listados específicos para cada uno de ellos, siendo información de gran ayuda en busca de aportar opciones específicas al estudiante para la toma de decisiones y acciones académicas enfocadas en su avance en la malla curricular.

En lo concerniente a los Sistemas de Recomendación, el documento relaciona un estudio de literatura de las características generales de estos sistemas y de las diferentes técnicas que existen o se usan actualmente. De esta manera, se sustenta la selección de la técnica a utilizar en el desarrollo propuesto que corresponde al Filtrado Colabo-

rativo, resaltando ventajas y desventajas de esta técnica y relacionando las diferentes metodologías que se pueden aplicar para el diseño de los algoritmos a cargo de estas funciones particulares.

El objetivo principal de este trabajo está centrado en el desarrollo de un sistema de recomendación de cursos electivos a estudiantes de Ingeniería Electrónica e Ingeniería de Telecomunicaciones de la UNAD, generando sugerencias a los usuarios o grupo de usuarios, teniendo en cuenta la forma de abordar el reto de los sistemas de recomendación en términos de optimización de tiempo y recursos de máquina de acuerdo con el manejo de la información que forma parte de la base de datos del sistema y el problema del cold-start o arranque en frío, principal inconveniente de los sistemas de recomendación.

Para el alcance de los objetivos propuestos se tiene como punto de partida obtención de la información relacionada en las bases de datos utilizadas, con su posterior transformación, preprocesamiento y procesamiento, teniendo en cuenta las características requeridas por la herramienta de programación seleccionada y su respectivo lenguaje, lo cual corresponde al uso de Jupyter Notebook y Python respectivamente. Como se evidencia en la información relacionada en el documento, esta información se utiliza en los dos modelos propuestos y se tienen diferentes estrategias, metodologías y/o técnicas de uso de la información para la generación del algoritmo entrega la recomendación al usuario.

En este sentido, este documento presenta el desarrollo de dos modelos con el uso conjunto de dos enfoques importantes para los sistemas de recomendación, de esta forma se tienen en primera instancia las predicciones realizadas a través del enfoque de Factorización matricial no negativa que estima una métrica de similitud entre los ítems relacionados en la base de datos, a partir de la cual se obtienen los usuarios que conforman el vecindario, es decir, los usuarios más cercanos. El segundo enfoque utilizado aplica Factorización Matricial SVD utilizando las calificaciones obtenidas por los usuarios en cada ítem de la base de datos y de esta forma entrenar un modelo que calcula

las variables latentes con el fin de generar la predicción de los ítems que carecen de información. Los dos enfoques son evaluados por medio del uso de las métricas utilizadas para los sistemas de recomendación y teniendo en cuenta la optimización de tiempo y recursos de máquina en la ejecución de cada uno.

La evaluación de los modelos propuestos se sustenta en la hipótesis que se tiene de validez de las recomendaciones obtenidas en los sistemas de recomendación por filtrado colaborativo que establece que, la eficiencia de los algoritmos para estos sistemas se fundamenta en la información relacionada en la matriz de de Usuarios/Ítems y la función de similitud que se utiliza para la creación del vecindario conformado por los vecinos más cercanos. Para lo cual es importante evaluar, desde la información que se tiene en la base de datos, si el sistema tendrá enfoque colaborativo basado en ítems o basado en contenido, evidenciando que el enfoque utilizado para este desarrollo es basado en ítems, teniendo presente que se cuenta con la matriz de ratings o calificaciones correspondiente.

Uno de los objetivos principales de los sistemas de recomendación es lograr que el algoritmo propuesto genere una buena recomendación en un tiempo relativamente pequeño y con el desarrollo del segundo modelo se evidencia que es posible lograr una optimización del tiempo cuando se aplica factorización matricial SVD.

En el procedimiento de validación de resultados, utilizando el valor obtenido con el RMSE de cada modelo, se evidencia que los rangos obtenidos se encuentran dentro de los valores permisibles como indica la teoría, que es un valor inferior a 0.5. Adicionalmente, al momento de comprobar los datos obtenidos con los dos algoritmos diseñados con respecto a la recomendación realizada por un docente de la Cadena de formación ETR, se evidencia que en el primer modelo propuesto se tiene un porcentaje de similitud aproximadamente del 33% y con el segundo modelo propuesto, este porcentaje aproximado de similitud oscila alrededor del 80%. Con estos datos es posible concluir que el sistema de recomendación propuesto con factorización matricial con implementación de SVD genera mejores resultados tanto en el valor correspondiente al RMSE, el

porcentaje de similitud en la recomendación y el tiempo de ejecución del algoritmo.

Finalmente, se evidencia de manera práctica que, es posible generar por medio de la aplicación de técnicas de inteligencia artificial, una herramienta que entregue al usuario una serie de sugerencias y/o recomendaciones sobre los cursos electivos a seleccionar, con el fin de dar continuidad con su proceso académico adelantado en UNAD, teniendo en cuenta las calificaciones obtenidas en los diferentes cursos matriculados, tomando como base el historial académico de un grupo de estudiantes con características similares y que a su vez, aporte a disminuir los trámites de novedades ante registro y control por selección errónea de cursos electivos.



## Trabajos futuros

En el desarrollo del sistema de recomendación propuesto, se evidenciaron posibles líneas futuras de trabajo a tenerse en cuenta para la mejora del mismo y que pueden formar parte de aplicaciones en proyectos futuros.

En primera medida, el sistema propuesto se puede optimizar o mejorar garantizando una base de datos con mayor cantidad de información, específicamente de usuarios, dado que se está utilizando información relacionada con estudiantes matriculados en un centro de la UNAD y estudiantes de algunos cursos electivos y que se encuentran en un alto porcentaje de avance en la malla curricular, superando el 80 % de la malla. Esta mejora permite que el universo o vecindario creado genere mejores resultados en el momento de la evaluación del sistema por medio de métricas, garantizando así un menor error en la predicción.

Analizando el proceso de inicialización de las matrices que se requiere para aplicar la técnica de factorización es importante tener claro que el proceso para encontrar los valores que garanticen la obtención del mejor modelo es bueno solucionar la selección del número de componentes latentes que lleven a una disminución en la dimensionalidad y así obtener mejores resultados en la predicción. Por lo tanto, se recomienda encontrar un método que facilite la elección del número de componentes latentes sin tener que recurrir a pruebas, teniendo en cuenta que la base de datos puede ir en aumento.

Los dos algoritmos desarrollados corresponden a un sistema de recomendación de matrícula centrado en cursos de tipo Electivos, con el fin de complementar el sistema es conveniente implementar en el algoritmo los comandos que le faciliten la reco-

mendación de cursos Obligatorios al estudiante teniendo en cuenta su avance en la malla curricular del programa de formación matriculado en la universidad y, de esta manera, recomendar al estudiante la matrícula de acuerdo con el desarrollo de su programa de formación, sus calificaciones y posibles líneas de trabajo.

El sistema propuesto, en sus dos modelos, implementa la técnica de filtrado colaborativo, lo cual hace que el sistema no genere una predicción totalmente cercana a una recomendación analizando diferentes puntos de vista, por lo cual es importante combinar este algoritmo con recomendación basada en contenido, analizando las características puntuales de los diferentes ítems que conforman la base de datos del sistema, esto con el fin de ampliar las opciones de recomendación relacionando temáticas que sean de mayor afinidad para el usuario, logrando de esta manera la obtención de un algoritmo de un sistema de recomendación híbrido, que como lo indica la teoría expuesta, son los mejores algoritmos para los sistemas de recomendación.

## Bibliografía

- Ahn, H. J. (2008). A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem. *Information Sciences*, 178(1):37 – 51.
- Albatayneh, N. A., Ghauth, K. I., and Chua, F.-F. (2018). Utilizing learners' negative ratings in semantic content-based recommender system for e-learning forum. *Educational Technology and Society*, 21(1):112–125.
- Angée, S., Lozano, S., Montoya-Munera, E., Ospina Arango, J., and Tabares, M. (2018). *Towards an Improved ASUM-DM Process Methodology for Cross-Disciplinary Multi-organization Big Data & Analytics Projects: 13th International Conference, KMO 2018, Žilina, Slovakia, August 6–10, 2018, Proceedings*, pages 613–624.
- Ansari, M. H., Moradi, M., Nikrah, O., and Kambakhsh, K. M. (2017). CodERS: A hybrid recommender system for an E-learning system. In H., T., editor, *Proceedings - 2016 2nd International Conference of Signal Processing and Intelligent Systems, ICSPIS 2016*. Institute of Electrical and Electronics Engineers Inc.
- Antony Rosewelt, L. and Arokia Renjit, J. (2020). A content recommendation system for effective e-learning using embedded feature selection and fuzzy DT based CNN. *Journal of Intelligent and Fuzzy Systems*, 39(1):795–808.
- Bertani, R. M., Bianchi, R. A. C., and Costa, A. H. R. (2020). Combining novelty and popularity on personalised recommendations via user profile learning. *Expert Systems with Applications*, 146:113149.
- Blanco-Fernandez, Y., Pazos-Arias, J. J., Gil-Solla, A., Ramos-Cabrera, M., Lopez-Nores, M., Garcia-Duque, J., Fernandez-Vilas, A., Diaz-Redondo, R. P., and Bermejo-Munoz, J. (2008). A flexible semantic inference methodology to reason

- about user preferences in knowledge-based recommender systems. *Knowledge-based systems*, 21(4):305–320.
- Blanco-Fernandez, Y., Pazos-Arias, J. J., Lopez-Nores, M., Gil-Solla, A., and Ramos-Cabrer, M. (2006). Avatar: an improved solution for personalized tv based on semantic interface. In *2006 Digest of Technical Papers International Conference on Consumer Electronics*, pages 145–146.
- Blanco-Fernández, Y., López-Nores, M., Gil-Solla, A., Ramos-Cabrer, M., and Pazos-Arias, J. J. (2011). Exploring synergies between content-based filtering and spreading activation techniques in knowledge-based recommender systems. *Information Sciences*, 181(21):4823 – 4846.
- Bobadilla, J., Serradilla, F., and Hernando, A. (2009). Collaborative filtering adapted to recommender systems of e-learning. *Knowledge-Based Systems*, 22(4):261–265.
- Bridge, D. and Ferguson, A. (2002). An expressive query language for product recommender systems. *Artif. Intell. Rev.*, 18:269–307.
- Burke, R. (2002). Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, 12.
- Burke, R. (2007). Hybrid web recommender systems. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 4321 LNCS:377–408.
- Burke, R. (2017). Knowledge-based recommender systems. *Journal of the American Association of Nurse Practitioners*, 29(9):514–520.
- Cai, W., Pan, W., Liu, J., Chen, Z., and Ming, Z. (2020). k-reciprocal nearest neighbors algorithm for one-class collaborative filtering. *Neurocomputing*, 381:207 – 216.
- Cardenas, M., Abadía, C., Vela, P., and Montero, R. (2017). Lineamientos generales del currículo en la unad.
- Caro Piñeres, M. F., Hernández, J., and Jiménez Builes, J. A. (2011). Diseño de un sistema de recomendación en repositorios de objetos de aprendizaje basado en

- la percepción del usuario: Caso rodas. *Ciencia e Ingeniería Neogranadina*, 21(1):51.
- Castillo, É. Y., García, N. N., Corredor, C. M., and Malaver, C. E. C. (2017). Evolución de la educación superior a distancia: desafíos y oportunidades para su gestión. *Revista Virtual Universidad Católica del Norte*.
- Chakraborty, P. S. (2009). A scalable collaborative filtering based recommender system using incremental clustering. In *2009 IEEE International Advance Computing Conference*, pages 1526–1529.
- Chen, G., Wang, F., and Zhang, C. (2009). Collaborative filtering using orthogonal nonnegative matrix tri-factorization. *Information Processing & Management*, 45(3):368 – 379.
- Christakou, C., Vrettos, S., and Stafylopatis, A. (2007). A hybrid movie recommender system based on neural networks. *International Journal on Artificial Intelligence Tools*, 16:771–792.
- Collado Sánchez, A. (2014). Sistema de recomendación de recursos basado en filtrado colaborativo para la plataforma edX. *2016*, page 171.
- Cremonesi, P., Koren, Y., and Turrin, R. (2010). Performance of recommender algorithms on top-n recommendation tasks. pages 39–46.
- de Campos, L. M., Fernández-Luna, J. M., Huete, J. F., and Rueda-Morales, M. A. (2010). Combining content-based and collaborative recommendations: A hybrid approach based on bayesian networks. *International Journal of Approximate Reasoning*, 51(7):785 – 799.
- De Nart, D. and Tasso, C. (2014). A personalized concept-driven recommender system for scientific libraries. *Procedia Computer Science*, 38:84 – 91. 10th Italian Research Conference on Digital Libraries, IRCDL 2014.
- Demir, G. N., Uyar, A. S., and Ögüdücü, S. G. (2007). Graph-based sequence clustering through multiobjective evolutionary algorithms for web recommender systems. In *Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation*, GECCO '07, page 1943–1950, New York, NY, USA. Association

for Computing Machinery.

- Dierk, S. F. (1972). The smart retrieval system: Experiments in automatic document processing. *IEEE Transactions on Professional Communication*, PC-15(1):17–17.
- Ekstrand, M. D. (2018). Lenskit for python: Next-generation software for recommender system experiments.
- Ekstrand, M. D. (2020). Lenskit for python: Next-generation software for recommender systems experiments. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management*.
- Ekstrand, M. D., Ludwig, M., Kolb, J., and Riedl, J. T. (2011). Lenskit: A modular recommender framework. In *Proceedings of the Fifth ACM Conference on Recommender Systems*, RecSys '11, page 349–350, New York, NY, USA. Association for Computing Machinery.
- Faulconbridge, R. I. and Ryan, M. J. M. J. (2005). *Engineering a system : managing complex technical projects / R. Ian Faulconbridge, Michael J. Ryan*. Argos Press, Canberra, A. C.T.
- Felfernig, A., Friedrich, G., Jannach, D., and Zanker, M. (2006). An integrated environment for the development of knowledge-based recommender applications. *International Journal of Electronic Commerce*.
- Funk, S. (2006). Netflix update: Try this at home.
- Gantner, Z., Rendle, S., Freudenthaler, C., and Schmidt-Thieme, L. (2011). MyMediaLite: A free recommender system library. In *Proceedings of the 5th ACM Conference on Recommender Systems (RecSys 2011)*.
- Goldberg, D., Nichols, D., Oki, B. M., and Terry, D. (1992). Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 35:61–70.
- Hdioud, F., Frikh, B., and Ouhbi, B. (2012). A comparison study of some algorithms in Recommender Systems. In *2012 Colloquium in Information Science and Technology*, pages 130–135.
- Hernández Sampieri, R. and Fernández Collado, C. (2014). *Metodología de la investiga-*

- ción* /. McGraw-Hill Interamericana, México, D. F. :, 6a ed. edition.
- Hug, N. (2020a). Surprise: A python library for recommender systems. *Journal of Open Source Software*, 5(52):2174.
- Hug, N. (2020b). Surprise: A python library for recommender systems. *Journal of Open Source Software*, 5:2174.
- Jannach, D., Zanker, M., Felfernig, A., and Friedrich, G. (2010). Recommender systems: An introduction. *Recommender Systems: An Introduction*, pages 1–335.
- Jensen, F. and Nielsen, T. (2001). *Bayesian Networks and Decision Graphs*. Springer.
- Jones, E., Oliphant, T., Peterson, P., et al. (2001). SciPy: Open source scientific tools for Python.
- Khan, A. I. and Al-Badi, A. (2020). Open source machine learning frameworks for industrial internet of things. *Procedia Computer Science*, 170:571 – 577. The 11th International Conference on Ambient Systems, Networks and Technologies (ANT) / The 3rd International Conference on Emerging Data and Industry 4.0 (EDI40) / Affiliated Workshops.
- Kiyoki, Y., Kato, K., Yamaguchi, N., and Masuda, T. (1988). *Database Machines and Knowledge Base Machines*, chapter A Stream-Oriented Approach to Parallel Processing for Deductive Databases, pages 475–488. Springer.
- Kluyver, T., Ragan-Kelley, B., Pérez, F., Granger, B., Bussonnier, M., Frederic, J., Kelley, K., Hamrick, J., Grout, J., Corlay, S., Ivanov, P., Avila, D., Abdalla, S., and Willing, C. (2016). Jupyter notebooks – a publishing format for reproducible computational workflows. In Loizides, F. and Schmidt, B., editors, *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, pages 87 – 90. IOS Press.
- Koren, Y. (2010). Factor in the neighbors: Scalable and accurate collaborative filtering. *TKDD*, 4.
- Koren, Y., Bell, R., and Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, 42:30–37.
- Kotkov, D., Wang, S., and Veijalainen, J. (2016). A survey of serendipity in recommen-

- der systems. *Knowledge-Based Systems*, 111:180 – 192.
- Krzywicki, A., Wobcke, W., Kim, Y. S., Cai, X., Bain, M., Mahidadia, A., and Compton, P. (2015). Collaborative Filtering for people-to-people recommendation in online dating: Data analysis and user trial. *International Journal of Human-Computer Studies*, 76:50–66.
- Lopez-Nores, M., Blanco-Fernandez, Y., Pazos-Arias, J. J., and Gil-Solla, A. (2012). Property-based collaborative filtering for health-aware recommender systems. *EXPERT SYSTEMS WITH APPLICATIONS*, 39(8):7451–7457.
- Lorenzi, F. and Ricci, F. (2003). Case-based recommender systems: A unifying view. In *Intelligent Techniques for Web Personalization*, volume 3169.
- Loukis, E., Janssen, M., and Mintchev, I. (2019). Determinants of software-as-a-service benefits and impact on firm performance. *Decision Support Systems*, 117:38 – 47.
- Lu, J., Wu, D., Mao, M., Wang, W., and Zhang, G. (2015). Recommender system application developments: A survey. *Decision Support Systems*, 74:12–32.
- Marlin, B. (2004). Collaborative filtering: A machine learning perspective. *Master Thesis*, page 137.
- Martinez, L., Barranco, M., Perez, L., and Espinilla, M. (2008). A knowledge based recommender system with multigranular linguistic information. *International Journal of Computational Intelligence Systems*, 1:225–236.
- Miller, B. N., Konstan, J. A., and Riedl, J. (2004). PocketLens: Toward a Personal Recommender System. *ACM Transactions on Information Systems*, 22(3):437–476.
- Muthukrishnan, N., Maleki, F., Ovens, K., Reinhold, C., Forghani, B., and Forghani, R. (2020). Brief history of artificial intelligence. *Neuroimaging Clinics of North America*, 30(4):393 – 399. Machine Learning and Other Artificial Intelligence Applications.
- Muñoz, H. (2014). Sistema de recomendación multimedia basado en perfiles de usuario. Technical report, Univerisdad Politecnica de Cataluña.
- Nicholas, J. and Francis, S. (2019). A comprehensive survey of neighborhood-based re-



- commendation methods used in e-learning recommender systems. *International Journal of Computer Sciences and Engineering*, 7:443–450.
- Pazzani, M. J. (1999). A framework for collaborative, content-based and demographic filtering. *Artificial intelligence review*, 13:393–408.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.
- Perconti, P. and Plebe, A. (2020). Deep learning and cognitive science. *Cognition*, 203:104365.
- Reback, J., McKinney, W., jbrockmendel, den Bossche, J. V., Augspurger, T., Cloud, P., gyoung, Sinhrks, Klein, A., Roeschke, M., Tratner, J., She, C., Ayd, W., Hawkins, S., Petersen, T., Schendel, J., Hayden, A., Garcia, M., Jancauskas, V., MomIsBestFriend, Battiston, P., Seabold, S., chris b1, h vetinari, Hoyer, S., Overmeire, W., alimcmaster1, Mehyar, M., Whelan, C., and Kluyver, T. (2020). pandas-dev/pandas: Pandas.
- Rich, E. (1979). User modeling via stereotypes. *Cogn. Sci.*, 3:329–354.
- Rodríguez, P. A. (2017). *Modelo Genérico para la Recomendación Híbrida y Adaptativa de Recursos Educativos Digitales*. PhD thesis, Universidad Nacional, sede Medellín.
- Rodríguez, R., Espinilla, M., Sánchez, P., and Martínez, L. (2010). Using linguistic incomplete preference relations to cold start recommendations. *Internet Research*, 20:296–315.
- Rodríguez Marín, P., Duque, N., and Ovalle, D. (2016). Método híbrido de recomendación adaptativa de objetos de aprendizaje basado en perfiles de usuario. *Formación universitaria*, 9:83–94.
- Ruiz-Iniesta, A., Jiménez-Díaz, G., and Gómez-Albarrán, M. (2010). Personalización en recomendadores basados en contenido y su aplicación a repositorios de objetos

- de aprendizaje. *Revista Iberoamericana de Tecnologías del Aprendizaje*, 5(1):31–38.
- Salakhutdinov, R. and Mnih, A. (2008). Probabilistic matrix factorization. In *Advances in Neural Information Processing Systems*, volume 20.
- Salehi, M. (2013). Application of implicit and explicit attribute based collaborative filtering and BIDE for learning resource recommendation. *Data & Knowledge Engineering*, 87:130–145.
- Salehi, M., Kmalabadi, I. N., and Ghoushchi, M. B. G. (2012). A New Recommendation Approach Based on Implicit Attributes of Learning Material. *IERI Proceedings*, 2:571–576.
- Shardanand, U. and Maes, P. (2015). Social Information Filtering: Algorithms for Automating "Word of Mouth. *ENR (Engineering News-Record)*, 274(27):1–15.
- Symeonidis, P., Nanopoulos, A., and Manolopoulos, Y. (2007). Feature-weighted user model for recommender systems. In *User Modeling 2007*, pages 97–106.
- Tarazona, M. B. G., Chavez, L. S., and Ferro, E. R. (2013). Modelacion de sistemas de recomendacion aplicando redes neuronales artificiales. *Vision electronica*, 2:45–56.
- Tarus, J. K., Niu, Z., and Mustafa, G. (2018). Knowledge-based recommendation: a review of ontology-based recommender systems for e-learning. *Artificial Intelligence Review*, 50(1):21–48.
- Taslaman, L. and Nilsson, B. (2012). A framework for regularized non-negative matrix factorization, with application to the analysis of gene expression data. *PloS one*, 7:e46331.
- Tekin, C. and Van Der Schaar, M. (2015). RELEAF: An algorithm for learning and exploiting relevance. *IEEE Journal on Selected Topics in Signal Processing*, 9(4):716–727.
- Tzung-Pei Hong and Jyh-Bin Chen (1999). Finding relevant attributes and membership functions. *Fuzzy Sets and Systems*, 103(3):389 – 404.
- Universidad Nacional Abierta y a Distancia - UNAD (2020). Proceso: Ciclo de vida del

- estudiante - procedimiento relacionado: Inscripción y matrícula.
- Van Rossum, G. and Drake, F. L. (2009). *Python 3 Reference Manual*. CreateSpace, Scotts Valley, CA.
- Vera, J., Mamani, A. O., and Villalba, K. (2017). Modelo de sistema de recomendación de Objetos de Aprendizaje en dispositivos móviles, caso: Desarrollo del pensamiento computacional. *Revista Referencia Pedagógica*, 5(1):96–108.
- Viappiani, P., Pu, P., and Faltings, B. (2008). Preference-based search with adaptive recommendations. *Ai Communications*, 21:155–175.
- Villaverde, C., Bustos, A., and Salvador, C. (2010). Los entornos virtuales como espacios de enseñanza y aprendizaje. Una perspectiva psicoeducativa para su caracterización y análisis. *Revista mexicana de investigación educativa*, 15:163 – 184.
- Wang, K., Zhang, T., Xue, T., Lu, Y., and Na, S.-G. (2019). E-Commerce Personalized Recommendation Analysis by Deeply-learned Clustering. *Journal of Visual Communication and Image Representation*, page 102735.
- Wei, J., He, J., Chen, K., Zhou, Y., and Tang, Z. (2017). Collaborative filtering and deep learning based recommendation system for cold start items. *Expert Systems with Applications*, 69:29–39.
- Xue, G.-R., Lin, C., Yang, Q., Xi, W., Zeng, H.-J., Yu, Y., and Chen, Z. (2005). Scalable collaborative filtering using cluster-based smoothing. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '05*, page 114–121, New York, NY, USA. Association for Computing Machinery.
- Zanker, M., Jessenitschnig, M., and Schmid, W. (2010). Preference reasoning with soft constraints in constraint-based recommender systems. *Constraints*, 15:574–595.
- Zhang, Y., Li, J., Zakharov, Y. V., Li, J., Li, Y., Lin, C., and Li, X. (2019). Deep Learning Based Single Carrier Communications Over Time-Varying Underwater Acoustic Channel. *IEEE Access*, 7:38420–38430.
- Zhen, L., Huang, G. Q., and Jiang, Z. (2010). An inner-enterprise knowledge recommender system. *Expert Systems with Applications*, 37(2):1703 – 1712.

- Zhou, X., He, J., Huang, G., and Zhang, Y. (2015). Svd-based incremental approaches for recommender systems. *Journal of Computer and System Sciences*, 81(4):717–733.
- Zhou, Y., Wilkinson, D., Schreiber, R., and Pan, R. (2008). Large-scale parallel collaborative filtering for the netflix prize. In Fleischer, R. and Xu, J., editors, *Algorithmic Aspects in Information and Management*, pages 337–348, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Zhuang, F., Zheng, J., Chen, J., Zhang, X., Shi, C., and He, Q. (2018). Transfer collaborative filtering from multiple sources via consensus regularization. *Neural Networks*, 108:287–295.
- Zuhadar, L., Nasraoui, O., Wyatt, R., and Romero, E. (2009). Multi-model ontology-based hybrid recommender system in e-learning domain. In *Proceedings - 2009 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology - Workshops, WI-IAT Workshops 2009*, volume 3, pages 91–95, Milano.

## ANEXOS

- Anexo 1. Archivo Base de datos del sistema formato .xlsx: <https://cutt.ly/flefju4>
- Anexo 2. Manual de usuario software "SRMat - V1" formato .pdf: <https://cutt.ly/hk4Pvvh>