

**UNIVERSIDAD NACIONAL DE SAN ANTONIO ABAD DEL CUSCO**  
**FACULTAD DE INGENIERÍA ELÉCTRICA, ELECTRÓNICA, INFORMÁTICA Y**  
**MECÁNICA**  
**ESCUELA PROFESIONAL DE INGENIERÍA INFORMÁTICA Y DE SISTEMAS**



**TESIS**

**ESTUDIO COMPARATIVO DE MODELOS DE CLASIFICACIÓN BASADOS EN ELO,  
GLICKO-2 Y TRUESKILL EN LA DETECCIÓN DE ESTUDIANTES CON BAJO  
RENDIMIENTO EN EL PERÍODO POST-COVID DE LA UNIVERSIDAD NACIONAL  
DE SAN ANTONIO ABAD DEL CUSCO**

**PRESENTADO POR:**

Br. SEBASTIAN ISRAEL MACEDO GHEILER

**PARA OPTAR AL TÍTULO PROFESIONAL DE  
INGENIERO INFORMÁTICO Y DE SISTEMAS**

**ASESOR:**

Dra. NILA ZONIA ACURIO USCA

**CUSCO – PERÚ**

**2025**



# Universidad Nacional de San Antonio Abad del Cusco

## INFORME DE SIMILITUD

(Aprobado por Resolución Nro.CU-321-2025-UNSAAC)

El que suscribe, el Asesor Nila Zonia Acuña Usca, quien aplica el software de detección de similitud al trabajo de investigación/tesis titulada: "Estudios Comparativo de modelos de clasificación Basadas en ELO, GLICKO-2 y TRUESKILL en la detección de estudiantes con bajo rendimiento en el período post covid de la Universidad Nacional de San Antonio Abad del Cusco". Presentado por: Sebastián Israel Maredo Gheiler DNI N° 72796109; presentado por: ..... DNI N°: ..... Para optar el título Profesional/Grado Académico de Ingiero Informático y de Sistemas.

Informo que el trabajo de investigación ha sido sometido a revisión por 9 veces, mediante el Software de Similitud, conforme al Art. 6º del **Reglamento para Uso del Sistema Detección de Similitud en la UNSAAC** y de la evaluación de originalidad se tiene un porcentaje de 0%.

### Evaluación y acciones del reporte de coincidencia para trabajos de investigación conducentes a grado académico o título profesional, tesis

Porcentaje	Evaluación y Acciones	Marque con una (X)
Del 1 al 10%	No sobrepasa el porcentaje aceptado de similitud.	X
Del 11 al 30 %	Devolver al usuario para las subsanaciones.	
Mayor a 31%	El responsable de la revisión del documento emite un informe al inmediato jerárquico, conforme al reglamento, quien a su vez eleva el informe al Vicerrectorado de Investigación para que tome las acciones correspondientes; Sin perjuicio de las sanciones administrativas que correspondan de acuerdo a Ley.	

Por tanto, en mi condición de Asesor, firmo el presente informe en señal de conformidad y adjunto las primeras páginas del reporte del Sistema de Detección de Similitud.

Cusco, 18 de Diciembre de 2025.

Firma

Post firma. Nila Zonia Acuña Usca

Nro. de DNI 23821762

ORCID del Asesor 0000-0001-7706-780X

#### Se adjunta:

- Reporte generado por el Sistema Antiplagio.
- Enlace del Reporte Generado por el Sistema de Detección de Similitud: oid: 27259:540356859

# Tesis 2025 - SEBASTIAN ISRAEL MACEDO GHEILER.docx

 Universidad Nacional San Antonio Abad del Cusco

## Detalles del documento

Identificador de la entrega

trn:oid::27259:540356859

160 páginas

Fecha de entrega

13 dic 2025, 6:55 p.m. GMT-5

31.603 palabras

Fecha de descarga

13 dic 2025, 7:18 p.m. GMT-5

178.657 caracteres

Nombre del archivo

Tesis 2025 - SEBASTIAN ISRAEL MACEDO GHEILER.docx

Tamaño del archivo

4.5 MB

# 0% Similitud general

El total combinado de todas las coincidencias, incluidas las fuentes superpuestas, para ca...

## Filtrado desde el informe

- ▶ Bibliografía
- ▶ Texto citado
- ▶ Texto mencionado
- ▶ Coincidencias menores (menos de 20 palabras)
- ▶ Trabajos entregados

## Fuentes principales

- 0%  Fuentes de Internet  
0%  Publicaciones  
0%  Trabajos entregados (trabajos del estudiante)

## Marcas de integridad

### N.º de alerta de integridad para revisión

-  **Texto oculto**  
12 caracteres sospechosos en N.º de páginas

El texto es alterado para mezclarse con el fondo blanco del documento.

Los algoritmos de nuestro sistema analizan un documento en profundidad para buscar inconsistencias que permitirían distinguirlo de una entrega normal. Si advertimos algo extraño, lo marcamos como una alerta para que pueda revisarlo.

Una marca de alerta no es necesariamente un indicador de problemas. Sin embargo, recomendamos que preste atención y la revise.

## DEDICATORIA

*A mis queridos bisabuelos, Elva Duran de Sánchez y Leonidas Sanchez Valdeiglesias, quienes, desde el cielo, me cuidan y me llenan de fortaleza cada día. Su amor y legado siempre estarán presentes en mi corazón.*

*A mi madre, Pamela Evelyn Gheiler Sánchez, por su apoyo incondicional, su amor infinito y por ser mi ejemplo de enseñanza de vida. Gracias por ser mi roca y por enseñarme a luchar por mis sueños.*

*A Christian Adolfo Pazos Cuadros, quien me crio y guio mis pasos con sabiduría y amor, convirtiéndose en una figura pilar principal en mi vida. Todo lo que soy hoy es gracias a su guía y ejemplo.*

*Y a María Villar Chacon, la mujer de mis sueños, a quien amo profundamente. Gracias por tu amor, por creer en mí, y por caminar a mi lado en esta hermosa aventura que es la vida.*

## AGRADECIMIENTO

*A Mercedes Evelin Sanchez Duran, "Mamá Eve", a quien debo este trabajo de investigación y mucho más. Gracias por estar siempre a mi lado, en las buenas y en las malas, por cuidarme más a mí que tú, incluso cuando no estabas bien. Nunca olvidaré cómo cada día me despiertas para enseñarme a ser un hombre de bien, cómo estas siempre presente cuando necesito un consejo, un abrazo o simplemente compañía. Tu amor incondicional y sacrificios constantes, velando por mí día y noche, son la razón por la que hoy puedo alcanzar este logro. Este trabajo es un reflejo de tu esfuerzo, tu dedicación y tu inmenso amor. Gracias, Mamá Eve, por todo lo que hiciste y sigues haciendo por mí.*

*También agradezco profundamente a la Facultad de Ingeniería Informática y de Sistemas, a sus distinguidos docentes y a todos los amigos que hice en el camino. Ustedes, que compartieron conmigo los salones de clase y vivieron de cerca mi vida universitaria, fueron una parte esencial de este recorrido. Cada experiencia, cada lección y cada amistad ha dejado una huella imborrable en mí, y este logro también es fruto de todo lo que compartimos juntos.*

## ÍNDICE GENERAL

<b>DEDICATORIA.....</b>	<b>2</b>
<b>AGRADECIMIENTO .....</b>	<b>3</b>
<b>ÍNDICE DE FIGURAS.....</b>	<b>7</b>
<b>ÍNDICE DE TABLAS.....</b>	<b>9</b>
<b>RESUMEN.....</b>	<b>11</b>
<b>ABSTRACT.....</b>	<b>12</b>
<b>ACRÓNIMOS .....</b>	<b>13</b>
<b>1. CAPÍTULO I: ASPECTOS GENERALES.....</b>	<b>14</b>
1.1. PLANTEAMIENTO DEL PROBLEMA .....	14
1.2. FORMULACIÓN DEL PROBLEMA .....	16
1.3. PROBLEMA GENERAL .....	18
1.4. PROBLEMAS ESPECÍFICOS .....	18
1.5. JUSTIFICACIÓN.....	19
1.6. OBJETIVOS.....	20
1.6.1. <i>OBJETIVO GENERAL</i> .....	20
1.6.2. <i>OBJETIVOS ESPECÍFICOS</i> .....	20
1.7. ALCANCES Y LIMITACIONES.....	21
1.7.1. <i>ALCANCES</i> .....	21
1.7.2. <i>LIMITACIONES</i> .....	21
1.8. METODOLOGÍA .....	23
1.8.1. <i>TIPO DE INVESTIGACIÓN</i> .....	23
1.8.2. <i>MÉTODO DE INVESTIGACIÓN</i> .....	24

1.8.2.1.Fase descriptiva. ....	24
1.8.2.2.Fase experimental.....	24
<b>2.CAPÍTULO II: MARCO TEÓRICO.....</b>	<b>25</b>
2.1. ANTECEDENTES O ESTADO DEL ARTE .....	25
2.2. ALGORITMOS DE EMPAREJAMIENTO.....	30
2.2.1. <i>ELO</i> .....	30
2.2.2. <i>GLICKO-2</i> .....	34
2.2.3. <i>TRUESKILL</i> .....	39
2.3. CGPA (CUMULATIVE GRADE POINT AVERAGES).....	46
2.4. CALIFICACIÓN Y EVALUACIÓN ACADÉMICA EN LA UNSAAC .....	49
2.5. DISTRIBUCIÓN ASIMÉTRICA ACADÉMICA .....	52
<b>3. CAPÍTULO III: DESARROLLO DE ALGORITMOS DE EMPAREJAMIENTO .....</b>	<b>54</b>
3.1. OBTENCIÓN Y PRE-PROCESAMIENTO DE DATOS.....	54
3.2. ANÁLISIS DE INCONSISTENCIAS EN LOS DATOS CONFORME A LA RESOLUCIÓN N. <sup>o</sup> CU-0359-2015-UNSAAC .....	64
3.3. ANÁLISIS Y VISUALIZACIÓN DE DATOS DE REGISTROS DE NOTAS .....	68
3.4. TRANSFORMACIÓN DE DATOS.....	75
3.5. CÁLCULO DE GPGA DE LOS ESTUDIANTES.....	77
3.5.1. <i>PROMEDIO POR CURSO (ASIGNATURA)</i> .....	78
3.5.2. <i>PROMEDIO SEMESTRAL</i> .....	78
3.5.3. <i>CGPA FINAL</i> .....	79
3.6. CLASIFICACIÓN DE LOS ESTUDIANTES UTILIZANDO EL ALGORITMO ELO. ....	81
3.7. IDENTIFICACIÓN DE LOS ALUMNOS CON BAJO RENDIMIENTO DESPUÉS DE LA CLASIFICACIÓN POR EL ALGORITMO ELO.....	88

3.8. CORRELACIÓN ENTRE ELO GENERAL PONDERADO Y CGPA FINAL .....	92
3.9. CLASIFICACIÓN DE LOS ESTUDIANTES UTILIZANDO EL ALGORITMO GLICKO-2 .....	93
3.10. IDENTIFICACIÓN DE LOS ALUMNOS CON BAJO RENDIMIENTO DESPUÉS DE LA CLASIFICACIÓN POR EL ALGORITMO GLICKO-2 .....	104
3.11. CORRELACIÓN ENTRE GLICKO-2 GENERAL PONDERADO Y CGPA FINAL .....	107
3.12. CLASIFICACIÓN DE LOS ESTUDIANTES UTILIZANDO EL ALGORITMO TRUESKILL .....	108
3.13. IDENTIFICACIÓN DE LOS ALUMNOS CON BAJO RENDIMIENTO DESPUÉS DE LA CLASIFICACIÓN POR EL ALGORITMO TRUESKILL.....	118
3.14. CORRELACIÓN ENTRE TRUESKILL GENERAL PONDERADO Y CGPA FINAL .....	120
<b>4. CAPÍTULO IV: ANÁLISIS Y DISCUSIÓN DE RESULTADOS.....</b>	<b>122</b>
4.1. RESULTADOS EN INCONSISTENCIAS EN LOS DATOS CONFORME A LA RESOLUCIÓN N.º CU-0359- 2015-UNSAAC .....	122
4.2. RESULTADO DESCRIPTIVO GENERAL DE LOS DATOS ACADÉMICOS Y DETECCIÓN DE VALORES ATÍPICOS EN REGISTROS DE NOTAS.....	124
4.3. RESULTADOS COMPARATIVOS EN TIEMPOS DE EJECUCIÓN DE LOS ALGORITMOS ELO, GLICKO-2, TRUESKILL.....	126
4.4. COMPARACIÓN EN LA IDENTIFICACIÓN DE LOS ALUMNOS CON BAJO RENDIMIENTO DESPUÉS DE LA CLASIFICACIÓN POR EL ALGORITMO ELO, GLICKO-2 Y TRUESKILL .....	129
<b>CONCLUSIONES.....</b>	<b>149</b>
<b>RECOMENDACIONES .....</b>	<b>151</b>
<b>BIBLIOGRAFÍA.....</b>	<b>152</b>
<b>ANEXOS.....</b>	<b>154</b>

## ÍNDICE DE FIGURAS

<b>Figura 1 Reconocimiento estructural de las 10 primeras filas hecho por método lattice de la primera página en PDF .....</b>	<b>57</b>
<b>Figura 2 Reconocimiento tabular de las 10 primeras filas hecho por método lattice de la primera página en PDF .....</b>	<b>58</b>
<b>Figura 3 Comparación de tiempos acumulados con ajuste lineal de funciones de conversión de datos .....</b>	<b>62</b>
<b>Figura 4 Porcentaje de datos inconsistentes imputados .....</b>	<b>68</b>
<b>Figura 5 Alumnos matriculados por semestre del semestre 2023-I al 2024-II.....</b>	<b>69</b>
<b>Figura 6 Promedio de cursos matriculados por alumno del semestre 2023-I al 2024-II.....</b>	<b>70</b>
<b>Figura 7 Distribución de notas por parcial del año 2023 .....</b>	<b>71</b>
<b>Figura 8 Distribución de notas por parcial del año 2024.....</b>	<b>71</b>
<b>Figura 9 Distribución de notas atípicas por Semestre y Parcial.....</b>	<b>74</b>
<b>Figura 10 Histograma de distribución CGPA de los estudiantes con Curva KDE .....</b>	<b>79</b>
<b>Figura 11 Distribución y clasificación por Estrellas de ELO Ponderado General.....</b>	<b>90</b>
<b>Figura 12 Correlación entre ELO General Ponderado y CGPA Final de los alumnos clasificados .....</b>	<b>93</b>
<b>Figura 13 Distribución y clasificación por Estrellas de Glicko-2 Ponderado General .....</b>	<b>106</b>
<b>Figura 14 Correlación entre Glicko-2 General Ponderado y CGPA Final de los alumnos clasificados .....</b>	<b>108</b>
<b>Figura 15 Distribución y clasificación por Estrellas de TrueSkill Ponderado General .....</b>	<b>119</b>
<b>Figura 16 Correlación entre TrueSkill General Ponderado y CGPA Final de los alumnos clasificados .....</b>	<b>121</b>

<b>Figura 17 Eficiencia de los algoritmos ELO, Glicko-2 y TrueSkill por segundo y curso desde el semestre 2023-I al 2024-II.....</b>	128
<b>Figura 18 Tiempo de ejecución por semestre de los algoritmos de clasificación de los algoritmos ELO, Glicko-2 y TrueSkill desde el semestre 2023-I al 2024-II.....</b>	128
<b>Figura 19 Diagrama de Venn de la intersección de alumnos con bajo rendimiento según ELO, Glicko-2 y TrueSkill (2023-I a 2024-II).....</b>	130
<b>Figura 20 Distribución de promedios finales de los estudiantes clasificados con bajo rendimiento por algoritmos de emparejamiento: ELO, Glicko-2 y TrueSkill.....</b>	141
<b>Figura 21 Tasa de aprobación de estudiantes clasificados con bajo rendimiento por algoritmos de emparejamiento ELO, Glicko-2 y TrueSkill.....</b>	142
<b>Figura 22 Análisis temporal por promedio de estudiantes clasificados con bajo rendimiento por algoritmos de emparejamiento ELO, Glicko-2 y TrueSkill .....</b>	143
<b>Figura 23 Top 5 asignaturas más críticas desaprobadas identificados por algoritmos de emparejamiento ELO, Glicko-2 y TrueSkill.....</b>	145
<b>Figura 24 Distribución por colores de cursos en alumnos con bajo rendimiento identificados por algoritmos de emparejamiento ELO, Glicko-2 y TrueSkill .....</b>	148
<b>Figura 25 Anexo 1: Solicitud de acceso a registros académicos dirigida al Centro de Cómputo .....</b>	154
<b>Figura 26 Anexo 2: Pseudocódigo de implementación de algoritmo ELO por Alumno.....</b>	155
<b>Figura 27 Anexo 3: Pseudocódigo de implementación de algoritmo ELO por Curso y Ejecución .....</b>	156
<b>Figura 28 Anexo 4: Pseudocódigo de implementación de algoritmo Glicko-2 adaptativo por Alumno .....</b>	157

<b>Figura 29</b> Anexo 5: Pseudocódigo de implementación de algoritmo Glicko-2 por Curso y Ejecución.....	158
<b>Figura 30</b> Anexo 6: Pseudocódigo de implementación de algoritmo TrueSkill TTT-D por Alumno .....	159
<b>Figura 31</b> Anexo 7: Pseudocódigo de implementación de algoritmo TrueSkill TTT-D por Curso y Ejecución.....	160

## ÍNDICE DE TABLAS

<b>Tabla 1</b> Escala de calificación CGPA estándar.....	47
<b>Tabla 2</b> Hiperparámetros utilizados para el método “lattice” de Camelot .....	55
<b>Tabla 3</b> Percentiles por rendimiento para asignación de estrellas ELO.....	85
<b>Tabla 4</b> Resultados obtenidos después de la ejecución del algoritmo ELO en los semestres 2023-I, 2023-II, 2024-I y 2024-II.....	87
<b>Tabla 5</b> Percentiles por rendimiento general para asignación de estrellas ELO .....	89
<b>Tabla 6</b> Percentiles por rendimiento para asignación de estrellas Glicko-2 .....	102
<b>Tabla 7</b> Resultados obtenidos después de la ejecución del algoritmo Glicko-2 en los semestres 2023-I, 2023-II, 2024-I y 2024-II.....	103
<b>Tabla 8</b> Resultados obtenidos después de la ejecución del algoritmo TrueSkill (TTT-D) en los semestres 2023-I, 2023-II, 2024-I y 2024-II .....	117
<b>Tabla 9</b> Consistencia de registros de notas UNSAAC del semestre 2023-I al 2024-II por semestre .....	123
<b>Tabla 10</b> Descripción general de los registros de notas de la UNSAAC del semestre 2023-I al 2024-II.....	124

<b>Tabla 11</b> Valores atípicos en registros de notas por parcial de la UNSAAC de los algoritmos ELO, Glicko-2 y TrueSkill desde el semestre 2023-I al 2024-II .....	125
<b>Tabla 12</b> Tiempo de ejecución y promedio de cursos procesados por segundo de los algoritmos ELO, Glicko-2 y TrueSkill desde el semestre 2023-I al 2024-II.....	127
<b>Tabla 13</b> Estadísticas comparativas de alumnos con bajo rendimiento identificados por los algoritmos ELO, Glicko-2 y TrueSkill desde el semestre 2023-I al 2024-II .....	129
<b>Tabla 14</b> Análisis detallado de intersecciones y exclusividades en la identificación de alumnos con bajo rendimiento por los algoritmos ELO, Glicko-2 y TrueSkill.....	131
<b>Tabla 15</b> Muestra aleatoria de alumnos identificados exclusivamente por el algoritmo TrueSkill (TTT-D) como de bajo rendimiento académico (★).....	132
<b>Tabla 16</b> Muestra aleatoria de alumnos identificados exclusivamente por el algoritmo Glicko-2 como de bajo rendimiento académico (★) .....	134
<b>Tabla 17</b> Muestra aleatoria de alumnos identificados exclusivamente por el algoritmo ELO como de bajo rendimiento académico (★).....	136
<b>Tabla 18</b> Top 10 asignaturas más desaprobadas identificados por algoritmos de emparejamiento ELO, Glicko-2 y TrueSkill.....	144
<b>Tabla 19</b> Distribución de calificación cursos en alumnos con bajo rendimiento identificados por algoritmos de emparejamiento ELO, Glicko-2 y TrueSkill .....	147

## RESUMEN

Se evalúa la efectividad de los algoritmos de emparejamiento ELO, Glicko-2 y TrueSkill para clasificar a estudiantes con bajo rendimiento académico en la UNSAAC del periodo postpandemia (2023-I a 2024-II) habiendo brechas tecnológicas, sociales y pedagógicas. El objetivo determina en qué medida dichos algoritmos clasifican efectivamente el bajo rendimiento según el desempeño académico. Bajo un enfoque cuantitativo, descriptivo y experimental, se analizaron 413791 registros académicos de 25162 estudiantes. Tras procesos de limpieza, verificación (99,76 % registros válidos), imputación y simulación de competencias académicas, se aplicaron los tres modelos. Los resultados evidencian correlaciones positivas y significativas con el Cumulative Grade Point Average (CGPA). ELO ( $r = 0.7970$  y  $\rho = 0.4630$ ) destaca por su capacidad para identificar casos críticos (67.6 % de estudiantes con promedios inferiores a 5.0). Glicko-2 alcanzó la mejor correlación ( $r = 0.8213$  y  $\rho = 0.7948$ ), combinando precisión y estabilidad. TrueSkill ( $r = 0.8078$ ,  $\rho = 0.7783$ ) mostró una mayor sensibilidad en rangos limítrofes (9.5–13.5), identificando a 8.9 % de estudiantes en incertidumbre académica procesando aprox. 1.86 cursos por segundo. Los algoritmos identificaron tasas similares de bajo rendimiento (73.5% – 73.7%) y a 841 alumnos de forma unánime: ELO más eficaz en casos extremos, TrueSkill más sensible en rangos intermedios y con mayor correlación; y Glicko-2 más robusto y eficiente en tiempo de ejecución. Concluyendo, Glicko-2 y TrueSkill son los mejores algoritmos en la detección de estudiantes con bajo rendimiento en el periodo post-covid de la Universidad Nacional de San Antonio Abad del Cusco.

**Palabras clave:** ELO, Glicko-2, TrueSkill, Bajo rendimiento académico

## ABSTRACT

The effectiveness of the ELO, Glicko-2, and TrueSkill rating algorithms is evaluated for classifying students with low academic performance at UNSAAC during the post-pandemic period (2023-I to 2024-II), in a context marked by technological, social, and pedagogical gaps. The objective is to determine the extent to which these algorithms effectively classify low performance based on academic achievement. Under a quantitative, descriptive, and experimental approach, 413,791 academic records from 25,162 students were analyzed. After data cleaning, verification (99.76% valid records), imputation, and simulation of academic competitions, the three models were applied. The results show positive and significant correlations with the Cumulative Grade Point Average (CGPA). ELO ( $r = 0.7970$  and  $\rho = 0.4630$ ) stands out for its ability to identify critical cases (67.6% of students with averages below 5.0). Glicko-2 achieved the highest correlation ( $r = 0.8213$  and  $\rho = 0.7948$ ), combining accuracy and stability. TrueSkill ( $r = 0.8078$ ,  $\rho = 0.7783$ ) showed greater sensitivity in borderline ranges (9.5–13.5), identifying 8.9% of students in academic uncertainty while processing approximately 1.86 courses per second. The algorithms identified similar rates of low performance (73.5%–73.7%) and unanimously detected 841 students: ELO proved more effective in extreme cases, TrueSkill more sensitive in intermediate ranges and with higher correlation, and Glicko-2 more robust and efficient in execution time. In conclusion, Glicko-2 and TrueSkill are the most effective algorithms for detecting students with low academic performance during the post-COVID period at Universidad Nacional de San Antonio Abad del Cusco

**Keywords:** *ELO, Glicko-2, TrueSkill, Low academic performance*

## ACRÓNIMOS

- **CGPA** (*Cumulative Grade Point Average*). Promedio acumulado de calificaciones.
- **CSV** (*Comma-Separated Values*). Formato de archivo tabular separado por comas.
- **CU** (*Consejo Universitario*). Órgano de gobierno universitario que emite resoluciones en la UNSAAC.
- **ELO**. Sistema de calificación creado por Arpad Elo para estimar habilidades relativas en competiciones.
- **GPA** (*Grade Point Average*). Promedio de calificaciones.
- **INEI** (*Instituto Nacional de Estadística e Informática*). Entidad oficial de estadística del Perú.
- **MINEDU** (*Ministerio de Educación del Perú*). Entidad rectora de la política educativa nacional.
- **MMR** (*Matchmaking Rating*). Clasificación por emparejamiento, usada en videojuegos y aplicada al rendimiento académico.
- **PDF** (*Portable Document Format*). Formato de documento portátil.
- **RD** (*Rating Deviation*). Desviación de calificación en el sistema Glicko-2.
- **TIC** (*Tecnologías de la Información y Comunicación*). Herramientas digitales aplicadas a la educación.
- **TTT-D** (*TrueSkill Through Time with Individual Draw Margins*). Variante del algoritmo TrueSkill con márgenes de empate individuales.
- **UNSAAC** (*Universidad Nacional de San Antonio Abad del Cusco*). Universidad pública ubicada en Cusco, Perú.
- **W/L** (*Winner/Loser*). Notación en el algoritmo TrueSkill para ganador/perdedor.

## CAPÍTULO I: ASPECTOS GENERALES

### 1.1. Planteamiento del Problema

El rendimiento académico en los estudiantes universitarios es un indicador determinante para valorar la calidad educativa y el éxito en el aprendizaje. Sin embargo, en la actualidad, los métodos tradicionales de seguimiento académico en instituciones educativas superiores presentan limitaciones significativas: en primer lugar, los reportes de notas convencionales ofrecen únicamente una visión numérica del desempeño sin establecer comparaciones entre los estudiantes, lo que dificulta la identificación de patrones de rendimiento. Esto impide que los docentes tengan una visión más amplia del alumnado ni que puedan brindar una orientación personalizada a alumnos con bajo rendimiento, basada en sus necesidades individuales; tampoco ayuda a que los estudiantes comprendan su posición respecto a sus pares, reduciendo así su potencial de mejora y motivación para mejorar su desempeño.

En el ámbito nacional, la tasa bruta de matrícula universitaria alcanza un 41,9 %; sin embargo, esta cifra se reduce en regiones de la sierra sur, donde el acceso se ve limitado por factores socioeconómicos y geográficos (Ministerio de Educación Peruano & Diario El Peruano, 2020). En el caso específico de la ciudad del Cusco, la tasa de culminación universitaria presenta un panorama preocupante: apenas 27 % de los estudiantes logra graduarse dentro del tiempo teórico establecido, mientras que el abandono intermedio puede superar hasta el 40%, en determinadas carreras. La Universidad Nacional de San Antonio Abad del Cusco (UNSAAC), como principal referente de educación superior universitaria en la región, concentra una parte significativa de la matrícula total del departamento. El Decreto Supremo N.º 012-2020-MINEDU señala que esta casa de estudios registra una matrícula total que supera los 20 mil estudiantes distribuidos en programas de pregrado y posgrado, siendo el acceso mayoritario para jóvenes

provenientes de la misma región y de departamentos vecinos de la sierra sur. En cuanto a indicadores de permanencia, se evidencia que menos del 30% de los estudiantes logran egresar dentro del tiempo teórico establecido para sus programas de estudio. La tasa de abandono intermedio supera el 40 %, lo que constituye un problema vinculado tanto a factores económicos como académicos (Ministerio de Educación Peruano & Diario El Peruano, 2020).

El Decreto Supremo N.<sup>o</sup> 012-2020-MINEDU también destaca que, a pesar del licenciamiento institucional, persisten brechas en equipamiento tecnológico, cobertura de bibliotecas virtuales y conectividad digital, especialmente para estudiantes que provienen de zonas rurales. (Ministerio de Educación Peruano & Diario El Peruano, 2020).

A partir del año 2020, en la época de la pandemia del COVID-19, el sistema educativo se vio muy afectado a nivel mundial debido al cierre de las instituciones públicas y privadas, dando fin a la modalidad presencial; por consiguiente, implementando la modalidad de educación en línea para mantener una continuidad con la enseñanza. Sin embargo, la realidad se demostró que muchas instituciones no se encontraban preparadas para ese cambio tan significativo. Esto ahondó en las desigualdades relacionadas con el acceso y uso de las Tecnologías de la Información y Comunicación (TIC). De acuerdo con Vilela et al. (2021), la brecha en el acceso a estas tecnologías representó una gran limitación para que muchos estudiantes pudieran desarrollar con normalidad sus estudios universitarios, llevando incluso a la interrupción de su formación académica. Aun cuando el uso de las TIC alcanzó su mayor nivel de importancia durante la pandemia, estas condiciones desiguales continuaron afectando negativamente el desempeño estudiantil incluso en el escenario presencial del periodo postpandemia. (Alberto Caycho-Valencia et al., 2023).

## 1.2. Formulación del Problema

Producto de la pandemia, el contexto educativo experimentó un cambio profundo en la forma de enseñar de los docentes y de aprender de los estudiantes (Alberto Caycho-Valencia et al., 2023). Este cambio brusco de paradigma pedagógico generó dificultades de adaptación que incrementaron el número de alumnos con bajo rendimiento académico, influido además por factores familiares, sociales y de recuperación económica.

La Universidad Nacional de San Antonio Abad del Cusco evidencia una problemática crítica de eficiencia terminal, caracterizada por una brecha sostenida entre matrícula y titulación y un incremento agudo de la deserción estudiantil. Entre 2018 y 2022, la matrícula semestral se mantuvo en niveles altos y constantes (de 19432 a 19710 estudiantes) (Cañihua Florez & Meza de Loayza, 2018; Cañihua Florez et al., 2023). Sin embargo, el promedio de titulados por semestre fue de apenas 1267.5, ilustrando una baja tasa de culminación (Cañihua Florez et al., 2023; Cañihua Florez & Meza de Loayza, 2018). Esta brecha se agrava al analizar la deserción: las estimaciones pasaron de 1979 estudiantes en 2018 a 8435 en 2022, lo que representa un aumento superior al 400% en dicho periodo (Cañihua Florez & Meza de Loayza, 2018; Cañihua Florez et al., 2023).

Este escenario señala que, pese a una demanda constante, un número elevado de estudiantes no finaliza sus estudios, situación vinculada a factores como la prolongación de carrera y posterior abandono académico (INEI, 2023). A la problemática mencionada se añade la falta de sistemas de clasificación que permitan identificar a los estudiantes de bajo rendimiento, especialmente en los semestres postpandemia posteriores a los mencionados: del 2023-I al 2024-II en la Universidad Nacional de San Antonio Abad del Cusco. Esta carencia obstaculiza la implementación de

estrategias pedagógicas personalizadas a alumnos con bajo rendimiento, lo que agrava la situación y dificulta la búsqueda de una solución.

La presente investigación opta por algoritmos de emparejamiento (ELO, Glicko-2, TrueSkill) en lugar de métodos de aprendizaje automático supervisado o no supervisado debido a la naturaleza y estructura de los datos disponibles. Los datos de entrada, organizados por alumno, semestre, asignatura y notas parciales, carecen de una etiqueta predefinida que clasifique categóricamente el “rendimiento”. Implementar aprendizaje supervisado requeriría una costosa y subjetiva etiquetación manual de cada registro (Azlinah et al., 2020).

Por otro lado, el aprendizaje no supervisado, si bien opera sin etiquetas, genera agrupaciones (clusters) difíciles de interpretar y validar en este contexto; además las características adicionales cualitativas como el código y nombre de asignatura en los datos base no aportan información adicional a los clusters ni mucho menos tienen un tipo de correlación con el semestre académico. Estas agrupaciones se basan en similitudes estadísticas brutas entre notas, sin garantizar que se correspondan con categorías académicamente significativas o estables de rendimiento y riesgo. Por ello, no ofrecen una base accionable para su aplicación.

Los algoritmos de emparejamiento superan estas limitaciones al utilizar directamente la estructura comparativa inherente a los datos. Tratan cada curso y semestre como un entorno competitivo, donde las notas de los estudiantes se convierten en resultados de “enfrentamientos” simulados. Esto permite una clasificación relativa y dinámica basada en el desempeño comparativo, sin depender de etiquetas absolutas ni de la interpretación abstracta de agrupaciones (Ruiperez-Valiente et al., 2023; Vesin et al., 2022). Así, ofrecen una solución robusta, interpretable y directamente aplicable a la estructura matricial de rendimiento disponible.

### **1.3. Problema General**

¿En qué medida los algoritmos de emparejamiento: ELO, Glicko-2 y TrueSkill pueden proporcionar una clasificación precisa de los estudiantes con bajo rendimiento según su desempeño académico en la Universidad Nacional de San Antonio Abad del Cusco, en el periodo postpandemia (semestre Académico 2023-I al 2024-II)?

### **1.4. Problemas Específicos**

- ¿Cómo realizar el preprocesamiento de los datos académicos, garantizando su consistencia y validez para el análisis, incluyendo la descripción, limpieza, detección de valores atípicos y visualización de los registros de notas de los estudiantes de la UNSAAC durante los semestres 2023-I al 2024-II?
- ¿Cómo adaptar e implementar los algoritmos ELO, Glicko-2 y TrueSkill al contexto académico universitario para evaluar el rendimiento de los estudiantes?
- ¿Cuál de los algoritmos ELO, Glicko-2 o TrueSkill presenta el mejor desempeño en términos de tiempo de ejecución y correlación estadística con el Cumulative Grade Point Average (CGPA) de los estudiantes con bajo rendimiento académico?
- ¿Cómo se comparan los resultados de las clasificaciones generadas por los algoritmos ELO, Glicko-2 y TrueSkill en la identificación de estudiantes con bajo rendimiento, y qué diferencias se evidencian al aplicarlos?

## 1.5. Justificación

La presente investigación se justifica por la necesidad de implementar mecanismos eficaces que permitan la detección y atención oportuna de estudiantes universitarios con bajo rendimiento académico, particularmente en el contexto postpandemia, donde las brechas tecnológicas, sociales y pedagógicas agudizaron esta problemática. Existiendo casos registrados en los que estudiantes que, pese a atravesar períodos críticos de bajo rendimiento y riesgo de abandono, no recibieron tutorías especializadas ni estrategias por parte de la universidad de reinserción, lo que evidencia la carencia de sistemas efectivos para su identificación y acompañamiento, lo que prolonga su recuperación y limita sus oportunidades de mejora académica.

En este escenario, la aplicación de algoritmos de emparejamiento como ELO, Glicko-2 y TrueSkill, efectivamente utilizados en entornos competitivos como el ajedrez, Dota 2, Halo o League of Legends para clasificar y reconocer a jugadores con un bajo nivel de habilidad, se presenta como una alternativa innovadora y prometedora en el ámbito educativo. Estos modelos permiten realizar clasificaciones objetivas y adaptativas sin depender de grandes volúmenes de datos etiquetados, posibilitando la identificación precisa de estudiantes con bajo rendimiento y la generación de estrategias pedagógicas personalizadas orientadas a su mejora y permanencia en la universidad.

## 1.6. Objetivos

### 1.6.1. *Objetivo General*

Determinar en qué medida los algoritmos de emparejamiento ELO, Glicko-2 y TrueSkill pueden proporcionar una clasificación precisa de los estudiantes con bajo rendimiento, en la Universidad Nacional de San Antonio Abad del Cusco, según su desempeño académico.

### 1.6.2. *Objetivos Específicos*

- Realizar el preprocesamiento de los datos académicos de los estudiantes de la UNSAAC correspondientes a los semestres 2023-I al 2024-II, garantizando su consistencia y validez para el análisis mediante la descripción, limpieza, detección de valores atípicos y visualización de los registros de notas.
- Adaptar e implementar los algoritmos ELO, Glicko-2 y TrueSkill en el contexto académico universitario, con el propósito de evaluar y clasificar el rendimiento académico de los estudiantes.
- Evaluar el desempeño comparativo de los algoritmos ELO, Glicko-2 y TrueSkill en términos de tiempo de ejecución y correlación estadística con el Cumulative Grade Point Average (CGPA), determinando su efectividad en la identificación de estudiantes con bajo rendimiento académico.
- Comparar los resultados de las clasificaciones generadas por los algoritmos ELO, Glicko-2 y TrueSkill en la identificación de estudiantes con bajo rendimiento, analizando las diferencias al aplicarlos.

## **1.7. Alcances y limitaciones**

### ***1.7.1. Alcances***

Este estudio se desarrolla en la Universidad Nacional de San Antonio Abad del Cusco (UNSAAC), utilizando datos académicos desde el semestre 2023-I hasta el 2024-II, abarcando una población de 25162 estudiantes universitarios. Los datos recolectados incluyen las calificaciones parciales y finales de los alumnos, con los siguientes campos:

- id del alumno (anonimizado para proteger su privacidad)
- semestre académico
- código y nombre de la asignatura
- notas de evaluaciones (Parcial 1, Parcial 2, Parcial 3, Sustitutorio, Subsanatorio)
- promedio final

Los alumnos se clasifican mediante los algoritmos de emparejamiento ELO, Glicko-2 y TrueSkill; utilizando como métrica de evaluación la correlación el Cumulative Grade Point Average (CGPA) de cada estudiante con el fin de identificar a los estudiantes con rendimiento académico. Esto permite identificar los cuartiles inferiores de alumnos con bajo rendimiento y ofrece una relación basada en los resultados de clasificación de los dichos.

### ***1.7.2. Limitaciones***

El estudio enfrentará limitaciones relacionadas con la obtención y procesamiento de los datos. La recopilación de información académica requirió de un trámite formal ante el Centro de Cómputo de la UNSAAC, a través del sistema PLADDES, dirigido al Ing. Aguedo Huamani

Huayhua, director de dicha unidad, lo que generó demoras en la disponibilidad de la información. En la solicitud, se requirieron los datos correspondientes desde el semestre 2022-II hasta el 2024-II; sin embargo, a pesar de que en la cabecera del archivo entregado se consigna “NOTAS OBTENIDAS POR LOS ESTUDIANTES DEL 2022-2 AL 2024-2”, únicamente se recibieron registros desde el semestre 2023-I hasta el 2024-II, lo que reduce el alcance temporal inicialmente previsto.

Además, los registros fueron remitidos al correo institucional en un formato no estructurado (PDF), lo que demanda un considerable esfuerzo de limpieza, transformación y compilación de datos, al tratarse de más de 35 000 registros distribuidos en 3338 páginas. Este proceso no solo exige herramientas computacionales potentes y adecuadas, sino también un tiempo significativo para estandarizar la información antes de su análisis. Otra limitación relevante radica en la aplicabilidad de los resultados, ya que, al centrarse en el contexto específico de la UNSAAC, los hallazgos podrían no ser directamente extrapolables a otras instituciones con sistemas de evaluación distintos. Asimismo, la implementación de los algoritmos de emparejamiento (ELO, Glicko-2 y TrueSkill) requerirá una adaptación exhaustiva al formato de datos recibidos, así como el ajuste de sus fundamentos matemáticos y teóricos para garantizar exactitud y resultados óptimos, lo que implica numerosas pruebas de validación, detección de errores y optimización algorítmica.

## 1.8. Metodología

### 1.8.1. *Tipo de investigación*

La presente investigación adopta un enfoque cuantitativo con alcance descriptivo y experimental, orientado a analizar la efectividad de los algoritmos de emparejamiento (ELO, Glicko-2 y TrueSkill) en la detección temprana de estudiantes con bajo rendimiento académico en el contexto postpandemia de la Universidad Nacional de San Antonio Abad del Cusco.

En la fase descriptiva se indaga el impacto de la pandemia en el rendimiento académico de los estudiantes de la UNSAAC. Esto permitirá comprender las particularidades del contexto educativo actual y establecer líneas base para la investigación. Incluye el preprocesamiento, limpieza y visualización de datos; identificando distribuciones, tendencias, valores atípicos y patrones que puedan influir en la adaptación de los algoritmos de emparejamiento.

En la fase experimental, se adaptan modelos matemáticos de los algoritmos de emparejamiento a los registros notas obtenidas por los estudiantes de la UNSAAC del 2023-I al 2024-II y se mide el tiempo de ejecución. Se determina su correlación con el Cumulative Grade Point Average (CGPA) como métrica de evaluación, esto evalúa las capacidades de estos algoritmos para poder clasificar a los estudiantes, y por último se hallan a los estudiantes con bajo rendimiento aplicando cada uno de los algoritmos de emparejamiento: ELO, Glicko-2 y TrueSkill; por último, se compara los resultados y rendimiento de los algoritmos y se determina cual es el mejor.

Este enfoque permite validar empíricamente su utilidad en la clasificación estudiantil, proporcionó evidencia cuantitativa para futuras implementaciones en el ámbito educativo.

### **1.8.2. *Método de investigación***

El presente estudio emplea un método cuantitativo de diseño experimental, centrado en la aplicación y validación de algoritmos de emparejamiento (ELO, Glicko-2 y TrueSkill) para la clasificación del rendimiento académico; este enfoque permite validar empíricamente la utilidad de los algoritmos en un contexto educativo real. La investigación se estructura en las fases principales antes mencionadas:

**1.8.2.1. Fase descriptiva.** Esta fase comprende la preparación y adaptación de los datos académicos mediante cuatro procesos principales. Inicia con la transformación de datos no estructurados (PDF) a formato estructurado (CSV) y la medición de su complejidad algorítmica. Continúa con el análisis de inconsistencias conforme al Reglamento Académico (Resolución N.<sup>º</sup> CU-0359-2015-UNSAAC) y la imputación de datos faltantes. Posteriormente, se realiza el análisis descriptivo, visualización de datos e identificación de valores atípicos. Finalmente, se ejecuta la transformación estructural de los registros académicos mediante operaciones de melt y pivot table para adaptar los datos al formato requerido por los algoritmos de emparejamiento ELO, Glicko-2 y TrueSkill.

**1.8.2.2. Fase experimental.** La fase experimental implementa y evalúa los tres algoritmos de emparejamiento en el contexto académico. Comienza con el cálculo del CGPA de estudiantes como métrica de referencia. Posteriormente, se aplica de manera secuencial cada algoritmo (ELO, Glicko-2 y TrueSkill) siguiendo el mismo procedimiento: clasificación de estudiantes, identificación de aquellos con bajo rendimiento y cálculo de correlación entre el MMR ponderado del algoritmo y el CGPA Final. La fase culmina con un análisis comparativo de los tres modelos en la detección de estudiantes con bajo rendimiento académico en el contexto postpandemia la UNSAAC.

## CAPÍTULO II: MARCO TEÓRICO

### 2.1. Antecedentes o estado del arte

Zafari et al., (2021), en su investigación titulada “*A Practical Model for the Evaluation of High School Student Performance Based on Machine Learning*” propone un modelo práctico para evaluar el rendimiento de los estudiantes de secundaria basado en el aprendizaje automático. Este modelo se probó en una muestra de 1.000 estudiantes de secundaria de varias escuelas de Corea. Esto plantea la cuestión de que los métodos tradicionales para evaluar el rendimiento de los estudiantes, como las pruebas estandarizadas y las evaluaciones de los docentes, tienen limitaciones y pueden no reflejar con precisión las capacidades reales de los estudiantes. Además, estos métodos pueden consumir mucho tiempo y ser costosos. El sistema propuesto utiliza una combinación de selección de características y algoritmos de aprendizaje automático para evaluar el desempeño de los estudiantes. El sistema recopila datos de múltiples fuentes, incluidos registros académicos, patrones de comportamiento e información demográfica, luego utiliza técnicas de feature mapping para extraer las características más influyentes en la predicción de los resultados del aprendizaje de los estudiantes. Luego, el sistema entrena y evalúa varios modelos de aprendizaje automático, incluido el Decision Tree, Random Forest y los modelos de máquinas de soporte vectorial, para identificar los modelos más eficientes y óptimos para predecir el rendimiento de los estudiantes. El sistema fue capaz de clasificar a los estudiantes en cuatro clases (excelente, bueno, medio y mal) con una precisión del 85,5%. Los resultados del estudio sugieren que el sistema basado en el aprendizaje automático propuesto es efectivo para evaluar el rendimiento de los estudiantes de secundaria y puede proporcionar información valiosa para educadores y formuladores de políticas. El sistema tiene el potencial de ampliarse e implementarse en otros entornos educativos para mejorar los resultados de los estudiantes. Además, los datos

recopilados se pueden usar para otros trabajos, como la predicción de la deserción escolar y el rendimiento académico, agregando etiquetas relacionadas.

Reddick (2019), en el paper “*Using a Glicko-based Algorithm to Measure In-Course Learning*” propone y evalúa un algoritmo basado en Glicko para estimar la capacidad del alumno y la dificultad de los elementos en plataformas de aprendizaje en línea, con el objetivo de respaldar recomendaciones personalizadas y sistemas de aprendizaje adaptativo. La población estudiada son los estudiantes de cursos en línea en la plataforma de Coursera, quienes realizan evaluaciones para demostrar sus conocimientos y habilidades. El problema abordado en el paper es la necesidad de un enfoque confiable y escalable para medir la capacidad del alumno y la dificultad de los temas a lo largo del tiempo, para brindar recomendaciones personalizadas y respaldar los sistemas de aprendizaje adaptativo. Los enfoques tradicionales para medir la capacidad del alumno, como la teoría de respuesta al objetivo (IRT), pueden ser costosos desde el punto de vista computacional y requieren grandes cantidades de datos. La solución propuesta en el paper es un sistema de calificación basado en el algoritmo Glicko, que puede estimar la capacidad del alumno y la dificultad de los elementos a lo largo del tiempo utilizando datos de las evaluaciones del curso. El algoritmo Glicko es una variante del sistema de calificación Elo, que se usa comúnmente en ajedrez y torneos deportivos. El paper demuestra la eficacia de este enfoque para estimar la dificultad de los elementos y la capacidad del alumno, y muestra cómo se puede utilizar para respaldar recomendaciones personalizadas y sistemas de aprendizaje adaptativo. La solución propuesta en el paper es un sistema de calificación basado en el algoritmo Glicko, que puede estimar la capacidad del alumno y la dificultad de los elementos a lo largo del tiempo utilizando datos de las evaluaciones del curso. El algoritmo Glicko es una variante del sistema de calificación Elo, que se usa comúnmente en ajedrez y torneos deportivos. El paper demuestra la eficacia de este enfoque

para estimar la dificultad de los elementos y la capacidad del alumno, y muestra cómo se puede utilizar para respaldar recomendaciones personalizadas y sistemas de aprendizaje adaptativo.

Ruiperez-Valiente et al. (2023) en la investigación “*The Affordances of Multivariate Elo-Based Learner Modeling in Game-Based Assessment*” tiene como objetivo principal explorar los beneficios del modelo de aprendizaje basado en Elo multivariado en el contexto de la evaluación basada en juegos (GBA, por sus siglas en inglés). Este enfoque busca medir la competencia de los estudiantes en múltiples componentes de conocimiento, predecir su desempeño en tareas dentro del juego y estimar la dificultad de dichas tareas. Para llevar a cabo este estudio, se utilizó un conjunto de datos recolectado de 322 estudiantes de séptimo a décimo grado que participaron en clases de matemáticas y geometría. Los datos fueron obtenidos a través de la herramienta de evaluación llamada Shadowspect, la cual fue utilizada por siete profesores en diversas clases en los Estados Unidos durante un período mínimo de dos horas. El método implementado en el estudio consistió en un algoritmo adaptado del sistema de clasificación Elo, que permite realizar un modelado de aprendizaje multivariado. Este enfoque es particularmente útil, ya que permite evaluar simultáneamente las competencias de los estudiantes en diferentes componentes de conocimiento a partir de sus interacciones con los puzzles del juego. A través de metodologías específicas, se midieron las competencias, se predijo el rendimiento en las tareas y se estimó la dificultad de las mismas. Los resultados obtenidos mostraron una distribución diversa de competencias entre los estudiantes en los diferentes componentes de conocimiento. Se evidenció que el modelo de Elo multivariado es capaz de diferenciar entre las competencias de los estudiantes y predecir su desempeño en las tareas del juego. Además, se logró estimar la dificultad de las tareas de manera efectiva, lo cual es crucial para mantener un flujo de juego adecuado. En conclusión, el estudio resalta que el modelado de aprendizaje basado en Elo multivariado presenta numerosas

ventajas en el contexto de la evaluación basada en juegos. Este enfoque permite un modelado preciso del rendimiento del aprendiz sin la necesidad de acumular grandes cantidades de datos. Los autores sugieren que este modelo puede ser valioso para futuras prácticas educativas, ya que facilita una evaluación más dinámica y adaptativa del aprendizaje de los estudiantes en entornos de juego.

Lee Youngjin (2019) desarrolla la investigación “*Estimating student ability and problem difficulty using item response theory (IRT) and TrueSkill*” teniendo como objetivo investigar métodos eficientes para estimar la habilidad de los estudiantes en la resolución de problemas dentro de un entorno de aprendizaje basado en computadoras. La investigación se centra en cómo estas estimaciones pueden mejorar la personalización del aprendizaje y la predicción del éxito académico. Para llevar a cabo el estudio, se utilizó una población de estudiantes que interactuaron con un sistema de aprendizaje en línea, donde se recopilaron datos sobre su rendimiento en la resolución de problemas. El método principal empleado fue la Teoría de Respuesta al Ítem (IRT), que permite modelar la relación entre la habilidad del estudiante y la dificultad de los problemas, complementado por el algoritmo TrueSkill, que se utiliza para evaluar la dificultad de los problemas de manera más precisa. Los resultados del estudio indican que la combinación de IRT y TrueSkill proporciona estimaciones más precisas de la habilidad de los estudiantes y la dificultad de los problemas. Esto sugiere que es posible predecir el rendimiento académico de los estudiantes en exámenes finales basándose en su desempeño inicial en el curso. Además, se discute la posibilidad de desarrollar sistemas de alerta temprana que puedan estimar la probabilidad de deserción de los estudiantes en entornos de aprendizaje en línea, utilizando las habilidades estimadas a partir de su rendimiento en las primeras semanas del semestre. En conclusión, el estudio de Lee destaca la importancia de utilizar modelos de minería de datos para mejorar la

comprensión del aprendizaje y la enseñanza en entornos digitales, sugiriendo que futuras investigaciones podrían explorar más a fondo la efectividad de estos modelos predictivos en la educación.

El paper titulado “*Adaptive Assessment and Content Recommendation in Online Programming Courses: On the Use of Elo-rating*” de Vesin et al. (2022), presenta un enfoque innovador para mejorar la experiencia de aprendizaje en cursos de programación en línea mediante la implementación de un algoritmo de clasificación basado en el sistema de Elo, combinado con pruebas unitarias. El objetivo principal del estudio es evaluar la eficiencia y efectividad de este método adaptativo en la recomendación de contenido, con el fin de personalizar la experiencia de aprendizaje de los estudiantes. La población estudiada incluyó a 701 estudiantes que utilizaron el sistema ProTuS durante los semestres de otoño de 2019 y 2020. Sin embargo, para obtener datos significativos, se centraron en un grupo de 87 estudiantes que resolvieron más de 10 ejercicios de codificación. La investigación se llevó a cabo a lo largo de un semestre, abarcando dos generaciones de estudiantes, y se recopiló información a través de registros de interacción, clasificaciones generadas por el sistema y encuestas sobre los resultados de aprendizaje. El método utilizado consistió en la implementación del algoritmo de Elo para clasificar a los estudiantes según su interacción con los recursos de aprendizaje. Se introdujeron métricas para evaluar la calidad de las recomendaciones, la efectividad del método propuesto y la eficiencia del algoritmo modificado. Los resultados mostraron que la precisión y el recall de las recomendaciones de ejercicios de codificación fueron de 0.65 y 0.62, respectivamente, lo que indica un rendimiento aceptable del sistema en la personalización del contenido. En conclusión, el estudio sugiere que la combinación del algoritmo de Elo con un sistema de tutoría adaptativa puede mejorar significativamente la experiencia de aprendizaje en línea para los estudiantes de programación. Los hallazgos resaltan

la importancia de utilizar datos de interacción para personalizar las recomendaciones y fomentar un aprendizaje más efectivo, lo que podría tener implicaciones positivas tanto para los estudiantes como para los educadores en el ámbito de la educación en línea.

## **2.2. Algoritmos de emparejamiento**

Es de los principales pilares de los juegos competitivos online encargados de emparejar jugadores y equipos, cuya aplicación también se puede llevar al campo educativo como lo demuestra Reddick en el año 2019 en su investigación Using a Glicko-based Algorithm to Measure In-Course Learning. Estos sistemas modelan las habilidades de los participantes usando representaciones numéricas. Los números se actualizan después de cada competición en función de los resultados según Reddick (2019) aplicado a la educación, después de cada periodo educativo. Los sistemas de emparejamiento difieren principalmente en la forma en que actualizan esos números. Se presentan por lo tanto los tres para esta investigación los siguientes tres algoritmos más relevantes:

### **2.2.1. ELO**

Elo se ha utilizado como el principal algoritmo de emparejamiento para estimar estadísticamente las habilidades relativas de los jugadores en muchos juegos competitivos. Asume que la habilidad sigue una distribución Gaussiana con media  $\mu$  y una desviación estándar ajustada. Esta media es utilizada para clasificar la habilidad dentro de este modelo. En un espacio  $F$  donde  $N$  grupos compiten contra otros al mismo tiempo, la probabilidad de gacía por grupo  $t_i$  puede calcularse como:

$$\Pr(t, wins, F) = \frac{\sum_{i \leq j \leq N, i \neq j} (1 + e^{\frac{(\mu t_j - \mu t_i)}{D}})^{-1}}{\binom{N}{2}}$$

Donde  $\mu_{t_i}$  y  $\mu_{t_j}$  representan la habilidad del grupo  $t_i$  y grupo  $t_j$ . Y el parámetro  $D$  controla el impacto de la diferencia entre las calificaciones.

El resultado observado  $R_{t_i}^{obs}$  que representa el rango real de gacia del equipo  $t_i$  al final del encuentro, se normaliza para que los rangos observados suman 1:

$$R'_{t_i} = \frac{N - R_{t_i}^{obs}}{\binom{N}{2}}$$

La clasificación del equipo  $t_i$  se actualiza luego de la siguiente forma:

$$\mu'_{t_i} = \mu_{t_i} + K[R'_{t_i} - \Pr(t, wins, F)]$$

donde  $K$  es un factor de ponderación que determina la magnitud del cambio en la clasificación del equipo. Si el equipo con  $t_i$  consta de jugadores  $p_1, p_2, p_3, \dots, p_n$ , la clasificación de cada miembro del equipo  $p_j$  se actualiza como:

$$\mu'_{t_i} = \mu_{t_i} + w_{p_j}(\mu'_{t_i} - \mu_{t_i})$$

donde  $w_{p_j}$  es el peso de contribución del jugador  $p_j$ . Se calcula como la relación entre la clasificación de un jugador y la sumatoria de las calificaciones de cada uno de miembros del equipo (Dehpanah et al., 2021).

El anterior planteamiento delimita la actualización de las calificaciones por grupo, este enfoque da una referencia a como se interpreta inicia y matemáticamente el algoritmo ELO, pero para esta investigación en el espacio F los N grupos tienen que ser los mismos que el número de participantes, para ello se necesita que generalizar la especificidad de los autores Arman Dehpanah y Jonathan Gemmell.

Por ello se propone para caso de calificación de competencias individuales los modelos: Win-Loss (ganar-perder) y Draws (empates) que son fundamentales para entender cómo se aplica el algoritmo Elo. El modelo win-loss es el enfoque tradicional del algoritmo Elo, que se utiliza para calcular las calificaciones de los jugadores en juegos donde solo hay dos resultados posibles: ganar o perder. En este modelo, la probabilidad de que un jugador gane se relaciona directamente con su calificación en comparación con la calificación de su oponente. La fórmula básica del algoritmo Elo ajusta las calificaciones de los jugadores después de cada partida, basándose en el resultado (ganar o perder) y la diferencia de calificaciones entre los jugadores (Szczecinski & Djebbi, 2020). El modelo de Draws se introduce para abordar situaciones en las que hay un tercer resultado posible: el empate. Este modelo es crucial en deportes donde los empates son comunes, como el fútbol. Se argumenta que el algoritmo Elo tradicional asume implícitamente que la frecuencia de empates es del 50%, lo cual no es realista en la mayoría de los deportes o competiciones (Szczecinski & Djebbi, 2020).

Para mejorar la precisión del algoritmo en estos contextos, se propone el Elo-Davidson algorithm, que se basa en el modelo de draws de Davidson (1970). Este nuevo enfoque permite ajustar las calificaciones no solo en función de las victorias y derrotas, sino también considerando la frecuencia de empates. Esto se logra introduciendo un parámetro adicional que se relaciona con la frecuencia de empates, lo que proporciona una mejor representación de los resultados de los partidos (Szczecinski & Djebbi, 2020).

Basado en la especificación de Szczecinski & Djebbi con los modelos Win-Loss y Draws se implementa el siguiente modelo en esta investigación:

Como primer paso, se asigna un puntaje inicial  $ELO_i$  de 1000 para todos los competidores.

$$ELO_i = initial\_elo$$

$$ELO_i = 1000$$

Como segundo paso; calculamos la probabilidad esperada de ganar, que se explica en la investigación de Arman Dehpanah, en la cual se calcula la probabilidad esperada de que un competidor  $A$  con calificación  $R_A$  gane contra otro competidor  $B$  con rating  $R_B$ . Esta probabilidad se calcula con la fórmula.

$$P_A = \frac{1}{1 + 10^{\frac{R_B - R_A}{400}}}$$

Donde:

- $P_A$  es la probabilidad del que competidor  $A$  gane contra  $B$ ;
- $R_A$  y  $R_B$  son las puntuaciones ELO de los competidores respectivamente.

Como tercer paso; actualizamos las calificaciones ELO del competidor  $A$  tras un enfrentamiento contra  $B$  . La actualización se realiza de la siguiente manera:

$$R'_A = R_A + K(S_A - P_A)$$

Donde:

- $R'_A$  es la nueva puntuación ELO del estudiante  $A$ ;
- $K$  es el factor de ajuste (por defecto 32), que determina la magnitud de actualización;
- $S_A$  es el puntaje real del enfrentamiento: 1 si gana  $A$ , 0.0 si pierde y 0.5 si empata;
- $P_A$  es la probabilidad esperada de que A gane contra B.

Como cuarto paso; se simula las competiciones entre los estudiantes y actualiza las puntuaciones ELO en función de sus resultados totales. Para cada par de competidores  $(i, j)$  se calcula sus resultados totales  $S_i$  y  $S_j$ . Por lo que se compara  $S_i$  y  $S_j$ :

- Si  $S_i > S_j$ ,  $i$  gana y  $j$  pierde;

$$R'_i = R_i + K(1 - P_i) \wedge R'_j = R_j + K(0 - P_j)$$

- Si  $S_i < S_j$ ,  $i$  pierde y  $j$  gana;

$$R'_i = R_i + K(0 - P_i) \wedge R'_j = R_j + K(1 - P_j)$$

- Si  $S_i = S_j$ , sí hay empate.

$$R'_i = R_i + K(0.5 - P_i) \wedge R'_j = R_j + K(0.5 - P_j)$$

Como quinto paso; se ajusta el ELO inicial para evitar valores negativos al simular todas las competiciones y calcular el mínimo ELO obtenido  $ELO_{min}$ .

$$adjusted\_elo = 1000 - ELO_{min}$$

Esto asegura que el ELO más bajo sea igual o superior a cero. El algoritmo ELO implementado ajusta las puntuaciones basadas en el rendimiento relativo en competiciones simuladas, donde los resultados totales determina la calificación Elo total. Las actualizaciones ELO se basan en la comparación de los resultados de los competidores, ajustados para mantener los valores positivos (Szczecinski & Djebbi, 2020).

### 2.2.2. *Glicko-2*

En el sistema Glicko-2, cada jugador tiene una calificación ( $r$ ), una desviación de calificación (RD) y una volatilidad de calificación  $\sigma$ . La medida de volatilidad indica el grado de

fluctuación esperada en la calificación de un jugador. La volatilidad es alta cuando un jugador tiene desempeños erráticos (por ejemplo, cuando el jugador ha tenido resultados excepcionalmente buenos después de un período de estabilidad), y la volatilidad es baja cuando el jugador tiene un desempeño constante. Al igual que en el sistema Glicko original, suele ser preciso al resumir el rendimiento de un jugador en forma de un intervalo (en lugar de reportar solo una calificación). Una forma de hacerlo es reportar un intervalo de confianza del 95%. El valor más bajo del intervalo es la calificación del jugador menos dos veces la RD, y el valor más alto es la calificación del jugador más dos veces la RD (Glickman, 2022).

Para aplicar el algoritmo de calificación, tratamos una colección de partidas dentro de un período de calificación como si hubieran ocurrido simultáneamente. Los jugadores tienen calificaciones, RDs (desviaciones de calificación) y volatilidades al comienzo del período de calificación, se observan los resultados de las partidas, y luego se calculan las calificaciones, RDs y volatilidades actualizadas al final del período de calificación (que se utilizarán como información previa para el siguiente período de calificación). El sistema Glicko-2 funciona mejor cuando el número de partidas en un período de calificación es moderado a grande, es decir, un promedio de al menos 10-15 partidas por jugador en un período de calificación. La escala de calificación para Glicko-2 es diferente de la del sistema Glicko original. Sin embargo, es fácil alternar entre ambas escalas. Los siguientes pasos suponen que las calificaciones están en la escala original de Glicko, pero las fórmulas convierten a la escala de Glicko-2 y luego vuelven a convertir al final a Glicko (Glickman, 2022).

Como primer paso; se determina la calificación y el RD inicial para cada jugador en el período de calificación. La constante del sistema  $\tau$ , que limita el cambio en la volatilidad a lo largo del tiempo, este debe ser establecido antes de la aplicación del sistema, este valor tiene que estar

entre 0.3 y 1.2 y ser probado para obtener la máxima exactitud. Valores pequeños de  $\tau$  evita que la volatilidad cambie en grandes cantidades que a su vez evitan cambios enormes en las calificaciones basadas en resultados muy improbables. Si se espera que la aplicación de Glicko-2 trabaje con datos con resultados muy improbables lo mejor es ajustar el valor de  $\tau$  a 0.2. Si el jugador no tiene calificación, se establece la calificación a 1500 y el RD a 350, y la volatilidad a 0.06 (pudiendo variar este valor); caso contrario se usa el RD más reciente y la volatilidad  $\sigma$ .

Como segundo paso; para cada jugador se convierte las calificaciones y los RD's dentro de la escala de Glicko-2:

$$\begin{aligned}\mu &= (r - 1500)/173.7178 \\ \phi &= RD/173.7178\end{aligned}$$

El valor de la volatilidad  $\sigma$  no cambia. Ahora queremos actualizar la calificación del jugador con la calificación Glicko-2  $\mu$ , desviación de calificación  $\phi$  y volatilidad  $\sigma$ . El jugador compite contra  $m$  oponentes con calificaciones  $\mu_1, \dots, \mu_m$  y desviaciones de calificación  $\phi_1, \dots, \phi_m$ . Dejemos que  $s_1, \dots, s_m$  sean los puntajes contra cada oponente (0 al perder, 0.5 para empate y 1 para victoria). Las volatilidades oponentes no son importantes en este paso.

Como tercer paso; calculamos la cantidad  $v$ . Es la varianza estimada de la calificación equipo – jugador basado únicamente en los resultados.

$$v = \left[ \sum_{j=1}^m g(\phi_j)^2 E(\mu, \mu_j, \phi_j) \{1 - E(\mu, \mu_j, \phi_j)\} \right]^{-1}$$

donde:

$$g(\phi) = \frac{1}{\sqrt{1 + 3\phi^2/\pi^2}}$$

$$(\mu, \mu_j, \phi_j) = \frac{1}{1 + \exp(-g(\phi_j)(\mu - \mu_j))}$$

Como cuarto paso; calculamos la cantidad  $\Delta$ , la mejora estimada en la calificación comparando la calificación previa al periodo de calificación de desempeño basada únicamente en los resultados por juego.

$$\Delta = v \sum_{j=1}^m g(\phi_j) \{s_j - E(\mu, \mu_j, \phi_j)\}$$

Como quinto paso; determinamos el nuevo valor de  $\sigma'$  la volatilidad. Este cálculo es un proceso iterativo. Establecemos  $a = \ln(\sigma^2)$  y definimos

$$f(x) = \frac{e^x(\Delta^2 - \phi^2 - v - e^x)}{2(\phi^2 + v + e^x)^2} - \frac{(x - a)}{\tau^2}$$

También definimos una tolerancia a la convergencia con un valor suficientemente pequeño,  $\epsilon = 0.00001$

Establecemos los valores iniciales del algoritmo iterativo:

Asignamos  $A = a = \ln(\sigma^2)$ ;

- Si  $\Delta^2 > \phi^2 + v$ , entonces  $B = \ln(\Delta^2 - \phi^2 - v)$ ;
- Si  $\Delta^2 \leq \phi^2 + v$ , establecemos  $k = 1$ ;

- Si  $f(a - kr) < 0$ , entonces iterar  $k \leftarrow k + 1$  y establecer  $B = a - kr$  hasta que  $\Delta^2 > \phi^2 + \nu$ . Los valores de A y B se eligen en el intervalo  $\ln(\sigma'^2)$  y el resto del algoritmo se reduce iterativamente en ese intervalo.

Establecemos  $f_A = f(A) \wedge f_B = f(B)$ . Mientras  $|B - A| > \varepsilon$  se seguirán los siguientes pasos:

- Primero: establecer  $C = A + (A - B)f_A/(f_B - f_A)$  y  $f_C = f(C)$ ;
- Segundo: si  $f_C f_B \leq 0$  entonces establecer  $A \leftarrow B$  y  $f_A \leftarrow f_B$ ,
- caso contrario, sólo establecer  $f_A \leftarrow f_A/2$ ;
- Tercero: establecer  $B \leftarrow C \wedge f_B = f_C$ . Detenerse si  $|B - A| \leq \varepsilon$ . De lo contrario repetir los 3 pasos anteriores

Una vez  $|B - A| \leq \varepsilon$  establecer:

$$\sigma' \leftarrow e^{A/2}$$

Como sexto paso; actualizamos la desviación de la calificación al nuevo valor del periodo precalificación,  $\phi^*$

$$\phi^* = \sqrt{\phi^2 + \sigma'^2}$$

Como séptimo paso; actualizamos la calificación y el RD a los nuevos valores  $\mu'$  y  $\phi'$

$$\phi' = 1 / \sqrt{\frac{1}{\phi^{*2}} + \frac{1}{\nu}}$$

$$\mu' = \mu + \phi'^2 \sum_{j=1}^m g(\phi_j) \{s_j - E(\mu, \mu_j, \phi_j)\}$$

Como último paso; convertimos las calificaciones y RD a su escala original

$$r' = 173.7178\mu' + 1500$$

$$RD' = 173.7178\phi'$$

Si el jugador no compite durante el periodo de calificación, sólo se aplica el sexto paso. En ese caso los parámetros de calificación y volatilidad del jugador permanecen siendo los mismos, pero el RD incrementa de acuerdo a

$$\phi' = \phi^* = \sqrt{\phi^2 + \sigma^2}$$

(Glickman, 2022)

### 2.2.3. *TrueSkill*

Estimar las habilidades de los jugadores de manera retrospectiva permite considerar más información y, por consiguiente, se espera que conduzca a estimaciones más precisas en las calificaciones de los jugadores al considerar la información histórica. Arpad Elo fue el pionero en ello, al crear un método para poder calcular la habilidad de los jugadores, con un ajuste de curva suave a las estimaciones de habilidades en periodos de cinco años. Después, un mejor modelo con un enfoque estadísticamente más fundamentado fue desarrollado por Mark Glickman con los sistemas Glicko y Glicko-2, mejorando el método Elo utilizando un modelo bayesiano de habilidades como variables gaussianas, cuyas varianzas indican la fiabilidad de la estimación de la habilidad, una idea adoptada más tarde en el modelo TrueSkill. Glicko-2 añade medidas de volatilidad, que indican el grado de fluctuación esperada en la calificación de un jugador. Después

de una estimación inicial, las estimaciones pasadas se suavizan propagando la información hacia atrás en el tiempo (Dangauthier et al., 2007). Después fue introducido oficialmente por investigaciones de Microsoft, el algoritmo TrueSkill; utilizado en juegos como Halo o Gears of Wars, propiedades de la compañía. Está basado en la inferencia Bayesiana que combina factores de teoría de grafos y algoritmos de propagación de expectativas para calificar las habilidades de los jugadores (Dehpanah et al., 2021).

Considerar un número  $N$  de jugadores  $\{1, \dots, N\}$  que están compitiendo en un periodo  $T$ , digamos años. Denotar hay una serie de resultados entre dos jugadores  $i$  y  $j$  en el año  $t$  donde  $y_{ij}^t(k) \in \{+1, -1, 0\}$  donde  $k \in \{1, \dots, K_{ij}^t\}$  es el resultado de los jugadores disponibles para ese par de jugadores en ese año. Establecemos además  $y = +1$  si el jugador  $i$  gana,  $y = -1$  si el jugador  $i$  gana y por último  $y = 0$  en caso de empate.

En primera instancia se desarrolló el modelo TrueSkill base llamado Vanilla TrueSkill, en el que se asume cada jugador  $i$  tiene una habilidad desconocida  $s_i^t \in \mathbb{R}$  en un tiempo  $t$ . Asumimos que se genera el siguiente resultado  $y_{ij}^t(k)$ . Para cada uno de los rendimientos  $p_{ij}^t(k)$  y  $p_{ji}^t(k)$  de los dos jugadores  $i$  y  $j$ , respectivamente. Y el empate de acuerdo a  $p(p_{ij}^t(k)|s_i^t) = \mathcal{N}(p_{ij}^t(k); s_i^t; \beta^2)$ . El resultado  $y_{ij}^t(k)$  del juego entre los jugadores  $i$  y  $j$  esta determinado como:

$$y_{ij}^t(k) := \begin{cases} +1 & ; p_{ij}^t(k) > p_{ji}^t(k) + \varepsilon \\ -1 & ; p_{ij}^t(k) < p_{ji}^t(k) + \varepsilon \\ 0 & ; |p_{ij}^t(k) - p_{ji}^t(k)| \leq \varepsilon \end{cases}$$

Donde el parámetro  $\varepsilon > 0$  es el margen de empate. Para poder inferir las habilidades desconocidas  $s_i^t$  el modelo TrueSkill asume la factorización del prior Gaussiano  $p(s_i^0) = \mathcal{N}(s_i^0; \mu_0; \sigma_0^2)$  sobre las habilidades y deriva Gaussiana de habilidad entre los periodos de tiempo

dado por  $p(s_i^t | s_i^{t-1}) = \mathcal{N}(s_i^t; s_i^{t-1}; \tau^2)$ . En el modelo se define al jugador ganador como  $W$  y al perdedor como  $L$ , y se descarta al índice de tiempo por ahora; la inferencia Bayesiana aproximada (filtrado de densidad Bayesiana) conduce a las siguientes ecuaciones de actualización para  $\mu_W$ ,  $\mu_L$ ,  $\sigma_W$  y  $\sigma_L$ .

$$\begin{aligned}\mu_W &\leftarrow \mu_W + \frac{\sigma_W^2}{c_{ij}} \cdot v\left(\frac{\mu_W - \mu_L}{c_{ij}}, \frac{\varepsilon}{c_{ij}}\right) \wedge \sigma_W \leftarrow \sigma_W \sqrt{1 - \frac{\sigma_W^2}{c_{ij}^2} \cdot w\left(\frac{\mu_W - \mu_L}{c_{ij}}, \frac{\varepsilon}{c_{ij}}\right)} \\ \mu_L &\leftarrow \mu_L - \frac{\sigma_L^2}{c_{ij}} \cdot v\left(\frac{\mu_W - \mu_L}{c_{ij}}, \frac{\varepsilon}{c_{ij}}\right) \wedge \sigma_L \leftarrow \sigma_L \sqrt{1 - \frac{\sigma_L^2}{c_{ij}^2} \cdot w\left(\frac{\mu_W - \mu_L}{c_{ij}}, \frac{\varepsilon}{c_{ij}}\right)}\end{aligned}$$

La varianza general es  $c_{ij}^2 = 2\beta^2 + \sigma_W^2 + \sigma_L^2$  y las dos funciones  $v$  y  $w$  están dadas por:

$$v(t, \alpha) := \frac{\mathcal{N}(t - \alpha; 0; 1)}{\Phi(t - a)} \wedge w(t, \alpha) := v(t, \alpha) \cdot (v(t, \alpha) + (t - a))$$

En el caso de empate tenemos las siguientes ecuaciones de actualización:

$$\mu_i \leftarrow \mu_i + \frac{\sigma_i^2}{c_{ij}} \cdot \tilde{v}\left(\frac{\mu_i - \mu_j}{c_{ij}}, \frac{\varepsilon}{c_{ij}}\right) \wedge \sigma_i \leftarrow \sigma_i \sqrt{1 - \frac{\sigma_i^2}{c_{ij}^2} \cdot \tilde{w}\left(\frac{\mu_i - \mu_j}{c_{ij}}, \frac{\varepsilon}{c_{ij}}\right)}$$

Y de manera similar para el jugador  $j$ . Definimos  $d := \alpha - t$  y  $s := \alpha + t$  entonces  $\tilde{v}$  y  $\tilde{w}$  están dados por:

$$\tilde{v}(t, \alpha) := \frac{\mathcal{N}(-s; 0; 1) - \mathcal{N}(d; 0; 1)}{\Phi(d) - \Phi(-s)}$$

$$w(t, \alpha) := \tilde{v}^2(t, \alpha) + \frac{(d)\mathcal{N}(d; 0; 1) - (s)\mathcal{N}(s; 0; 1)}{\Phi(d) - \Phi(-s)}$$

Para poder aproximar los parámetros de habilidad  $\mu_i^t$  y  $\sigma_i^t$  para todos los jugadores  $i \in \{1, \dots, N\}$  para todos los tiempos  $t \in \{0, \dots, T\}$  para el modelo del algoritmo Vanilla TrueSkill se inicializa cada indicio de habilidad con  $\mu_i^0 \leftarrow \mu_0$  y  $\sigma_i^0 \leftarrow \sigma_0$ . Ahora se procede a través de los años  $t \in \{0, \dots, T\}$  en orden, se va desde los resultados  $y_{ij}^t(k)$  en orden aleatorio y se actualiza los indicios de habilidad de acuerdo a las ecuaciones mostradas.

(Dangauthier et al., 2007)

Después se encontró algunos problemas con respecto al algoritmo TrueSkill por lo tanto se desarrolló una mejora llamada “TrueSkill Through Time” (TrueSkill a través del Tiempo ó TTT) para corregirlos, las desventajas que surgen del algoritmo Vanilla TrueSkill son las siguientes:

La inferencia dentro de un año determinado  $t$  depende del orden aleatorio elegido para las actualizaciones. Dado que no se asume conocimiento sobre los resultados de los juegos dentro de un año dado, los resultados de la inferencia deberían ser independientes del orden de los enfrentamientos dentro de ese año.

La información entre los años solo se propaga hacia adelante en el tiempo. Más concretamente, si el jugador A vence al jugador B y luego el jugador B resulta ser muy fuerte (es decir, como se evidencia al vencer repetidamente a un jugador muy fuerte C), entonces TrueSkill Vanilla no puede propagar esa información hacia atrás en el tiempo para corregir al alza la estimación de la habilidad del jugador A.

Ambos problemas pueden abordarse extendiendo el filtrado de densidad Gaussiana para ejecutar la propagación completa de expectativas (EP) hasta la convergencia. La idea básica es actualizar repetidamente los mismos resultados de los juegos, pero asegurándose de que el efecto

de la actualización previa sobre ese resultado se elimine antes de añadir el nuevo efecto. De esta manera, el modelo se mantiene igual, pero las inferencias son menos aproximadas.

Más específicamente podemos ir a través de los resultados  $y_{ij}^t$  entre los años  $t$  varias veces hasta hallar una convergencia. La actualización para el resultado de un juego  $y_{ij}^t(k)$  es realizado de la misma manera que antes, pero almacena los mensajes ascendentes que pueden ser calculados de la siguiente manera:

$$m_{f(p_{ij}^t(k), s_i^t) \rightarrow p_{ij}^t(k)}(p_{ij}^t(k)) = \int_{-\infty}^{\infty} f(p_{ij}^t(k), s_i^t) \frac{p(s_i^t)}{m_{f(p_{ij}^t(k), s_i^t) \rightarrow s_i^t}} ds_i^t$$

Dividiendo así efectivamente el mensaje ascendente para evitar un doble conteo. La integral anterior se evalúa fácilmente ya que tanto los mensajes como los marginales  $p(s_i^t)$  han sido asumidos Gausianos. El nuevo mensaje ascendente sirve para el propósito de predicción efectiva del prior  $p_{ij}^t(k)$ . Al converger, la dependencia de las habilidades inferidas en el orden de los resultados desaparece.

El segundo problema se soluciona realizando una inferencia para TrueSkill a través del tiempo (TTT), es decir, suavizando repetidamente hacia adelante y hacia atrás en el tiempo. El primer paso hacia adelante de TTT es idéntico al paso de inferencia de Vanilla TrueSkill, excepto que los mensajes descendentes  $m_{f(s_i^{t-1}, s_i^t) \rightarrow s_i^t}(s_i^t)$  son almacenados. Ellos representan la influencia de la estimación de la habilidad  $s_i^{t-1}$  al tiempo  $t - 1$  y la estimación de habilidad  $s_i^t$  en el tiempo  $t$ . En el paso hacia atrás, estos mensajes se utilizan luego para calcular los nuevos mensajes hacia atrás, que sirven efectivamente como nuevo prior de el paso de tiempo  $t - 1$ , se calculan así:

$$m_{f(s_i^{t-1}, s_i^t) \rightarrow s_i^{t-1}}(s_i^{t-1}) = \int_{-\infty}^{\infty} f(s_i^{t-1}, s_i^t) \frac{p(s_i^t)}{m_{f(s_i^{t-1}, s_i^t) \rightarrow s_i^t}} ds_i^t$$

Este procedimiento es repetido hacia adelante y hacia atrás a través de las series de tiempo de habilidad hasta la convergencia. Los pasos hacia atrás hacen posible la propagación de la información del futuro al pasado.

(Dangauthier et al., 2007)

Después se consideró un ajuste más adecuado para los casos de empate por lo que se desarrolló el concepto de “TTT with Individual Draw Margins” (TrueSkill a través del Tiempo con márgenes de empate ó TTT-D).

Realizando una exploración de datos es conocido que la probabilidad de empate no solo incrementa de manera marcada, sino que esta positivamente correlacionada con las habilidades de los jugadores y varía considerablemente a través de cada jugador individualmente. Por eso se extiende el modelo TrueSkill para incorporar otro parámetro específico para cada jugador que indique la habilidad del jugador cuando se fuerza un empate. Supongamos que cada jugador  $i$  cada paso de tiempo  $t$  está caracterizado por su habilidad desconocida  $s_i^t \in \mathbb{R}$  y un jugador específico con un margen de empate  $\varepsilon_i^t > 0$ .

Otra vez, el rendimiento  $p_{ij}^t(k)$  y  $p_{ji}^t(k)$  y han empatado de acuerdo a  $p(p_{ij}^t(k)|s_i^t) = \mathcal{N}(p_{ij}^t(k); s_i^t; \beta^2)$ . En este modelo los resultados del juego  $y_{ij}^t(k)$  entre el jugador  $i$  y  $j$  en un tiempo  $t$  están generados de la siguiente manera:

$$y_{ij}^t(k) := \begin{cases} +1 & ; p_{ij}^t(k) > p_{ji}^t(k) + \varepsilon_i^t \\ -1 & ; p_{ij}^t(k) < p_{ji}^t(k) + \varepsilon_i^t \\ 0 & ; -\varepsilon_i^t \leq p_{ij}^t(k) - p_{ji}^t(k) \leq \varepsilon_i^t \end{cases}$$

Además del supuesto Gaussiano acerca de la habilidad de los jugadores en el modelo Vanilla TrueSkill, asumimos que factorizando la distribución Gaussiana específicamente para márgenes de empate en los jugadores  $p(\varepsilon_i^0) = \mathcal{N}(\varepsilon_i^0; v_0; \varsigma_0^2)$ .

El margen específico de empate de cada jugador en los pasos  $t$  están forzados por un factor  $> 0$ . La inferencia en el modelo TTT-D es realizado por la expectativa de propagación, dado un año  $t$  así como a través de los años de una forma hacia atrás y hacia adelante. Se nota en este modelo que las estimaciones actuales sobre la habilidad de los jugadores están representadas por cuatro números:  $\mu_i^t$  y  $\sigma_i^t$  para la habilidad; y  $v_i^t$  y  $\varsigma_i^t$  para el margen de empate específico para cada jugador. Un jugador con un valor alto de  $v_i^t$  puede ser considerado para obtener un empate contra un oponente fuerte, mientras que aquellos que tienen un alto valor de  $\mu_i^t$ , tienen una gran posibilidad de ganar.

En su investigación del año 2021 llamada “Evaluating Team Skill Aggregation in Online Competitive Games”, Arman Dehpanah, resume el algoritmo TrueSkill de la siguiente manera: Similar a Glicko, los jugadores en el modelo TrueSkill usan dos valores:  $\mu$  que es la calificación de habilidad y la desviación  $\sigma$  que representa la incertidumbre del sistema con respecto a la calificación. Ambos valores son actualizados después de cada enfrentamiento comparando sus clasificaciones predichas con las observadas. El método de actualización depende si hay posibilidad de empate. Para el caso de no haya empate si  $\mu_{t_i}, \mu_{t_j}, \sigma_{t_i}$ , y  $\sigma_{t_j}$  representa la calificación de la habilidad y las desviaciones de  $t_i$  y  $t_j$ ; asumiendo que  $t_i$  gana el enfrentamiento contra  $t_j$ , la calificación de habilidad es actualizada de la siguiente manera:

$$\mu'_{t_i} = \mu_{t_i} + \frac{\sigma_{t_i}^2}{c} \left[ \frac{\mathcal{N}(\frac{t}{c})}{\Phi(\frac{t}{c})} \right]$$

Donde  $t = \mu_{t_i} - \mu_{t_j}$  y  $c = \sqrt{2\beta^2 + \sigma_{t_i}^2 + \sigma_{t_j}^2}$ .  $\mathcal{N}$  y  $\Phi$  representan las funciones de probabilidad de densidad y distribución acumulada de distribución normal estándar. El parámetro  $\beta$  es el factor de escala que determina como la magnitud cambia con las calificaciones. La desviación de  $t_i$  es actualizada por:

$$\sigma'_{t_i} = \sigma_{t_i} - \sigma_{t_i}^2 \left( \frac{\sigma_{t_i}^2}{c^2} \left[ \frac{\mathcal{N}(\frac{t}{c})}{\Phi(\frac{t}{c})} \right] \left[ \frac{\mathcal{N}(\frac{t}{c})}{\Phi(\frac{t}{c})} + t \right] \right)$$

Al final TrueSkill suma las calificaciones individuales y las suma para cada jugador. La mayoría de los sistemas de calificación planteados y desarrollados posteriormente utilizaron un enfoque similar en sus cálculos

(Dehpanah et al., 2021).

### 2.3. CGPA (Cumulative Grade Point Averages)

Los GPAs (Grade Point Averages) son usados para expresar la capacidad académica de los estudiantes con un valor numérico único. Y existen varios tipos: Acumuladas, por Semestre, Ponderados y No Ponderados. El CGPA (Cumulative Grade Point Averages) es un promedio de todas las notas recibidas de los cursos en toda la secundaria. A cada curso se le asigna una cierta cantidad de créditos, y aquellos con una mayor cantidad de créditos tienen más valor que aquellos con una menor cantidad de créditos. Un GPA es el promedio calculado del valor numérico correspondiente de sus calificaciones (por eso significa "promedio de calificaciones"). La escala estandarizada es de 0,0 a 4,0, donde 4,0 equivale a una A y 0 equivale a una F. ([What Is Cumulative GPA? How Do You Calculate It?](#), 2024)

Para calcular el CGPA se realizan los siguientes pasos:

- Multiplicar los créditos de cada curso por el puntaje numérico correspondiente a la calificación que obtuviste. (En la Tabla 1 muestra las calificaciones y sus valores numéricos);
- Sumar el total de los puntos de calificación;
- Sumar la cantidad de créditos que obtuviste;
- Dividir el total de puntos de calificación por la cantidad total de créditos, redondeando al centésimo más cercano.

**Tabla 1***Escala de calificación CGPA estándar*

<b>Nota</b>	<b>Valor Porcentual</b>	<b>Calificación</b>
A/A+	93-100%	4.0
A-	90-92%	3.7
B+	87-89%	3.3
B	83-86%	3.0
B-	80-82%	2.7
C+	77-79%	2.3
C	73-76%	2.0
C-	70-72%	1.7
D+	67-69%	1.3
D	65-66%	1.0
F	Debajo de 65%	0.0

*Nota:* Tabla extraída de la publicación de CollegeVine “*What is Cumulative GPA, how do you calculate it?*”. Recuperado de la página <https://blog.collegevine.com/what-is-cumulative-gpa/> el día 14 de agosto del 2024.

En el contexto estadounidense, el CGPA ayuda a reconocer en que escuelas los estudiantes podrán obtener una mayor oportunidad de admisión o no, considerando también el aspecto holístico de cada institución. Las universidades analizan el contexto de cada clase específica y notarán quienes tienen un CGPA más alto en comparación a los otros compañeros de clase. Los alumnos que soliciten becas para ayudar a cubrir sus costos universitarios, tienen que considerar, que algunas becas tienen requisitos de CGPA, incluso si estos no fueran tan exigentes es mejor apuntar al GPA más alto posible para mejorar las posibilidades. Un GPA alto siempre es muy importante, especialmente cuando se solicita una beca selectiva. El GPA juega un papel importante en el proceso de admisión, ya que influye en el índice académico (IA). El IA es un número único que refleja la solidez del CGPA y las calificaciones de los exámenes. Muchas universidades selectivas usan el IA para descartar a los solicitantes incluso antes de considerar factores cualitativos como actividades extracurriculares y ensayos (*What Is Cumulative GPA? How Do You Calculate It?*, 2024).

Existen diversos factores que influyen en el rendimiento académico de los estudiantes con respecto a su CGPA: la automotivación, el proceso de enseñanza y aprendizaje, y la actitud hacia el curso son los principales factores que afectan a los estudiantes. Los estudiantes que tienen una mayor motivación tienden a obtener mejor resultados académicos; la calidad y la efectividad del proceso de enseñanza y aprendizaje también impactan positivamente en el CGPA. Un ambiente de aprendizaje favorable contribuye a un mejor rendimiento académico; y por último los estudiantes que muestran una actitud positiva hacia sus estudios tienden a tener un CGPA más

alto. Se sugiere instituciones educativas deben enfocarse en fomentar la automotivación y crear un ambiente de aprendizaje positivo y competitivo para mejorar el rendimiento académico de los estudiantes (Ahmad et al., 2017).

#### **2.4. Calificación y evaluación académica en la UNSAAC**

La evaluación académica es un proceso que permite valorar el logro de competencias por parte del estudiante. La Universidad Nacional de San Antonio Abad del Cusco (UNSAAC) regula dicho proceso mediante su Reglamento Académico, aprobado por Resolución N.<sup>o</sup> CU-0359-2015-UNSAAC con fecha 24 de diciembre de 2015, y posteriormente modificado por la Resolución N.<sup>o</sup> CU-093-2017-UNSAAC. Esta modificatoria fue aprobada bajo la gestión del Rector Dr. Baltazar Nicolás Cáceres Huambo, como resultado de la elevación de la Propuesta de Modificatoria del Reglamento Académico de la UNSAAC. Dicho documento normativo establece las disposiciones que regulan el sistema de evaluación, calificación, obtención de promedios, obligatoriedad de evaluaciones, mecanismos de recuperación y criterios de valoración del rendimiento académico, organizados en capítulos y artículos que rigen el accionar de docentes y estudiantes dentro del proceso educativo. En particular, las disposiciones relacionadas con la calificación y la obtención de promedios se encuentran recogidas en el Capítulo V: Normas Administrativas de la Evaluación, específicamente en el Subcapítulo II: De la calificación y obtención de promedios, que comprende los artículos 64 al 72. En estos artículos se norman aspectos como la utilización de la escala vigesimal, la ponderación de las evaluaciones, los criterios para la aprobación de asignaturas, los procedimientos de recuperación y la forma en que se calcula el promedio ponderado. Estas normas constituyen el marco legal y técnico que garantiza la objetividad, transparencia y equidad en el proceso de evaluación académica en la UNSAAC (VICERRECTORADO ACADEMICO UNSAAC, 2017).

En cuanto a la obtención de promedios, el artículo 65° establece tres tipos fundamentales: los promedios de periodo lectivo, los promedios promocionales o finales, y los promedios ponderados. Los primeros se obtienen considerando las calificaciones parciales obtenidas a lo largo del curso, ponderadas según el tipo de evaluación correspondiente, y se registran con precisión hasta los décimos de punto. Los promedios finales, por su parte, resultan del promedio de cada periodo lectivo al concluir el semestre. En estos casos, se considera como una unidad a favor del estudiante toda fracción igual o mayor a 0.5. Finalmente, los promedios ponderados se calculan multiplicando las calificaciones finales de cada asignatura por el número de créditos asignados, y dividiendo la suma total de estos productos entre el total de créditos, proceso que es gestionado por el centro de cómputo de la universidad.

La normativa también enfatiza la obligatoriedad de la evaluación (Art. 66°), estipulando que el estudiante debe asistir a todas las actividades programadas en el sílabo, incluyendo prácticas preprofesionales y seminarios. En caso de inasistencia injustificada a una evaluación, se le asigna una nota de cero (0), afectando así su promedio final (VICERRECTORADO ACADEMICO UNSAAC, 2017).

En lo que respecta a la valoración del rendimiento, el artículo 67° diferencia entre una escala cuantitativa y una cualitativa. Según la escala cuantitativa, una calificación entre 0 y 9 puntos se considera reprobada; entre 10 y 13, desaprobada; y entre 14 y 20, aprobada. Paralelamente, la valoración cualitativa clasifica el rendimiento como deficiente (0 a 8), malo (9 a 13), regular (14 a 16), bueno (17 a 18) y excelente (19 a 20), lo cual permite una interpretación más amplia del desempeño del estudiante (VICERRECTORADO ACADEMICO UNSAAC, 2017).

El reglamento también contempla mecanismos de flexibilización y recuperación académica. El artículo 68° permite la postergación de evaluaciones siempre que se solicite con anticipación al docente responsable y se justifique debidamente. En caso de no asistir a la fecha reprogramada, el estudiante pierde la oportunidad de una nueva evaluación y se le asigna la nota mínima (VICERRECTORADO ACADEMICO UNSAAC, 2017).

Asimismo, en situaciones donde el porcentaje de estudiantes aprobados en una evaluación sea igual o inferior al 35%, el artículo 69° obliga al docente a aplicar una nueva evaluación de conocimiento para los estudiantes desaprobados, siendo opcional para los aprobados y considerándose la mejor calificación obtenida (VICERRECTORADO ACADEMICO UNSAAC, 2017).

Por otro lado, el artículo 70° introduce la figura de la evaluación de sustitución, la cual permite al estudiante rendir nuevamente una prueba por única vez si obtuvo una calificación baja o no rindió una evaluación parcial. En caso de obtener una calificación superior, esta reemplaza a la anterior, con la salvedad de que dicha evaluación no puede sustituir al promedio de salida y no se aplica en cursos de recuperación o dirigidos (VICERRECTORADO ACADEMICO UNSAAC, 2017).

El artículo 71° regula la evaluación de subsanación para estudiantes por egresar, permitiendo que aquellos con un máximo de dos asignaturas desaprobadas, con promedios mínimos de diez (10) puntos, puedan solicitar una evaluación adicional. Si persiste la desaprobación y se demuestra una actuación indebida por parte del docente, el estudiante puede solicitar una evaluación alternativa con otro profesor del mismo Departamento Académico (VICERRECTORADO ACADEMICO UNSAAC, 2017).

Finalmente, el artículo 72º establece que la postergación de evaluaciones de recuperación debe ser gestionada ante el Decano de la Facultad, acompañando la solicitud con los documentos probatorios correspondientes para su debida autorización mediante resolución (VICERRECTORADO ACADEMICO UNSAAC, 2017).

## 2.5. Distribución Asimétrica Académica

La distribución en estadística es fundamental para entender la variabilidad en los datos. Se concibe como el patrón de variación en una o varias variables, permitiendo a los estadísticos describir y modelar cómo se distribuyen los valores observados en una población o muestra. La distribución puede ser empírica, que representa los datos observados directamente, o teórica, que es un modelo matemático que describe cómo debería comportarse esa variabilidad bajo ciertas condiciones. La diferencia clave radica en que la distribución empírica refleja lo que realmente se observa, mientras que la teórica permite hacer inferencias y predicciones sobre fenómenos subyacentes(Wild, 2006).

El concepto de sampling distribution (distribución muestral) es también esencial. Se refiere a la distribución de una estadística (como la media o proporción) calculada a partir de muchas muestras tomadas de la misma población. Aunque no podemos observarla directamente en una sola muestra, el estudio de estas distribuciones permite realizar inferencias sobre la población original, evaluando la variabilidad de las estimaciones (Wild, 2006).

En el ámbito académico y en programación con Python, existe el concepto de `distribution_type='academic'` que se usa en algunas librerías para modelar situaciones específicas y facilitar la comprensión de distribuciones asimétricas (no simétricas). Una distribución asimétrica, a diferencia de la normal, tiene sesgo y no es simétrica respecto a su media. Este

concepto es muy importante en estadística, ya que muchas variables del mundo real muestran distribuciones asimétricas, y actores como los modelos estadísticos y pruebas de hipótesis deben ajustarse a esta realidad. En esta investigación se hace uso de la distribución académica asimétrica para agrupar resultados generados generales, ya que es adecuada para registrar niveles de desempeño en contextos educativos. La distribución asimétrica, en particular, es frecuente en muchas situaciones del mundo real, como los niveles de desempeño académico, donde no todos los estudiantes alcanzan los mismos niveles de excelencia ni presentan un rendimiento uniforme. En estos casos, una mayoría puede situarse en el rango promedio o bueno, mientras que solo una pequeña proporción destaca por su rendimiento excepcional, y otra pequeña proporción presenta dificultades (García Tárrago, 2020).

## CAPÍTULO III: DESARROLLO DE ALGORITMOS DE EMPAREJAMIENTO

### 3.1. Obtención y Pre-procesamiento de Datos

Los datos de los alumnos de muestra se obtuvieron mediante un trámite formal mediante una solicitud al Ing. Aguedo Huamani Huayhua, director de la Unidad de Centro de Cómputo; ante el Centro de Cómputo de la UNSAAC a través del sistema PLADDES donde se obtuvieron los registros de notas obtenidas por los estudiantes del semestre 2023-1 al 2024-2 periodo post covid. Originalmente contenían las siguientes características los datos:

- Tipo de Archivo: .pdf (Portable Document Format);
- Espacio en Disco: 29.3 MB (30,788,795 bytes);
- Número de Páginas: 3338 páginas;
- Cantidad de Registros: 413791 filas;
- Número de Columnas Originales: 10;
- Nombre de columnas originales: id. Alumno, Semestre, CodAsignatura, Nombre, Parcial 1, Parcial 2, Parcial 3, Sustitutorio, Subsanatorio, Promedio Final.

El primer paso fue la lectura y transformación de datos a un formato más legible para poderla procesarla a través de la librería Pandas de Python. Por ello para empezar esta investigación empezó por transformar el conjunto de datos a .csv, un formato de datos más compacto y fácil de leer y escribir. Por lo que se utilizó la librería Camelot con el API complementaria ghostscript, instalándose mediante el instalador de paquetes pip de Python siguiendo los siguientes comandos en un entorno global: !pip install "camelot-py[cv]" -q y !apt-get install ghostscript ; que permiten la descarga, descompresión e instalación de las librerías.

Se implementa un algoritmo que procesa sistemáticamente la extracción de información tabular en partes desde documentos con formato PDF, esto aplica en principio “Divide and Conquer”, ya que procesa en múltiples archivos .csv los datos de registros de notas, lo que permite una liberación de memoria en tiempo de ejecución muy efectiva. El algoritmo comienza con la apertura y lectura del documento PDF que contiene los registros de notas obtenidos por los estudiantes del semestre 2023-1 al 2024-2 periodo post covid, operación cuya complejidad escala linealmente con el número de páginas del archivo. Esta fase inicial tiene una complejidad temporal de  $O(n)$  donde  $n$  representa el número total de páginas en el documento PDF. Una vez cargado el documento, el algoritmo procede página por página para identificar y extraer las estructuras tabulares contenidas en cada una; utilizando la biblioteca de Python, Camelot con el método 'lattice', el sistema identifica las líneas que conforman la estructura de las tablas, delimitando así las celdas individuales, sin embargo, este método por sí solo no es eficiente si no se configura adecuadamente los hiperparámetros, a continuación se da una propuesta la configuración utilizada según los autores:

**Tabla 2**

*Hiperparámetros utilizados para el método “lattice” de Camelot*

Nombre del hiperparámetro	Valor
flavor	'lattice'
process_background	False
line_tol	2
joint_tol	2
line_scale	30
split_text	True
layout_kwarg	'line_overlap': 0.5, 'char_margin': 0.5,

---

```
'word_margin': 0.1,  
'line_margin': 0.0,  
'boxes_flow': 0.5,  
'detect_vertical': False,  
'all_texts': False
```

---

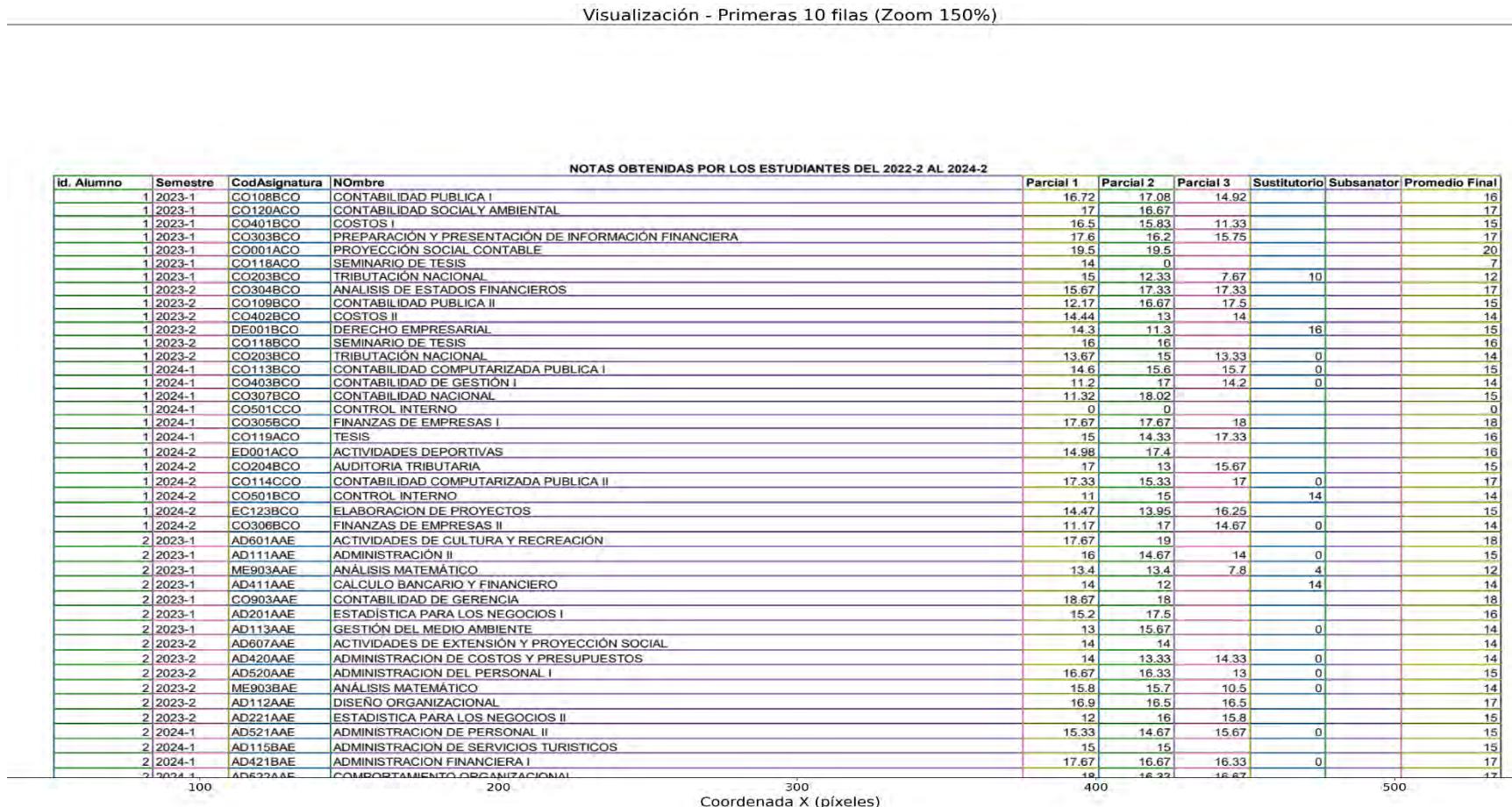
Fuente:<sup>a</sup>(Kekare et al., 2020). <sup>b</sup>(How It Works — Camelot 1.0.0 documentation, 2019).

Este proceso de extracción depende fundamentalmente de dos variables: el número de filas  $m$  y el número de columnas  $k$  en cada tabla. La complejidad de esta operación por página puede expresarse como  $O(m \cdot k)$ . Para cada tabla identificada, el algoritmo procesa su contenido celda por celda, extrayendo el texto y organizándolo en una estructura de datos matricial.

A continuación, en la Figura 1 y Figura 2, el algoritmo mediante la librería Camelot reconoce los datos de manera estructural y tabular:

**Figura 1**

*Reconocimiento estructural de las 10 primeras filas hecho por método lattice de la primera página en PDF*



Fuente: Elaboración personal

**Figura 2**

Reconocimiento tabular de las 10 primeras filas hecho por método lattice de la primera página en PDF

Visualización - Primeras 10 filas (Zoom 150%)

NOTAS OBTENIDAS POR LOS ESTUDIANTES DEL 2022-2 AL 2024-2

Id. Alumno	Semestre	CodAsignatura	Nombre	Parcial 1	Parcial 2	Parcial 3	Sustitutorio	Subsanador	Promedio Final
1 2023-1		CO109BCO	CONTABILIDAD PUBLICA I	16.72	17.08	14.92			16
1 2023-1		CO120ACO	CONTABILIDAD SOCIAL Y AMBIENTAL	17	16.67				17
1 2023-1		CO401BCO	COSTOS I	16.5	15.83	11.33			15
1 2023-1		CO303BCO	PREPARACIÓN Y PRESENTACIÓN DE INFORMACIÓN FINANCIERA	17.6	16.2	15.75			17
1 2023-1		CO001ACO	PROYECCIÓN SOCIAL CONTABLE	19.5	19.5				20
1 2023-1		CO119ACO	SEMINARIO DE TESIS	14	0				7
1 2023-1		CO203BCO	TRIBUTACIÓN NACIONAL	15	12.33	7.67	10		12
1 2023-2		CO304BCO	ANÁLISIS DE ESTADOS FINANCIEROS	15.67	17.33	17.33			17
1 2023-2		CO109BCO	CONTABILIDAD PUBLICA II	12.17	16.67	17.5			15
1 2023-2		CO402BCO	COSTOS II	14.44	13	14			14
1 2023-2		DE001BCO	DERECHO EMPRESARIAL	14.3	11.3		16		15
1 2023-2		CO119BCO	SEMINARIO DE TESIS	16	16				16
1 2023-2		CO203BCO	TRIBUTACIÓN NACIONAL	13.67	15	13.33	0		14
1 2024-1		CO113BCO	CONTABILIDAD COMPUTARIZADA PUBLICA I	14.6	15.6	15.7	0		15
1 2024-1		CO403BCO	CONTABILIDAD DE GESTIÓN I	11.2	17	14.2	0		14
1 2024-1		CO307BCO	CONTABILIDAD NACIONAL	11.32	18.02				15
1 2024-1		CO501BCO	CONTROL INTERNO	0	0				0
1 2024-1		CO305BCO	FINANZAS DE EMPRESAS I	17.67	17.67	18			18
1 2024-1		CO119ACO	TESIS	15	14.33	17.33			16
1 2024-2		ED001ACO	ACTIVIDADES DEPORTIVAS	14.98	17.4				16
1 2024-2		CO204BCO	AUDITORIA TRIBUTARIA	17	13	15.67			15
1 2024-2		CO114CCO	CONTABILIDAD COMPUTARIZADA PUBLICA II	17.33	15.33	17	0		17
1 2024-2		CO501BCO	CONTROL INTERNO	11	15		14		14
1 2024-2		EC123BCO	ELABORACIÓN DE PROYECTOS	14.47	13.95	16.25			15
1 2024-2		CO306BCO	FINANZAS DE EMPRESAS II	11.17	17	14.67	0		14
2 2023-1		AD601AAE	ACTIVIDADES DE CULTURA Y RECREACIÓN	17.67	19				18
2 2023-1		AD111AAE	ADMINISTRACIÓN II	16	14.87	14	0		15
2 2023-1		ME903AAE	ANÁLISIS MATEMÁTICO	13.4	13.4	7.8	4		12
2 2023-1		AD411AAE	CÁLCULO BANCARIO Y FINANCIERO	14	12		14		14
2 2023-1		CO903AAE	CONTABILIDAD DE GERENCIA	18.67	18				18
2 2023-1		AD201AAE	ESTADÍSTICA PARA LOS NEGOCIOS I	15.2	17.5				16
2 2023-1		AD113AAE	GESTIÓN DEL MEDIO AMBIENTE	13	15.67		0		14
2 2023-2		AD807AAE	ACTIVIDADES DE EXTENSIÓN Y PROYECCIÓN SOCIAL	14	14				14
2 2023-2		AD420AAE	ADMINISTRACIÓN DE COSTOS Y PRESUPUESTOS	14	13.33	14.33	0		14
2 2023-2		AD520AAE	ADMINISTRACIÓN DEL PERSONAL I	16.67	16.33	13	0		15
2 2023-2		ME903BAE	ANÁLISIS MATEMÁTICO	15.8	15.7	10.5	0		14
2 2023-2		AD112AAE	DISEÑO ORGANIZACIONAL	16.9	16.5	16.5			17
2 2023-2		AD221AAE	ESTADÍSTICA PARA LOS NEGOCIOS II	12	16	15.8			15
2 2024-1		AD521AAE	ADMINISTRACIÓN DE PERSONAL II	15.33	14.67	15.67	0		15
2 2024-1		AD115BAE	ADMINISTRACIÓN DE SERVICIOS TURÍSTICOS	15	15				15
2 2024-1		AD421BAE	ADMINISTRACIÓN FINANCIERA I	17.67	16.67	16.33	0		17
2 2024-1		AD604AAE	COMPORTAMIENTO ORGANIZACIONAL	18	16.33	16.67	17		17

Fuente: Elaboración personal

Esta estructura visualizada es posteriormente transformada en un archivo CSV, manteniendo la organización tabular original del documento. La complejidad de esta conversión y escritura también está determinada por las dimensiones de la tabla, siendo  $O(m \cdot k)$  por cada tabla procesada.

Paralelamente, el algoritmo registra los tiempos de ejecución para cada operación realizada en cada página. Este registro tiene una complejidad constante  $O(1)$ , ya que simplemente implica almacenar un valor numérico independientemente del tamaño o complejidad de la página procesada.

La complejidad total del algoritmo puede derivarse considerando todas las operaciones realizadas a lo largo del procesamiento completo del documento PDF. Para las  $n$  páginas del documento, cada una conteniendo potencialmente tablas de dimensiones  $m \times k$ , la complejidad temporal se puede expresar como:

$$O(n) + O(n \cdot m \cdot k) + O(n \cdot m \cdot k) + O(n) = O(n \cdot m \cdot k)$$

Aquí, el primer término representa la lectura inicial del documento, el segundo la extracción de tablas, el tercero la conversión y escritura a CSV, y el cuarto el registro de tiempos. Dado que los términos de menor orden se vuelven insignificantes a medida que aumenta el tamaño del problema, la complejidad es  $O(n \cdot m \cdot k)$ . Esta expresión nos indica que el tiempo de ejecución del algoritmo crece proporcionalmente con tres factores: el número de páginas en el documento, y las dimensiones (filas y columnas) de las tablas contenidas. En casos donde los documentos contienen tablas de gran tamaño distribuidas en numerosas páginas, el tiempo de procesamiento aumentará. Para procesar las 3338 páginas en formato PDF a .csv se requirió de un tiempo total de 9 horas, 3 minutos y 17 segundos, en una ratio promedio de conversión de 9.77 segundos por

iteración; resultando valga la redundancia 3338 archivos en formato .csv y un archivo con los tiempos de conversión con respecto a las iteraciones.

El siguiente algoritmo presenta segunda fase del proceso de extracción y consolidación de datos tabulares que se basa en la compilación de todas las partes individuales en una única para tener de manera uniformizada y reunida los datos. El algoritmo de agregación opera sobre un conjunto de archivos CSV fragmentados, denominados secuencialmente como `data_base_part_1`, `data_base_part_2`, hasta `data_base_part_3338`, que fueron generados por el algoritmo de extracción inicial. Su propósito fundamental es consolidar estos fragmentos dispersos en un único repositorio de datos coherente y completo. La ejecución comienza con la identificación y lectura secuencial de cada uno de  $k$  archivos CSV fragmentados. Para cada archivo  $i$ , donde  $1 \leq i \leq 3338$ , el algoritmo realiza una operación de lectura mediante la función `pd.read_csv()` de la librería Pandas, cargando su contenido en estructuras de datos temporales en memoria. Esta operación de lectura tiene una complejidad de  $O(n_i \cdot m)$  para cada archivo, donde  $n_i$  representa el número de filas del archivo  $i$  y  $m$  el número de columnas. Una vez cargado cada fragmento, el algoritmo procede a normalizar su estructura, asignando nombres estandarizados a las columnas mediante operaciones de remapeo. Esta estandarización asegura la consistencia estructural entre todos los fragmentos, preparándolos para su posterior unificación. La complejidad de esta operación de renombrado es  $O(m)$  para cada archivo, siendo relativamente insignificante en comparación con las operaciones de lectura y escritura.

Ocurre un fenómeno particular al en la anterior transformación de datos debido a que la columna denominada "Semestre", que erróneamente contiene el identificador de los estudiantes con la columna del semestre cursado; información concatenada es dividida en dos componentes distintos mediante operaciones de segmentación de texto. Esta división se aplica a cada una de las

$n_i$  filas del fragmento  $i$ , resultando en una complejidad de  $O(n_i)$  por archivo para esta operación específica. La fase crítica del algoritmo es la concatenación de todos los fragmentos procesados en una única estructura de datos cohesiva, esta operación es implementada mediante la instrucción `pd.concat()`, y tiene una complejidad de  $O(k \cdot n)$ , donde  $k$  es el número total de fragmentos (3338 en este caso) y  $n$  es el número promedio de filas por fragmento.

Finalmente, el algoritmo materializa esta estructura de datos consolidada en un archivo CSV único denominado "Base\_de\_Datos\_Notas\_alumnos.csv", que servirá como fuente centralizada para el análisis posterior de datos. La operación de escritura tiene una complejidad de  $O(k \cdot n \cdot m)$ , Paralelamente a estas operaciones, el algoritmo mantiene un registro de los tiempos de ejecución para cada iteración de este segundo proceso. Considerando todas las operaciones descritas, la complejidad temporal total del algoritmo de agregación puede expresarse matemáticamente como:

$$O\left(\sum_{i=1}^k n_i \cdot m\right) + O(k \cdot m) + O\left(\sum_{i=1}^k n_i\right) + O(k \cdot n) + O(k \cdot n \cdot m) = O(k \cdot n \cdot m)$$

Esta expresión matemática indica que el tiempo de ejecución del algoritmo escala linealmente con tres factores principales: el número de archivos fragmentados  $k$ , el número promedio de filas por fragmento  $k$ , y el número de columnas  $m$ .

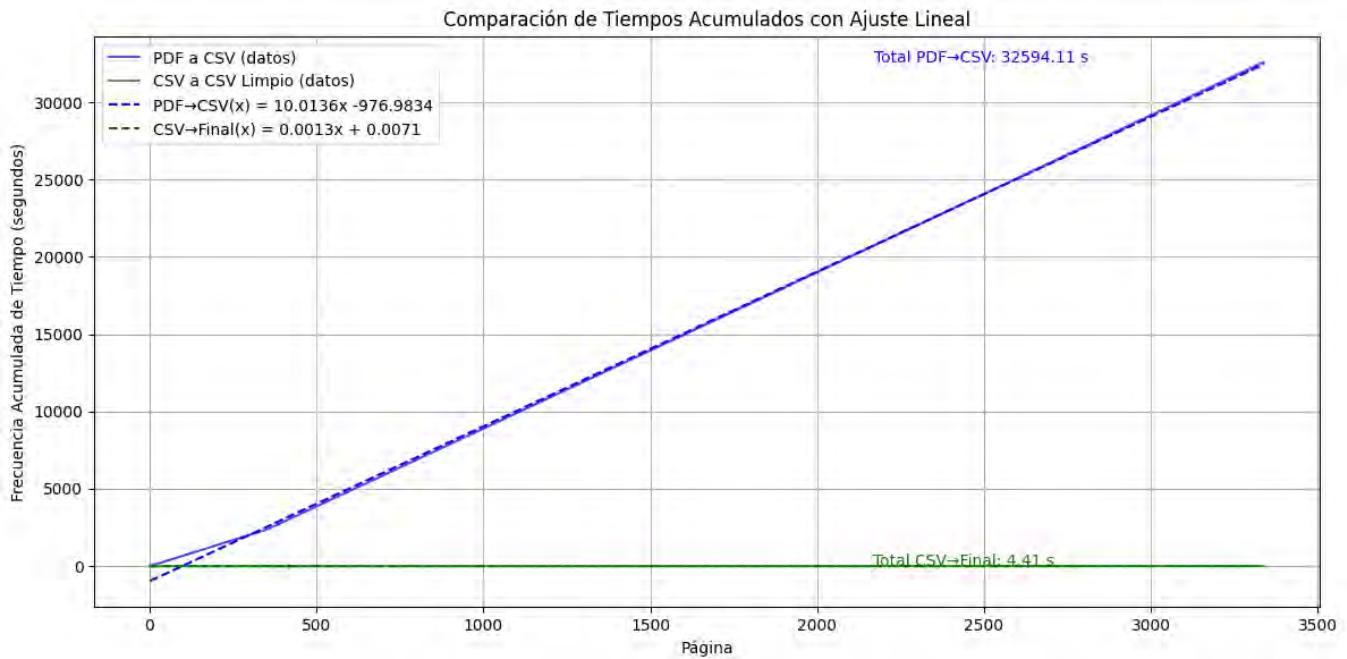
En el contexto de este algoritmo específico, con 3338 archivos fragmentados que deben procesarse, la complejidad práctica está significativamente influenciada por el factor  $k$ ; a parte se genera un archivo con los tiempos de conversión con respecto a las iteraciones. Para procesar los 3338 fragmentos .csv en formato a un .csv general se requirió de un tiempo total de 7 segundos, en una ratio promedio de conversión de 745.77 iteraciones por segundo; resultando un archivo

compilado en formato .csv de 29.8 MB (31,268,864 bytes) de espacio de disco. El número de estudiantes únicos encontrados fue de 25162

Este algoritmo de agregación, en conjunto con el algoritmo de extracción previo, constituye un flujo de trabajo completo para la transformación de datos tabulares desde documentos PDF hasta un formato estructurado y analizable, manteniendo íntegramente la información original mientras facilitando la lectura, exploración y análisis de los datos en los posteriores procesos.

**Figura 3**

*Comparación de tiempos acumulados con ajuste lineal de funciones de conversión de datos*



Fuente: Elaboración personal

La Figura 3 ilustra el análisis comparativo de los tiempos de ejecución acumulados para dos procesos algorítmicos secuenciales: la conversión de PDF a CSV (representada en azul) y la consolidación de múltiples archivos CSV en un único repositorio final (representada en verde).

La función lineal que caracteriza el primer proceso, PDF a CSV, está definida por la ecuación  $f(x) = 10.0136x - 976.9834$ , donde  $x$  representa el número de página y  $f(x)$  el

tiempo acumulado en segundos. El coeficiente de pendiente, aproximadamente procesa 10 segundos por página. El intercepto negativo de  $-976.9834$  segundos carece de significado físico directo, pero matemáticamente ajusta la función para optimizar la representación del comportamiento observado a lo largo de todo el dominio. Este proceso acumula un tiempo total de 6202.44 segundos (9 horas, 3 minutos y 17 segundos) para procesar las 3338 páginas.

En contraste marcado, la función que describe el segundo proceso, CSV a CSV Final, se expresa como  $g(x) = 0.0013x - 0.0071$ . Su pendiente significativamente menor, revela una eficiencia temporal aproximadamente 4656 veces superior a la del primer proceso con respecto al tiempo de ejecución. El tiempo total acumulado para este segundo algoritmo es de apenas 7 segundos, lo que constituye solo el 0.021% del tiempo requerido por el primer proceso.

A pesar de que ambos algoritmos presentan una complejidad teórica similar a  $O(n \cdot m \cdot k)$ , su rendimiento práctico difiere significativamente. Esta disparidad se debe principalmente a los coeficientes constantes ocultos en la notación asintótica y a la naturaleza de las operaciones que ejecutan. El primer algoritmo, encargado de extraer datos estructurados desde archivos PDF, involucra un procesamiento intensivo de imágenes para interpretar la información visual, identificar elementos y convertirlos. En contraste, el segundo algoritmo opera sobre datos ya estructurados, realizando tareas mucho más ligeras de lectura, transformación y concatenación.

La linealidad observada en el tiempo de ejecución de ambos algoritmos, confirmando su complejidad teórica, contrasta con la gran diferencia en sus pendientes. Esta diferencia revela que el primer algoritmo es aproximadamente 845 veces más lento que el segundo debido a las constantes multiplicativas asociadas a la complejidad de sus operaciones. Aunque ambos escalan linealmente con la cantidad de datos, la tarea de procesar archivos PDF introduce una sobrecarga constante mucho mayor. La intersección cercana a cero con el eje vertical indica una baja

sobrecarga de inicio para ambos, y la consistencia de la pendiente sugiere un comportamiento estable en su rendimiento independientemente del volumen de datos.

### **3.2. Análisis de inconsistencias en los datos conforme a la Resolución N.<sup>o</sup> CU-0359-2015-UNSAAC**

En el marco normativo de la Universidad Nacional de San Antonio Abad del Cusco (UNSAAC), el proceso de evaluación académica está debidamente reglamentado en su Reglamento Académico, aprobado por Resolución N.<sup>o</sup> CU-0359-2015-UNSAAC con fecha 24 de diciembre de 2015 y posteriormente modificado por la Resolución N.<sup>o</sup> CU-093-2017-UNSAAC. Esta última fue aprobada bajo la gestión del Rector Dr. Baltazar Nicolás Cáceres Huambo, producto de la propuesta de modificación del reglamento. Este documento establece las pautas para la calificación, obtención de promedios, y criterios de valoración del rendimiento académico, regulando así el accionar tanto del cuerpo docente como del estudiantado (VICERRECTORADO ACADEMICO UNSAAC, 2017)

Bajo estas disposiciones, y tomando como referencia el Capítulo V, Subcapítulo II: De la calificación y obtención de promedios, artículos 64 al 72 del reglamento, se realizó un análisis cuantitativo de las calificaciones de los estudiantes a fin de identificar posibles inconsistencias entre las notas parciales, las notas de recuperación (subsanatorio o sustitutorio), y el promedio final consignado en el registro académico. Para este propósito, se elaboró una función de redondeo especial que simula la regla interna de la universidad, en la que si el decimal de la nota es igual o mayor a 0.49999999999996, esta se aproxima al entero superior; en caso contrario, se aproxima al entero inferior. Este umbral específico se emplea para evitar errores derivados de la aritmética de punto flotante, ya que ciertos valores como 0.5 no siempre se representan con exactitud en los sistemas computacionales (VICERRECTORADO ACADEMICO UNSAAC, 2017).

A través de un algoritmo de verificación, se definieron criterios de consistencia según el tipo de recuperación académica: el subsanatorio debe reemplazar el promedio base únicamente si su valor es mayor, mientras que el sustitutorio puede influir positivamente siempre y cuando eleve la nota final. Para estudiantes sin notas de recuperación, se espera que el promedio final coincida con el promedio base redondeado (VICERRECTORADO ACADEMICO UNSAAC, 2017).

Desde una perspectiva matemática, la verificación de consistencia se formaliza como una función booleana  $C(x)$  definida sobre cada registro  $x$  del conjunto de datos, tal que:

$$C(x) = \begin{cases} \text{True, } & \text{si } P_f = \max(P_b, S_u) ; \forall(S_u > P_b) \\ \text{True, } & \text{si } P_f = P_b ; \forall(S_u \leq P_b) \vee S = \text{null} \\ \text{True, } & \text{si } P_f \geq P_b ; \exists S \\ \text{False, } & \text{caso contrario} \end{cases}$$

Donde:

- $P_f$ : es el promedio final;
- $P_b$ : es el promedio base calculado con notas parciales y redondeo especial;
- $S_u$ : es la nota del subsanatorio redondeado;
- $S$  : es el examen sustitutorio semestral.

Después calculamos la tasa de consistencia, siguiendo la anterior lógica: hallamos la cantidad de registros consistentes y no consistentes, planteando la siguiente fórmula:

$$\text{Rate}_{\text{consistency}}(\%) = (N_c/N_t) * 100$$

El análisis fue aplicado al total de registros académicos de notas evaluados ( $N_t = 413791$ ). Como resultado, se encontró que ( $N_c = 412797$ ) registros presentaban una

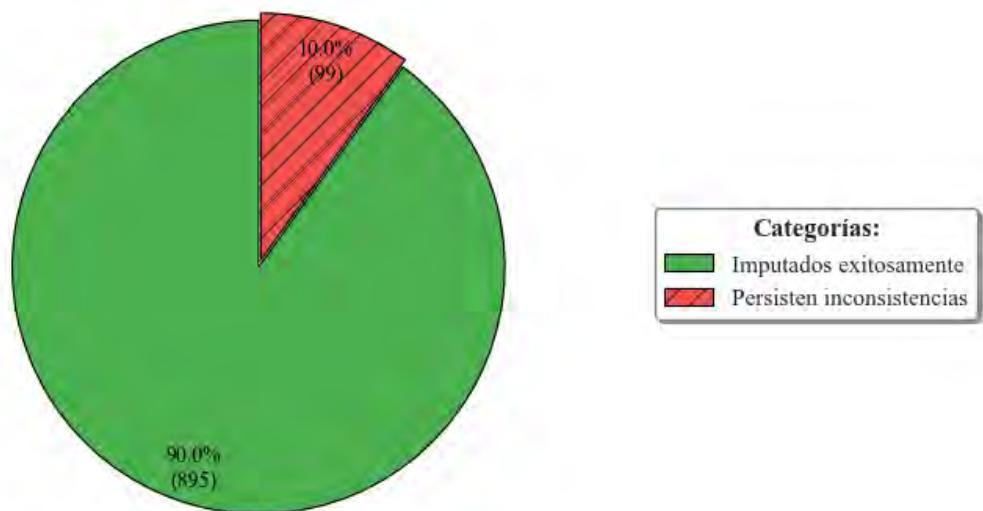
estructura de calificaciones consistente, lo que representa un  $Rate_{consistency} = 99.76\%$  con respecto al total. Sin embargo, se identificaron  $N_t - N_c = 994$  registros inconsistentes (0.24 %), en los que el promedio final no guarda correspondencia lógica con las reglas institucionales definidas. Estos resultados sugieren una alta confiabilidad en la aplicación de las normas de calificación, aunque la existencia de errores, aunque mínimos, debe ser abordada para garantizar la integridad académica. Este tipo de verificación es realizado para detectar errores sistemáticos o humanos en la digitalización de notas y comprobar las reglas formales de validación.

Se encontrados dos tipos de inconsistencias con dos comportamientos diferenciados. La primera, cuando las notas de los exámenes parciales  $P_1, P_2, P_3$  para cursos de 4 créditos o  $P_1, P_2$  para cursos de 3 o 2 créditos; están vacíos o son cero a pesar de tener una nota de Promedio Final mayor a cero, esto se vio evidenciado los siguientes cursos:

- INTERNADO CLÍNICO
- PRACTICAS PRE-PROFESIONALES/ INTERNADO FARMACEUTICO II
- PRACTICAS PRE-PROFESIONALES/ INTERNADO FARMACEUTICO I
- INTERNADO EXTRAHOSPITALARIO
- INTERNADO HOSPITALARIO
- INTERNADO DE MEDICINA
- INTERNADO DE PEDIATRIA
- INTERNADO DE GINECO-OBSTETRICIA
- INTERNADO DE CIRUGIA

- PRÁCTICAS PRE PROFESIONALES AREA RURAL
- INTERNADO RURAL
- PRÁCTICAS PRE PROFESIONALES AREA HOSPITALARIA
- SEMIOLOGIA Y FISIOPATOLOGIA CARDIOVASCULAR
- INTERNADO DE GINECOOBSTETRICIA

Esto nos podría indicar que algunos cursos (como internados, prácticas, áreas rurales) podrían tener un sistema de evaluación distinto, no basado en exámenes parciales donde la evaluación tenga un enfoque más cualitativo o continuo, con supervisión directa o informes de desempeño periódicos de parte de un médico evaluador. El segundo tipo de inconsistencia que se encontró fue cuando la nota del examen subsanatorio no reemplaza la nota del promedio parcial final a pesar de ser una nota mayor, esto podría suceder debido a una migración de datos mal realizada, actualización de plataformas o error en la validación del sistema. A la primera inconsistencia la reconoceremos como imputable, debido a que es posible imputar las notas faltantes de los parciales con respecto a la nota final, la segunda inconsistencia sería no imputable debido a que la inconsistencia solo se da en el examen subsanatorio y no afecta directamente a las notas parciales ni al final; en la Figura 4 se representa la proporción de los datos inconsistentes después de la imputación:

**Figura 4***Porcentaje de datos inconsistentes imputados***Estado de las Inconsistencias después de la Imputación***Fuente: Elaboración personal***3.3. Análisis y Visualización de Datos de Registros de Notas**

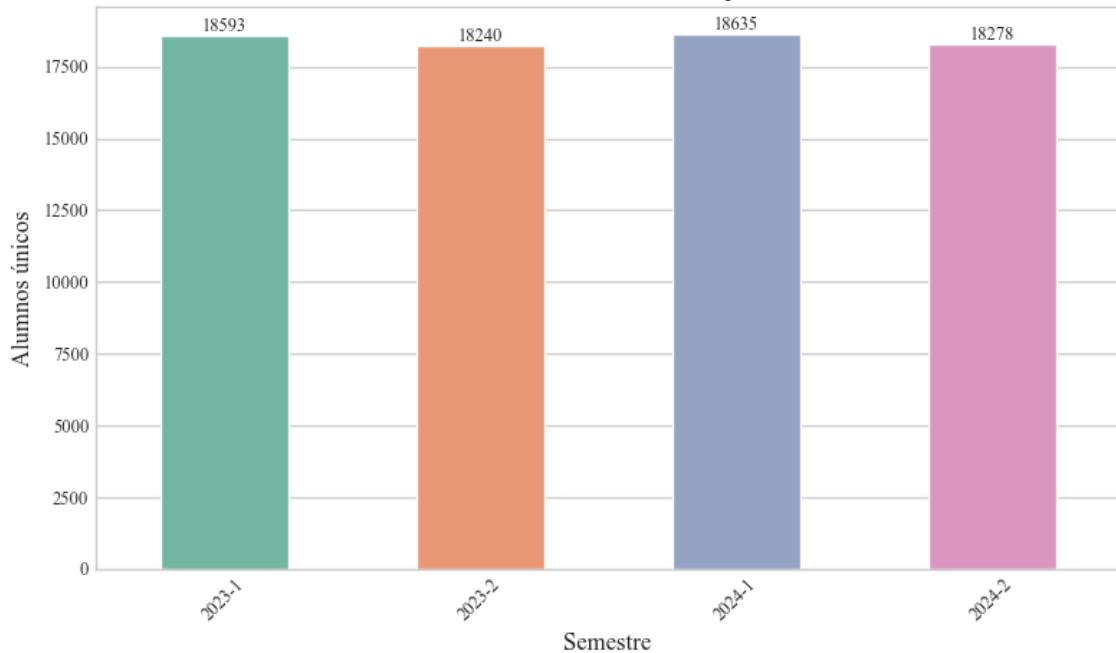
El análisis de datos y su visualización gráfica constituyen elementos importantes en la investigación cuantitativa, ya que transforman conjuntos de datos brutos —en este caso los registros de estudiantes de la Universidad Nacional de San Antonio Abad del Cusco (UNSAAC) del semestre 2023-I al 2024-II— en conocimiento accionable, permitiendo identificar patrones y el comportamiento de los datos, validando el riesgo en el cual se encuentran los alumnos al tener un bajo rendimiento con peligro a deserción o abandono estudiantil. De este modo, las representaciones gráficas sintetizan la complejidad de los datos multivariados inherentes al contexto universitario de esta investigación, facilitando el descubrimiento de hallazgos, al exponer de manera intuitiva la naturaleza de los datos ya compilados. La sinergia entre análisis y

visualización garantiza transparencia metodológica y replicabilidad en el caso específico de la UNSAAC, provee evidencia formal para aplicar los algoritmos de clasificación, descubriendo así cuales son los alumnos con bajo rendimiento. A continuación, se visualizan los datos en los siguientes gráficos:

**Figura 5**

*Alumnos matriculados por semestre del semestre 2023-I al 2024-II*

Cantidad de alumnos matriculados por semestre

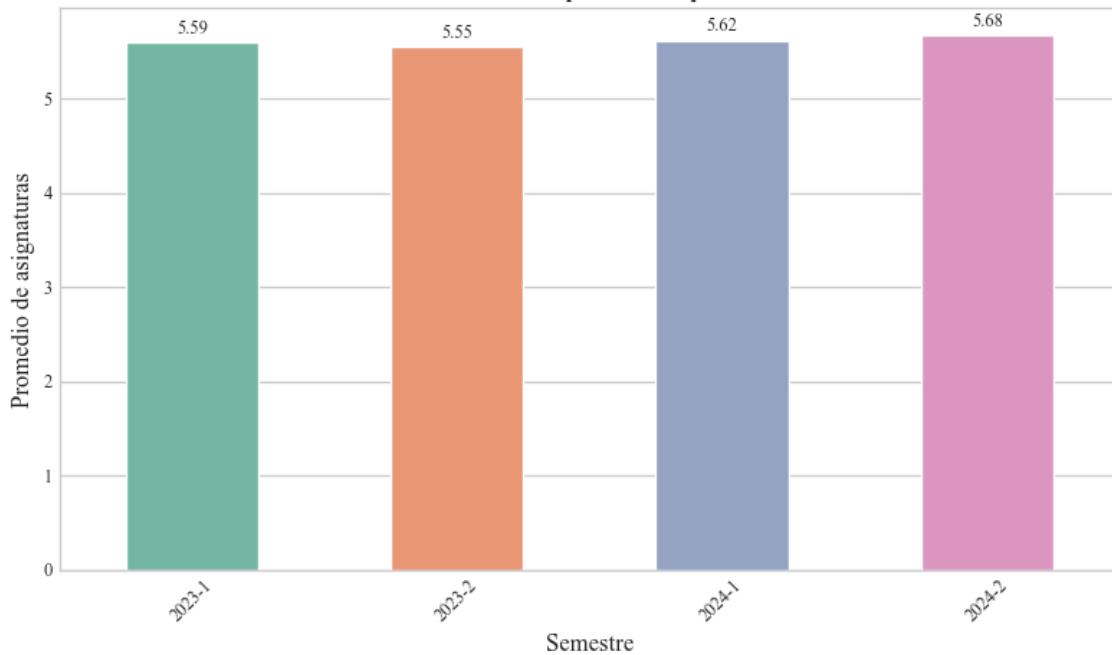


*Fuente: Elaboración personal*

**Figura 6**

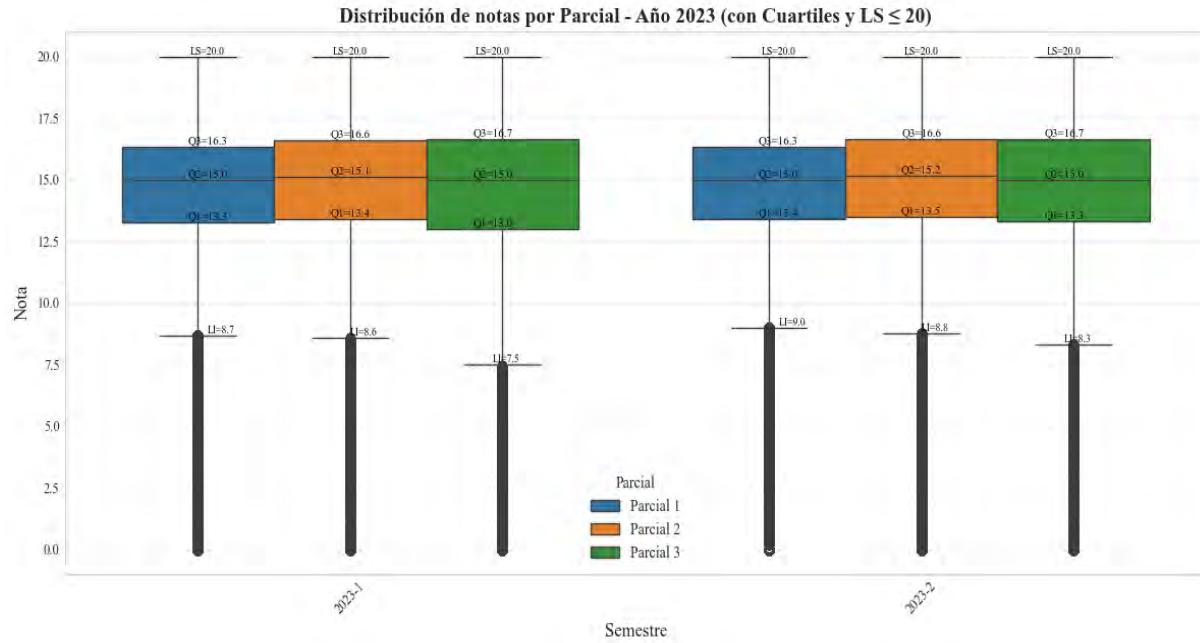
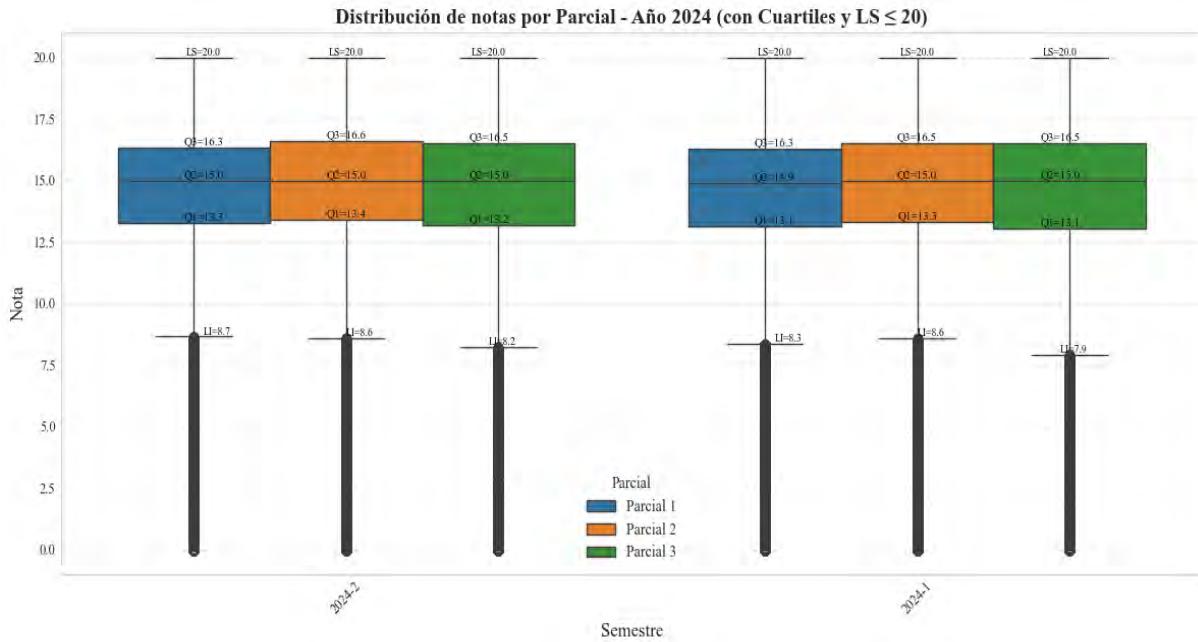
*Promedio de cursos matriculados por alumno del semestre 2023-I al 2024-II*

**Promedio de cursos por alumno por semestre**



*Fuente: Elaboración personal*

En la Figura 5 y Figura 6 se pudo observar una distribución uniforme con respecto a los alumnos matriculados por semestre y en el promedio de cursos por alumno por semestre, lo que mostraría que es constante la cantidad de alumnos que se matriculan al mes, aproximadamente 18437 alumnos; y además la cantidad promedio de cursos a los que se matriculan es de aproximadamente a 5.61 cursos. Lo que nos da un indicio que no hay deserción masiva de alumnos con respecto al tiempo, sin embargo, no nos asegura que no haya una cantidad considerable que se encuentre en riesgo. Por lo tanto, se presenta en los siguientes gráficos de cajas Figura 7 y Figura 8 cómo se comportan los registros de notas con respecto a los semestres:

**Figura 7***Distribución de notas por parcial del año 2023**Fuente: Elaboración Personal***Figura 8***Distribución de notas por parcial del año 2024**Fuente: Elaboración Personal*

Los diagramas de cajas constituyen una herramienta estadística robusta para la representación gráfica de distribuciones de datos, fundamentada en el análisis de cuartiles y la detección de valores atípicos mediante la regla de Tukey (Tukey, 1977). Esta representación gráfica utiliza cinco estadísticos descriptivos principales: el primer cuartil ( $Q_1$ ), la mediana ( $Q_2$ ), el tercer cuartil ( $Q_3$ ), y los límites inferior y superior calculados como  $L_i = Q_1 - 1.5 \times IQR$  y  $L_s = Q_3 + 1.5 \times IQR$ , respectivamente, donde  $IQR$  representa el rango intercuartílico ( $Q_3 - Q_1$ ). El factor 1.5 en la regla de Tukey tiene justificación teórica, ya que en distribuciones normales aproximadamente el 99.3% de los datos se encuentran dentro de estos límites, estableciendo un criterio estadísticamente sólido para la identificación de observaciones atípicas. En el contexto específico del análisis de calificaciones académicas, donde el dominio de datos se encuentra naturalmente acotado en el intervalo  $[0, 20]$  ya que representa el rango real de las notas, resulta metodológicamente apropiado aplicar una restricción al límite superior calculado mediante la función  $L_s = \min(Q_3 + 1.5 \times IQR, 20)$ . Esta modificación, denominada truncamiento, respeta las restricciones inherentes del dominio de estudio, manteniendo la validez estadística del método mientras se adapta a las limitaciones reales del sistema de evaluación. El análisis estadístico aplicado a la distribución de calificaciones mediante esta metodología revela una clasificación sistemática donde los valores típicos (inliers) corresponden a calificaciones dentro del intervalo  $[L_i, L_s]$ , siendo  $L_i$  con rango  $[7.495, 9.005]$  y  $L_s$  truncado sistemáticamente en 20, mientras que los valores atípicos (outliers) representan calificaciones inferiores al límite inferior, identificando estudiantes con rendimiento académico deficiente. Los resultados evidencian que los cuartiles  $Q_1 \in [13.06, 13.5]$ ,  $Q_2 \in [14.9, 15.17]$  y  $Q_3 \in [16.3, 16.67]$  configuran una distribución con asimetría negativa moderada, donde la cola izquierda contiene sistemáticamente entre 5.46% y 8.37% de

observaciones atípicas para parciales 1 y 2, y entre 4.90% y 5.09% para el parcial 3, manteniendo consistencia temporal en el período 2023-2024.

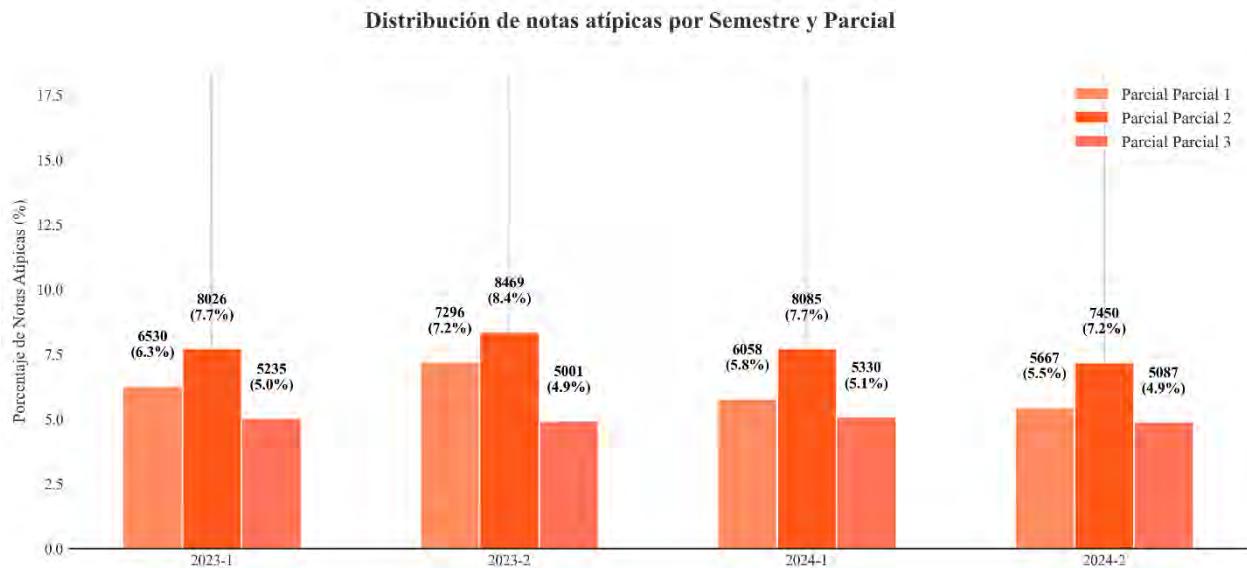
Esta configuración matemática demuestra que los valores atípicos capturan efectivamente el segmento poblacional con calificaciones reprobatorias ( $< 8.25$ ), considerablemente inferior a la nota mínima aprobatoria típica del sistema educativo universitario peruano (13-13.5), confirmando que aproximadamente 6-8% de la población estudiantil presenta un patrón estructural de bajo rendimiento académico, validando estadísticamente una proporción significativa de estudiantes en riesgo académico identificados mediante la metodología de detección de valores atípicos basada en cuartiles.

El análisis estadístico de la distribución de valores atípicos por semestre y parcial (Figura 9) evidencia que el Parcial 2 hay la mayor proporción de notas atípicas en todos los semestres analizados, con porcentajes que oscilan entre 7.2% y 8.4%, estableciendo un patrón estructural de bajo rendimiento en esta evaluación específica, contrastando con el Parcial 1 que mantiene valores relativamente estables entre 5.8% y 6.3%, y el Parcial 3 que exhibe los porcentajes más bajos entre 4.9% y 5.4%. La diferencia sistemática entre parciales, donde el Parcial 2 presenta aproximadamente 1.5-2.5 puntos porcentuales más de valores atípicos que el Parcial 1 y 2.5-3.5 puntos porcentuales más que el Parcial 3, resulta estadísticamente significativa considerando el tamaño muestral superior a 100,000 observaciones por parcial, sugiriendo que factores específicos del proceso evaluativo influyen en el rendimiento académico. Esta concentración de bajo rendimiento en el Parcial 2 puede atribuirse a factores temporales, ya que coincide con la mitad del semestre académico cuando se evalúan contenidos nuevos y de mayor complejidad con respecto al anterior Parcial, factores de acumulación de deficiencias académicas donde estudiantes con dificultades en el Parcial 1 tienden a mantener o agudizar sus problemas, y factores de carga

académica relacionados con la intensificación de actividades en el período medio semestral. El análisis longitudinal muestra una tendencia descendente en el porcentaje de valores atípicos del Parcial 2 desde 2023-1 (7.7%) hasta 2024-2 (7.2%); la persistencia del patrón donde el Parcial 2 mantiene sistemáticamente los valores más altos confirma que este constituye el punto crítico del rendimiento académico estudiantil, requiriendo la creación de estrategias de intervención focalizadas en el período medio del semestre, por ejemplo: realizar las tutorías especializadas a mitad de semestre en vez de final de semestre, reforzar académicamente e identificar de manera temprana a los estudiantes en riesgo antes de la culminación del segundo parcial antes de la segunda evaluación parcial.

**Figura 7**

*Distribución de notas atípicas por Semestre y Parcial*



*Fuente: Elaboración Personal*

### 3.4. Transformación de Datos

Partimos de un conjunto de datos tabulares de registro de notas en formato de dataframe, donde cada fila representa un registro de calificaciones individuales por estudiante, asignatura, parcial y semestre. La estructura original se puede representar como un conjunto de tuplas:

$$D = \{(a_i, c_j, s_k, t_l, n_{ijkl})\}$$

Donde:

- $a_i$ : identificador del alumno  $i$ ;
- $c_j$ : código de la asignatura  $j$ ;
- $s_k$ : semestre académico  $k$ ;
- $t_l$ : tipo de evaluación (Parcial 1, Parcial 2, Parcial 3, Sustitutorio);
- $n_{ijkl}$ : nota obtenida por el alumno  $a_i$  en la evaluación  $t_l$  de la asignatura  $c_j$  durante el semestre  $s_k$ .

Mediante la función “melt” de aplanamiento, se convierte el conjunto de datos en una estructura más uniforme y, donde la columna “Tipo\_Nota” actúa como una categoría, y la columna “Nota” representa su valor correspondiente. Esta técnica es conocida como “normalización de datos en formato largo” o tidy data, donde cada observación ocupa una fila.

Posteriormente, se crea una nueva etiqueta compuesta que concatena la información relevante:

$$\text{Etiqueta_Columna} = c_j + \text{str}(" - ") + \text{map}(tl) + \text{str}(" - ") + s_k$$

- $str("")$ : convierte cualquier valor a su representación como texto;
- $map(tl)$ : aplica la función de aplanamiento a cada  $tl$  perteneciente a un  $c_j$ .

Con esta clave, se realiza una transformación tipo “pivot table”, que reestructura el conjunto de datos en formato ancho, generando una matriz:

$$X \in R^{n \times m}$$

- $n$ : número de alumnos;
- $m$ : número total de columnas que representan evaluaciones distintas;
- $X$ : nota del estudiante  $i$  en la evaluación  $l$ , o vacío si: no la rindió, no se matriculó en el curso, el curso solo tenía dos exámenes parciales.

Obteniendo columnas compuestas para cada alumno, como, por ejemplo:

- AD103ZOS\_2\_2023-2
- AD110AAE\_1\_2023-1
- AD110AAE\_1\_2023-2

Esta transformación tiene como objetivo identificar de manera única cada instancia de evaluación académica; en estas columnas se almacena la información correspondiente a una asignatura en un examen parcial o sustitutorio determinado de un semestre, esta transformación ayuda ya que todas las notas pertenecientes a un alumno se pueden almacenar en un registro, lo que evita duplicados en los identificadores de los alumnos en cada fila y prepara los datos para la aplicación posterior de algoritmos de emparejamiento: ELO, Glicko-2 y TrueSkill.

### 3.5. Cálculo de GPGA de los Estudiantes

El CGPA (Cumulative Grade Point Average) es una medida agregada que busca representar de forma equitativa el rendimiento académico de un estudiante, en esta investigación se realiza a lo largo de diferentes cursos y períodos académicos. En el contexto de análisis de datos educativos, calcular el CGPA de forma precisa y representativa es esencial para aplicar modelos de clasificación como ELO, Glicko-2 y TrueSkill, ya que estos se alimentan de comparaciones fundamentadas en métricas de rendimiento.

A primera instancia, podría parecer conveniente calcular el CGPA de los estudiantes como un simple promedio de sus calificaciones. Esta aproximación resulta atractiva debido a la eficiencia computacional de la agregación plana, que se basa en una única operación de agrupación con una complejidad de  $O(n \log n)$  siendo  $n$  el número total de registros en la base de datos de notas.

$$\text{CGPA}_i^{plano} = \frac{1}{N_i} \sum_{e=1}^{N_i} n_{i,e}$$

Donde  $N_i$  es el total de evaluaciones registradas para el estudiante  $i$ , sin distinción de curso o semestre. Si bien el enfoque directo es visualmente simple en apariencia - una única agregación - este método no respeta la estructura pedagógica de los datos. Asignaturas con mayor número de evaluaciones influencian más el resultado final, lo cual genera un sesgo ponderativo no intencionado. Además, semestres con más cursos o cursos con más evaluaciones promediarían de forma desproporcionada sobre el CGPA. Este comportamiento de la función introduce un sesgo estructural dentro del análisis académico lo que generaría una mala representación del rendimiento estudiantil. Si se busca una interpretación académica coherente, resulta fundamental considerar que el sistema educativo valora el desempeño por asignatura y no

en función del número de exámenes realizados; por lo tanto, el anterior enfoque no daría buenos resultados.

En el presente estudio se ha optado por calcular el CGPA (Cumulative Grade Point Average) de manera jerárquica, siguiendo la estructura académica natural del sistema educativo: primero se calcula el promedio de notas a nivel de evaluación dentro de cada asignatura; luego, se promedian estas notas a nivel de curso; posteriormente, se obtiene un promedio semestral; y finalmente, se calcula el CGPA final como el promedio de estos promedios. Esta metodología se compara favorablemente frente a la anterior estrategia ya que conserva la misma complejidad  $O(n \log n)$  para cada fase de agregación y además de tener menos sesgo considerable.

Desde un punto de vista matemático, el cálculo jerárquico para cálculo del CGPA final se formula de la siguiente manera:

Sea  $n_{i,c,s,e}$  la nota obtenida por el estudiante  $i$  en la evaluación  $e$  del curso  $c$ , en el semestre  $s$ . El proceso de agregación lo definimos jerárquicamente como:

### **3.5.1. Promedio por curso (asignatura)**

$$\bar{n}_{i,c,s} = \frac{1}{E_{c,s}} \sum_{e=1}^{E_{c,s}} n_{i,c,s,e}$$

Donde  $E_{c,s}$  es el número de evaluaciones realizadas por el estudiante en el curso  $c$ , en el semestre  $s$ .

### **3.5.2. Promedio semestral**

$$CGPA_{i,s} = \frac{1}{C_s} \sum_{c=1}^{C_s} \bar{n}_{i,c,s}$$

Donde  $C_s$  representa el número de asignaturas que el estudiante llevó en el semestre  $s$ .

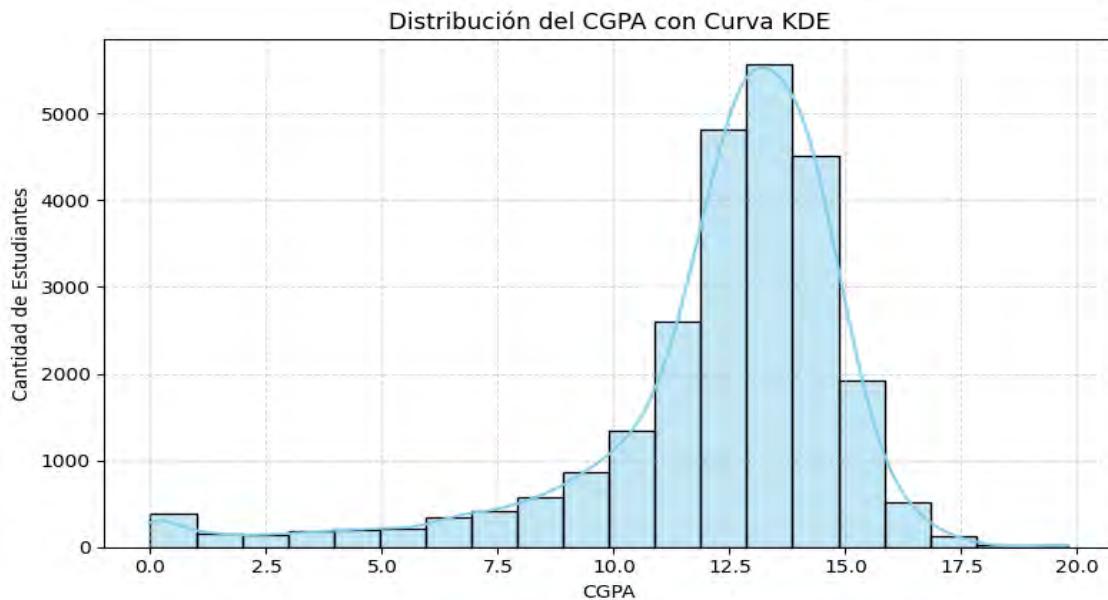
### 3.5.3. CGPA final

$$CGPA_{i,final} = \frac{1}{C_s} \sum_{c=1}^{S_i} CGPA_{i,s}$$

Donde  $S_i$  número de semestres cursados por el estudiante  $i$ . El cálculo jerárquico evita el problema del sesgo por densidad de datos, lo que garantiza que cada asignatura tenga el mismo peso, independientemente de su número de evaluaciones, y asegura que cada semestre contribuya equitativamente al promedio acumulado, independientemente de la carga académica. De este modo, se obtiene un indicador de rendimiento académico robusto, comparable entre estudiantes, y apto para medir los modelos de emparejamiento como ELO, Glicko-2 y TrueSkill, los cuales requieren métricas de rendimiento representativas y no sesgadas. El CGPA de cada estudiante en la Figura 10 muestra la siguiente distribución:

**Figura 8**

*Histograma de distribución CGPA de los estudiantes con Curva KDE*



Fuente: Elaboración Personal

El gráfico presenta un patrón unimodal y aproximadamente simétrico, con una concentración predominante de estudiantes en torno a la moda aproximada de 12.5 CGPA. Este pico pronunciado sugiere que la mayoría de los estudiantes (más del 60%) se agrupa en un rango relativamente estrecho, específicamente entre 10.0 y 15.0 CGPA. La ligera asimetría negativa (cola izquierda más extendida) indica una mayor frecuencia de estudiantes con rendimiento inferior a la moda en comparación con aquellos cuyo rendimiento es superior. Esta característica se evidencia en la suave prolongación hacia valores bajos ( $\text{CGPA} < 7.5$ ), mientras que la cola derecha decrece abruptamente después de 15.0 CGPA, con muy pocos casos cercanos a 17.5 CGPA y ninguno próximo al límite superior teórico (20.0).

La distribución del gráfico sugiere un sistema de evaluación consistente que tiende a agrupar a los estudiantes en niveles intermedios. La escasez de valores extremos es significativa: los casos con  $\text{CGPA} < 5.0$  son marginales, lo que podría indicar que se necesitan implementar mecanismos de apoyo para estudiantes en riesgo, y la ausencia de notas cercanas a 20.0 también nos indica posibles barreras estructurales, como podría ser; un techo en la calificación que limita el rendimiento excepcional.

La curva KDE suaviza la distribución real, lo que puede atenuar irregularidades en los extremos, lo que nos indicaría a primera vista pudiese tratarse de una distribución normal, debido que la cantidad de datos es mucho mayor a 50 descartamos la prueba de Shapiro-Wilk y aplicamos una prueba de Kolmogórov-Smirnov que maneja adecuadamente los datos densos, esta prueba compara la función de distribución acumulada empírica de los datos con la función de una distribución normal teórica ajustada a la media y desviación estándar de la muestra. Se considera para esta prueba un nivel de significancia estándar de  $\alpha = 0.05$ , correspondiente a un 95% de nivel de confianza; obteniendo un resultado  $D=0.1589$  y un valor  $p < 0.0001$ . Dado que el valor  $p$  es

menor al nivel de significancia establecido, se rechaza la hipótesis nula que plantea que los datos provienen de una distribución normal. Por lo tanto, se concluye que la distribución del CGPA no sigue una distribución normal, lo cual indica una posible presencia de asimetría, curtosis o valores atípicos dentro del conjunto de datos analizado. Ya que no se trata de una distribución normal esto es de utilidad ya que CGPA se define en esta investigación como métrica de validación que establecerá correlación con los algoritmos de emparejamiento ELO, Glicko-2 y TrueSkill.

### **3.6. Clasificación de los estudiantes utilizando el algoritmo ELO.**

Se implementa una versión adaptada del sistema de clasificación ELO, originalmente desarrollado para evaluar la habilidad de jugadores en juegos de competencia como el ajedrez, aplicado aquí al desempeño académico de estudiantes. La lógica del sistema se fundamenta en asignar y actualizar un puntaje MMR ELO a cada estudiante en función de sus calificaciones obtenidas por curso, consideradas como resultado de "enfrentamientos" implícitos entre pares.

Este enfoque permite clasificar las calificaciones académicas en un sistema dinámico de puntuación basado en competencia relativa, utilizando el algoritmo de emparejamiento ELO. A diferencia del promedio simple, el algoritmo ELO en esta investigación considera el contexto competitivo de cada curso: cantidad de estudiantes y notas por curso por semestre, proporcionó una mejor clasificación del rendimiento estudiantil en distintos escenarios y la identificación de estudiantes con bajo rendimiento en este entorno.

Primero, para cada estudiante  $i$  en un curso válido específico  $c$  perteneciente en a un semestre determinado, para que el curso se considera válido al menos debe haber más de 5 estudiantes sino la clasificación por ELO tendería a fallar por el número limitado de emparejamientos; se calcula una nota total  $NT_i^c$  considerando:

- Las notas parciales válidas  $P_1, P_2, P_3$  para cursos de 4 créditos o  $P_1, P_2$  para cursos de 3 o 2 créditos

Dado un examen sustitutorio S reemplaza exclusivamente al examen parcial (EP) más bajo del conjunto de notas parciales válidas, definida como  $P_{\min} = \min(P)$ . El promedio final corregido NT se calcularía idealmente como  $NT_i^c = \frac{(\sum_{k=1}^n P_k) - P_{\min} + S}{n} = \bar{P} + \frac{S - P_{\min}}{n}$  donde  $\bar{P}$  es el promedio inicial. El problema fundamental es la indeterminación de  $P_{\min}$  a nivel de registro académico agregado, ya que solo se dispone del promedio final oficial, no de las notas desagregadas por ítem o parcial. Por lo tanto, se debe construir un estimador  $\widehat{NT}$  que aproxime  $\widehat{NT}$  utilizando únicamente las variables observables:  $(\bar{P}, S, n)$ . Por lo que se propone un modelo de ajuste lineal de la forma  $\widehat{NT} = \bar{P} + \alpha \cdot (S - \bar{P})$  donde  $\alpha$  es un coeficiente de impacto a estimar. Este modelo supone que la ganancia promedio por aplicar el sustitutorio es proporcional a la diferencia entre  $S$  y el promedio original. Para relacionar  $\alpha$  con el modelo ideal, se toma el valor esperado de la ganancia real  $(S - P_{\min})/n$ . Asumiendo que  $P_{\min}$  es una variable aleatoria que depende de la distribución de  $P$ , y que  $S$  es independiente de  $P_{\min}$  una vez fijado, se tiene el valor esperado  $E\left[\frac{S - P_{\min}}{n}\right] = \frac{S - E[P_{\min}]}{n}$  bajo el supuesto de que las notas parciales se distribuyen de manera aproximadamente uniforme en un rango acotado, la esperanza del mínimo  $E[P_{\min}]$  puede aproximarse como  $\bar{P} - \Delta$ , donde  $\Delta$  es una desviación típica positiva. Así,  $E\left[\frac{S - P_{\min}}{n}\right] \approx \frac{S - (\bar{P} - \Delta)}{n} = \frac{S - \bar{P}}{n} - \frac{\Delta}{n}$ . Comparando con el modelo lineal  $\alpha \cdot (S - \bar{P})$  se infiere que un candidato para  $\alpha$  es  $1/n$ . Sin embargo, esta elección ignora el término  $\Delta/n$  y sobreestima el impacto cuando  $S$  es cercano a  $\bar{P}$ . Por tanto, se introduce un coeficiente de atenuación empírico  $\beta$ , definiendo:  $\alpha = \beta/n$  con  $0 < \beta < 1$ . El valor  $\beta = 0.9$  (es decir,  $\alpha = 0.9/n$ ) ajustaría parcialmente la sobreestimación. Para  $n = 3, \alpha \approx 0.3$ ; para  $n = 2, \alpha = 0.45$ . Dado que:

- La mayoría de cursos en el sistema analizado son de 4 créditos (n=3).
- Se busca un único coeficiente simplificado para homogenizar el cálculo en todos los cursos.
- Es preferible subestimar ligeramente el beneficio del sustitutorio a sobreestimarlo, para no inflar artificialmente los ratings en el algoritmo ELO.

Se elige el valor  $\alpha = 0.3$  como valor óptimo y conservador. Este valor equivale a  $\beta = 0.9$  para  $n = 3$  y a  $\beta = 0.6$  para  $n = 2$ , lo que significa que, para cursos con menos parciales, el modelo es aún más conservador, compensando la mayor incertidumbre en la estimación de  $P_{\min}$ .

En conclusión, la siguiente fórmula operativa es un estimador consistente y robusto del promedio final verdadero que incluye el sustitutorio. Minimiza el error cuadrático esperado bajo incertidumbre sobre  $P_{\min}$  y garantiza la equidad en la comparativa competitiva entre estudiantes de diferentes cursos (Amor Pulido et al., 2009):

$$NT_i^c = \frac{\sum_{k=1}^n P_k + 0.3S}{n}$$

Donde  $n$  es el número de parciales válidos, después inicializamos a cada estudiante  $i$  con un MMR ELO inicial:

$$R_i = R_0 \text{ donde } R_0 = 1000$$

Después se simulan los enfrentamientos académicos por pares de estudiantes  $i$  y  $j$ , comparando las notas calculadas por cada curso  $NT_i^c$  y  $NT_j^c$  respectivamente donde si  $NT_i^c > NT_j^c$ , se considera que el estudiante  $i$  ganó o si  $NT_i^c = NT_j^c$  se considera un empate.

Ahora calculamos la puntuación esperada hallando probabilidad esperada de victoria de un estudiante  $i$  sobre un estudiante  $j$ , según el sistema ELO se calcula como

$$E_i = \frac{1}{1 + 10^{(R_j - R_i)/400}}$$

Luego del enfrentamiento se actualizan los puntajes de acuerdo a la siguiente fórmula:

$$R_i^{Nuevo} = R_i + K(M_i - E_i)$$

Donde:

- $K$  es el coeficiente de sensibilidad del sistema ELO. Este parámetro controla el ritmo al cual los puntajes se ajustan: un valor alto hace el sistema más reactivo a nuevas actuaciones, mientras que uno bajo favorece la estabilidad del puntaje. Establecemos  $K=32$  ya que es común implementaciones estándar del algoritmo ELO y suficiente para simulaciones educativas. Este valor podría cambiarse si no se encuentran los resultados esperados
- $M_i \in \{1, 0.5, 0\}$  es el resultado observado del estudiante: 1 si gana, 0.5 si empata y si pierde 0.

Esto se repite para todos los pares de estudiantes  $i$  y  $j$  en un curso específico  $c$  perteneciente en a un semestre determinado. Una vez calculados los ELO individuales por curso, se consolida un ELO promedio por estudiante considerando todas sus participaciones:

$$ELO_m^{Promedio} = \frac{\sum_{c=1}^m R_i^c}{m}$$

Donde  $m$  es el número de cursos en el que el estudiante  $i$  se matriculó;  $R_i^c$  es el ELO final obtenido en cada curso. Después de ello asignamos las estrellas con respecto a los percentiles  $P(x)$

correspondientes al valor del ELO promedio  $x$  encontrado para cada estudiante en su semestre determinado como se observa en la Tabla 3:

**Tabla 3**

*Percentiles por rendimiento para asignación de estrellas ELO*

Condición del Percentil ELO	Estrellas (Stars)	Rendimiento
$P(x) \geq 90\%$	☆☆☆☆☆	Excelente
$75\% \leq P(x) < 90\%$	☆☆☆☆	Alto
$50\% \leq P(x) < 75\%$	☆☆☆	Promedio
$25\% \leq P(x) < 50\%$	☆☆	Malo
$P(x) < 25\%$	☆	Bajo

*Fuente: Elaboración personal*

- $x \in \mathbb{R}$ : Puntaje ELO promedio del estudiante en un semestre;
- $P(x) \in [0\%, 100\%]$  : Percentil de  $x$  dentro del conjunto de datos.

Para identificar a los alumnos con bajo rendimiento académico mediante el algoritmo de clasificación tipo ELO y la asignación de estrellas según percentiles, se considera a aquellos ubicados en el cuartil inferior (por debajo del percentil 25%). Es decir, estudiantes que solo tienen una sola estrella (☆). Estos alumnos no solo presentan calificaciones bajas, sino que también han sido muy superados en múltiples enfrentamientos académicos simulados por sus compañeros.

Ya definido el cálculo del MMR o puntaje ELO y el etiquetado de Estrellas para cada alumno por percentiles en un semestre determinado: evaluamos el tiempo de ejecución, el número de estudiantes y el número de cursos del algoritmo de emparejamiento ELO por semestre, aplicado alumnos post pandemia de la Universidad de San Antonio Abad del Cusco. Se realizó la ejecución

secuencial del mismo sobre los datos académicos correspondientes a cuatro semestres: 2023-1, 2023-2, 2024-1 y 2024-2. Sea:

- $s \in \{s_1, s_2, s_3, s_4\}$ : El conjunto de semestres donde  $s_1 = 2023 - 1$ ,
- $s_2 = 2023 - 2$ ,  $s_3 = 2024 - 1$  y  $s_4 = 2024 - 2$ ;
- $\forall s \exists D_n / D_n$  es el dataframe correspondiente a cada semestre  $s_n$

El algoritmo ELO se aplica de forma independiente a cada conjunto  $D_n$ , generando para cada alumno un rating de emparejamiento  $R_i^{(n)}$ , y asignando una etiqueta de estrellas basada en percentiles del ranking dentro del semestre  $s_n$ . Formalmente, para cada alumno  $i$  en el semestre  $s_n$ , se calcula:

$$R_i^{(n)} = ELO(D_n, i)$$

Donde la función  $ELO(\cdot)$  representa el proceso iterativo de actualización de ratings basado en los resultados académicos registrados en  $D_n$ . Luego, el conjunto de ratings  $R = \{R_i^{(n)} / n = 1, 2, 3, 4\}$  se ordena y se divide en percentiles para asignar categorías de estrellas:  $\star$ ,  $\star\star$ ,  $\star\star\star$ ,  $\star\star\star\star$ ,  $\star\star\star\star\star$ ; como anteriormente se definió. Se muestran a continuación los resultados obtenidos en tiempo de ejecución:

#### **Tabla 4**

*Resultados obtenidos después de la ejecución del algoritmo ELO en los semestres 2023-I, 2023-II, 2024-I y 2024-II.*

Semestre	Tiempo (s)	Estudiantes	Cursos
2023-1	1787.21	18540	3556
2023-2	1755.99	18204	3541
2024-1	1845.26	18607	3669
2024-2	2091.27	18221	3715

*Fuente: Elaboración Personal*

En primer lugar, se detectaron aproximadamente entre 9,200 y 9,900 columnas asociadas a cursos en cada uno de los DataFrames semestrales analizados; el elevado número de columnas refleja un gran volumen dimensional de información académica, así como amplitud del registro de datos disponibles para cada periodo.

Se identificaron entre 3,500 y 3,700 cursos por semestre, lo cual garantiza una muestra representativa y estadísticamente sólida para aplicar el algoritmo de emparejamiento. Las velocidades promedio de procesamiento fueron constantes y se situaron entre 2.7 y 3.3 cursos por segundo, lo que refleja la efectividad de la implementación optimizada del algoritmo ELO. Este rendimiento permitió ejecutar los cálculos en un tiempo razonable, considerando el gran volumen de datos involucrados.

Los tiempos totales de ejecución variaron entre aproximadamente 1,755 y 2,090 segundos por semestre, es decir, entre 29 y 35 minutos. Esta duración resulta adecuada y manejable dentro de un contexto de análisis masivo de datos académicos. Se observa que este desempeño se logró sin comprometer la precisión ni la integridad del análisis.

Finalmente, la cantidad de estudiantes únicos evaluados por semestre se mantuvo en torno a los 18,000, y el total de registros generados por el algoritmo ELO superó las 100,000 entradas en cada periodo. Se evidencia, por lo tanto, la robustez del procedimiento aplicado y la profundidad del análisis realizado sobre el rendimiento académico de los estudiantes en la etapa post pandemia. Por último, se almacenó los resultados en un documento Excel llamado Alumnos\_Clasificados\_ELO.xlsx que guarda los resultados de la clasificación Elo para poderla utilizar más adelante.

### **3.7. Identificación de los alumnos con bajo rendimiento después de la clasificación por el algoritmo ELO**

Lo que se busca generar una clasificación cualitativa del rendimiento académico de los estudiantes mediante un sistema de estrellas (de 1 a 5), basado en sus respectivos valores de MMR o puntaje ELO general de cada semestre. Para ello, se sigue un proceso estructurado que puede dividirse en tres fases: cálculo del rating ELO general ponderado, asignación de estrellas con una distribución empírica, y cálculo del percentil de cada estudiante.

Primero calculamos el ELO general ponderado de cada alumno  $i$ : sea  $R_i^{(n)}$  el valor de ELO promedio obtenido por dicho estudiante durante un semestre  $n$ . Se define el peso de competencia  $\omega_i$  para el estudiante  $i$  como:

$$\omega_i = T_i \cdot C_i$$

Donde:

- $T_i$  : Número total de estudiantes contra los que compitió;
- $C_i$  : Número de cursos tomados por el estudiante.

Calculamos el ELO ponderado individual como:

$$RP_i^{(n)} = R_i^{(n)} \cdot \omega_i$$

Agrupamos a los alumnos  $i$ , se obtiene el ELO General Ponderado para cada estudiante como:

$$ELO\_GP_i = \frac{\sum_1^n RP_i^{(n)}}{\sum_1^n \omega_i}$$

Para asignar una etiqueta cualitativa de rendimiento, se utiliza el valor ponderado  $ELO\_GP_i$  del estudiante. El algoritmo puede analizarse mediante múltiples esquemas de distribución, para nuestro caso del estudio se emplea una distribución académica asimétrica (García Tárrago, 2020); sean los percentiles de corte:  $PC = \{5\%, 25\%, 75\%, 95\%\}$  ya que son más cercanos a la realidad, propia y adecuada de utilizar para entornos educativos. Los percentiles de corte son los siguientes:

**Tabla 5**

*Percentiles por rendimiento general para asignación de estrellas ELO*

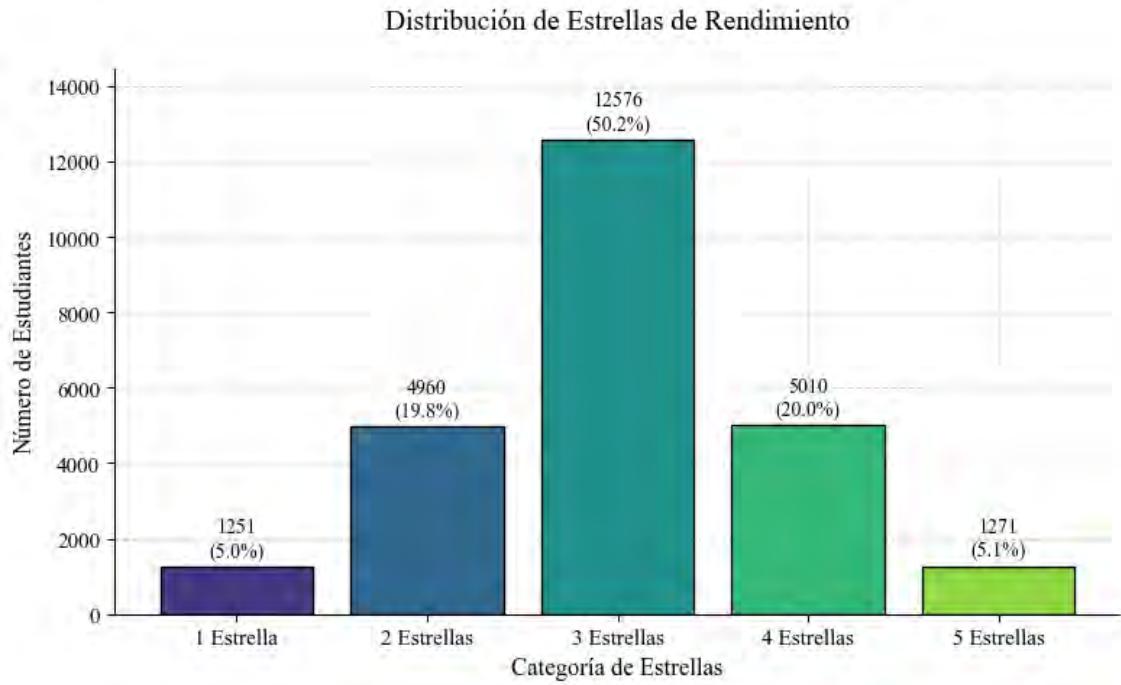
Condición del Percentil ELO	Estrellas Generales	Rendimiento General
$ELO\_GP_i \geq 95\%$	★★★★★	Excelente
$75\% \leq ELO\_GP_i < 95\%$	★★★★	Alto
$25\% \leq ELO\_GP_i < 75\%$	★★★	Promedio
$5\% \leq ELO\_GP_i < 25\%$	★★	Malo
$ELO\_GP_i < 5\%$	★	Bajo

Fuente: Elaboración personal

Después de aplicar la distribución Figura 11 se puede observar los siguientes resultados.

**Figura 9**

*Distribución y clasificación por Estrellas de ELO Ponderado General*



*Fuente: Elaboración personal*

Se puede observar la distribución de estudiantes según la categoría de estrellas de rendimiento académico asignadas mediante el algoritmo ELO. Este sistema de clasificación ha sido correctamente adaptado en esta investigación para evaluar y agrupar a los estudiantes en función de su desempeño académico, considerando el valor ponderado de su rating ELO acumulado a lo largo de varios semestres.

Del total de estudiantes evaluados, se observa que la mayor proporción se concentra en la categoría de 3 Estrellas, con 12,576 estudiantes, lo que representa aproximadamente el 50.2% de la muestra. Esta categoría constituye el núcleo del rendimiento promedio, indicando que la mitad de los estudiantes mantiene un desempeño académico intermedio de acuerdo con el sistema de

puntuación empleado. Esta tendencia sugiere una adecuada calibración del algoritmo, ya que agrupa de manera eficiente a la mayoría de los estudiantes en torno a una media de rendimiento.

En las categorías adyacentes, se identifican 4,960 estudiantes con 2 Estrellas (19.8%) y 5,010 estudiantes con 4 Estrellas (20.0%). Estos valores, relativamente simétricos respecto a la categoría central, refuerzan la hipótesis de una distribución en forma de campana o distribución normal, lo cual es deseable en sistemas de clasificación que buscan distinguir distintos niveles de desempeño con equidad y balance.

Por otro lado, los extremos de la distribución presentan una cantidad significativamente menor de estudiantes. La categoría de 1 Estrella agrupa a 1,251 estudiantes (5.0%), mientras que la de 5 Estrellas incluye a 1,271 estudiantes (5.1%). Estas proporciones reducidas en los niveles más bajos y más altos son coherentes con un enfoque meritocrático, donde solo un porcentaje reducido alcanza los extremos del rendimiento académico. En particular, la baja proporción de estudiantes en la categoría de 1 Estrella puede interpretarse como un indicador positivo, ya que evidencia que un número limitado de estudiantes se encuentra en los niveles más críticos de rendimiento, sin embargo, este es nuestro objetivo de investigación está el de encontrar a alumnos con bajo rendimiento utilizando el algoritmo ELO.

Al final almacenamos los resultados de la clasificación por el algoritmo de emparejamiento ELO en un archivo llamado Excel: “Alumnos bajo rendimiento ELO UNSAAC”, en este archivo adjuntaremos tanto los resultados de la clasificación ELO de aquellos alumnos con bajo rendimiento (Deficiente) en el percentil más bajo  $ELO\_GP_i < 5\%$  con una Estrella ★ como etiqueta, así como sus registros de notas consultados de la base de datos original.

### **3.8. Correlación entre ELO General Ponderado y CGPA Final.**

Con el objetivo de validar la efectividad del algoritmo de clasificación ELO aplicado al rendimiento académico, se realizó un análisis de correlación entre la variable ELO General Ponderado, obtenida mediante simulaciones de competencias académicas entre estudiantes, y el CGPA Final (Cumulative Grade Point Average), una métrica tradicionalmente utilizada para evaluar el desempeño académico acumulado.

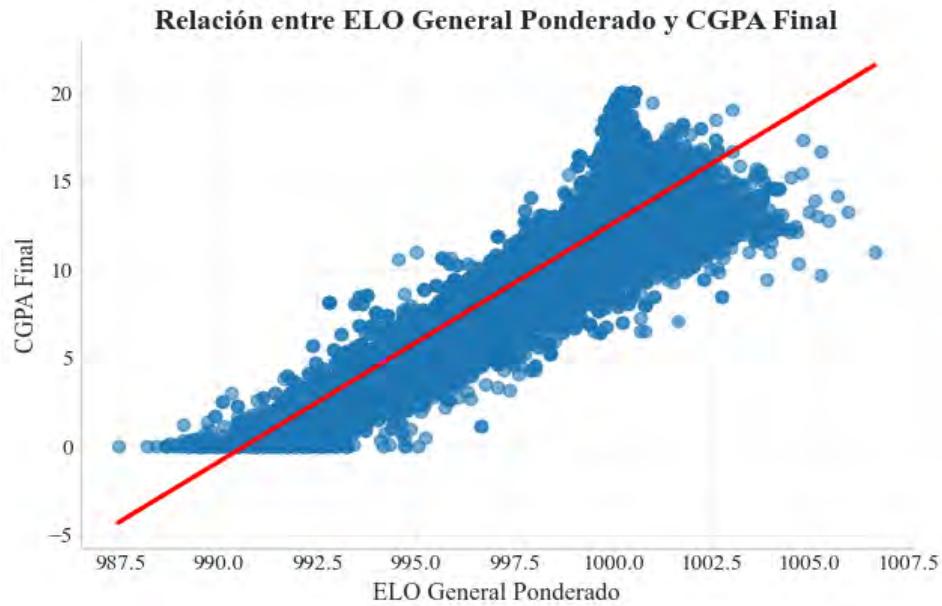
El análisis estadístico incluyó tanto el coeficiente de correlación de Pearson ( $r$ ) como el de Spearman ( $\rho$ ). El coeficiente de Pearson, es aquel que evalúa la relación lineal entre dos variables cuantitativas, arrojó un valor de  $r = 0.7970$  con un  $p$ -valor = 0.0000, lo que indica una correlación lineal fuerte y positiva, estadísticamente significativa. Este resultado evidencia que los estudiantes con un mayor puntaje ELO tienden también a presentar un mayor promedio acumulado, lo cual sugiere una coherencia sustantiva entre ambas métricas de evaluación.

Por otro lado, el coeficiente de Spearman, es la que mide la fuerza y dirección de una relación monótona entre los rangos de dos variables, resultó en  $\rho = 0.4630$  con un  $p$ -valor = 0.0000. Aunque esta correlación es más moderada que la de Pearson, también es positiva y significativa, lo que implica que, en términos generales, el orden relativo de los estudiantes según su clasificación por ELO guarda cierta correspondencia con su orden por CGPA.

Estos hallazgos se complementan con la Figura 12 donde se observa una clara tendencia ascendente entre los valores de ELO y CGPA, con una línea de regresión en la que se observa la relación positiva identificada en los coeficientes de correlación. La mayoría de los datos se agrupan alrededor de la línea de tendencia, confirmando la linealidad y consistencia de la relación.

**Figura 10**

*Correlación entre ELO General Ponderado y CGPA Final de los alumnos clasificados*



*Fuente: Elaboración personal*

En conclusión, del análisis del gráfico sugiere que el modelo de clasificación basado en el algoritmo ELO refleja con precisión el rendimiento académico de los estudiantes, alineándose de manera consistente con las calificaciones acumuladas. Por lo tanto, el ELO General Ponderado puede considerarse una variable válida y confiable para clasificar estudiantes a partir de sus resultados académicos, y reconocimiento de alumnos con bajo rendimiento.

### **3.9. Clasificación de los Estudiantes utilizando el Algoritmo Glicko-2.**

Se propone en esta investigación la implementación de una adaptación del sistema de clasificación Glicko-2, un modelo de estimación bayesiana dinámica originalmente concebido para la evaluación de habilidades en entornos de competencia, al ámbito del análisis del desempeño académico estudiantil. En esta adaptación, cada estudiante es caracterizado por un vector de estado compuesto por tres parámetros fundamentales: un puntaje de habilidad inicial ( $\mu$ ), que cuantifica

su nivel de rendimiento relativo; una desviación estándar de la habilidad (RD, por Rating Deviation), que mide la incertidumbre o confiabilidad estadística de dicha estimación; y un parámetro de volatilidad ( $\sigma$ ), que captura la tendencia inherente del desempeño del estudiante a fluctuar a lo largo del tiempo.

Este enfoque trifásico permite una evaluación diagnóstica multivariada que trasciende las métricas tradicionales. A nivel general, no solo identifica a estudiantes con rendimiento absoluto bajo, sino que también discrimina entre aquellos con desempeño inestable (alta volatilidad y/o RD) y aquellos con trayectorias emergentes o en mejora, todo ello con un nivel de confianza estadística cuantificable. La robustez del algoritmo ante datos incompletos —como los generados por estudiantes de reciente ingreso o por evaluaciones espaciadas— lo hace particularmente idóneo para entornos educativos con cohortes dinámicas y dispersión heterogénea en los resultados.

A diferencia de las métricas estáticas como el promedio simple, o incluso sistemas sofisticados como Elo, el algoritmo Glicko-2 incorpora explícitamente el contexto de cada actividad evaluativa. Este contexto incluye variables como el tamaño del grupo de referencia (número de estudiantes), la distribución de calificaciones y la frecuencia temporal de las evaluaciones, al tiempo que actualiza recursivamente el historial individual de cada estudiante. Esta capacidad para modelar la evolución temporal de la habilidad y su incertidumbre ha sido utilizada empíricamente en varios juegos de alto rendimiento y entornos competitivos con participantes con rendimiento variable (e.g., Counter-Strike: Global Offensive, Dota 2, Lichess, Chess.com). Su capacidad para ajustar la clasificación según las distintas asignaturas y la consistencia con respecto al tiempo lo hace ideal para semestres donde los estudiantes abandonan los cursos o tienen notas irregulares—como ocurre en muchos sistemas educativos.

Sea un conjunto de cursos  $C$ . Para cada curso  $c \in C$ , con  $n_c$  estudiantes  $i = 1, 2, \dots, n_c$  se dispone de dos a tres calificaciones parciales y un sustitutorio. El objetivo es estimar, para cada estudiante  $i$  y curso  $c$ , un parámetro latente de “desempeño relativo”  $\mu_{ic}$  en una escala comparativa (Glicko-2), junto con su incertidumbre  $\sigma_{ic}$  (en inglés “rating deviation”, RD) y su volatilidad  $\omega_{ic}$  (inestabilidad temporal), y luego recopilar estos resultados a nivel global del estudiante (promedios, medianas, ponderaciones, percentiles y “Estrellas”).

El modelo adopta la estructura de Glicko-2 (Glickman, 2022) para cada “periodo de valoración” (un curso por semestre), cada jugador (estudiante) enfrenta a todos los demás en un torneo round-robin o emparejamiento cíclico sintético donde los resultados de los matches o “partidos” derivan de las diferencias de notas. La actualización de  $(\mu, \sigma, \omega)$  sigue las ecuaciones de Glicko-2 en la escala interna.

Para cada estudiante  $i$  en curso  $c$ , se define una nota agregada  $N_{ic}$  como:

- Si el curso tiene tres parciales  $P_1, P_2$  y  $P_3$  y un sustitutorio  $S$  que producto con el coeficiente óptimo y conservador  $\alpha \approx 0.3$  ajustaría parcialmente la sobreestimación como se explicó en la implementación de ELO:

$$N_{ic} = \frac{P_1 + P_2 + P_3 + 0.3S}{3}$$

- Si el curso tiene dos parciales  $P_1$  y  $P_2$  y un sustitutorio  $S$ :

$$N_{ic} = \frac{P_1 + P_2 + 0.3S}{2}$$

Basado la nota agregada, calculamos los parámetros iniciales necesarios para el sistema Glicko-2, con un enfoque de inicialización adaptativa, para cada curso  $c$ : el rating inicial  $\mu_0^{(c)}$ , la incertidumbre inicial RD  $\sigma_0^{(c)}$  y volatilidad inicial  $\omega_0^{(c)}$ . El rating inicial lo calculamos como:

$$\bar{N}_c = \frac{1}{n_c} \sum_i N_{ic}$$

$$\mu_0^{(c)} = 100\bar{N}_c$$

La incertidumbre inicial RD y la volatilidad la calculamos como:

$$s_c = \sqrt{\frac{1}{n_c - 1} \sum_i (N_{ic} - \bar{N}_c)^2}$$

$$\sigma_0^{(c)} = \begin{cases} 200, & 30s_c < 200 \\ 30s_c, & 200 \leq 30s_c \leq 500 \\ 500, & 30s_c > 500 \end{cases}$$

$$\omega_0^{(c)} = \begin{cases} 0.1, & 0.005s_c < 0.1 \\ 0.005s_c, & 0.1 \leq 0.005s_c \leq 0.04 \\ 0.04, & 30s_c > 0.04 \end{cases}$$

$\sigma_0^{(c)}$  está acotada entre 200 y 500 y  $\omega_0^{(c)}$ , entre 0.04 y 0.1 respectivamente, lo que estabiliza numéricamente los parámetros en el algoritmo Glicko-2, especialmente en cursos con dispersión muy baja o muy alta.

Después aplicamos el reescalado a interno de Glicko-2: donde operamos con una escala externa  $(\mu, \sigma)$  y una escala interna  $(\bar{\mu}, \bar{\phi})$  donde  $\bar{\phi}$  es el RD interno con una constante logística base  $\kappa = 1/Q$ . En la formulación estándar de Glicko-2:

$$\bar{\mu} = \frac{\mu - 1500}{\kappa} , \bar{\phi} = \frac{\sigma}{\kappa} , \kappa = \frac{400}{\ln 10} \approx 173.7178$$

Aplicamos el reescalado por curso en la implementación de esta manera:

$$\bar{\mu} = \frac{\mu - \mu_0^{(c)}}{\kappa_c}, \bar{\phi} = \frac{\sigma}{\kappa_c}, \kappa_c = Q[\sigma_0^{(c)}]^2$$

Despejando  $\mu$  y  $\sigma$  como inversas también se pueden expresar como:

$$\mu = \kappa_c \bar{\mu} + \mu_0^{(c)}, \sigma = \kappa_c \bar{\phi}$$

Este reescalado es una reparametrización coherente en la que todas las fórmulas de actualización se aplican en la escala interna Glicko-2 y luego regresa a la escala original del curso.

Intuitivamente  $\kappa_c$  ajusta la unidad de rating de acuerdo a la incertidumbre de cada curso.

Ahora se inicializa los parámetros de cada estudiante dentro del curso de la siguiente manera. Cada estudiante  $i$  inicia con un factor conservador ideal  $\beta$ :

$$\mu_{ic}^{(c)} = \mu_0^{(c)} + \beta(N_{ic} - \bar{N}_c), \beta = 20$$

$$\sigma_{ic}^{(c)} = \sigma_0^{(c)} \wedge \omega_{ic}^{(c)} = \omega_0^{(c)}$$

En este punto, ya se tiene una tripleta inicial por estudiante en escala externa  $(\mu_{ic}^{(c)}, \sigma_{ic}^{(c)}, \omega_{ic}^{(c)})$ . El desplazamiento lineal o factor ideal por  $(N_{ic} - \bar{N}_c)$  preserva el orden de mérito inherente de los estudiantes por curso y hace que quien por este por encima de la media empiece más alto y viceversa.

Definimos los resultados de la simulación de los enfrentamientos, para lo cual definimos un umbral de “empate” adaptativo para curso  $c$ , este umbral ayuda a corregir los empates en cursos donde la variación es muy baja, sin embargo, la nota promedio  $\bar{N}_c$  es muy alta, sin este ajuste los alumnos con alto rendimiento se puntuarian como falsos positivos de bajo rendimiento.

$$\theta_c = \max \{0.3, 0.1s_c\}$$

Para cada par de alumnos por competir  $(i, j)$ , tal que  $i \neq j$  donde  $M_i = \{1, 0.5, 0\}$  es el resultado observado tras el enfrentamiento: 1 si gana (WIN), 0.5 si empata (DRAW), y si pierde 0 (LOSS).

$$s_{ij} = \begin{cases} 1, & N_{ic} - N_{jc} > \theta_c \\ 0.5, & |N_{ic} - N_{jc}| < \theta_c \\ 0, & N_{jc} - N_{ic} > \theta_c \end{cases}$$

Esto convierte las diferencias pequeñas de nota en empates, amortiguando el ruido en calificaciones casi iguales.

Mas adelante necesitamos actualizar los ratings mediante ecuaciones de Glicko-2 en la escala interna las variables internas para el estudiante  $i$  con oponentes  $j$ :

- Función de impacto, que reduce el peso de los oponentes con alta incertidumbre;

$$g(\bar{\phi}_j) = \frac{1}{\sqrt{1 + \frac{3(\bar{\phi}_j)^2}{\pi^2}}}$$

- Probabilidad esperada de victoria, logística en base  $e$ ;

$$E_{ij} = \frac{1}{1 + e^{-g(\bar{\phi}_j)[\bar{\mu}_i - \bar{\mu}_j]}}$$

- Varianza del estimador;

$$v_i^{-1} = \sum_j g(\bar{\phi}_j)^2 E_{ij}(1 - E_{ij})$$

- Gacia agregada;

$$\Delta_i = v_i \sum_j g(\bar{\phi}_j) (s_{ij} - E_{ij})$$

- Actualización de volatilidad, se debe resolver  $= \ln(\omega'_i)^2$  cuando  $f(x)$  es cero;

$$f(x) = \frac{e^x (\Delta_i^2 - \bar{\phi}_i^2 - v_i^2 - e^x)}{2(\bar{\phi}_i^2 + v_i^2 + e^x)^2} - \frac{x - \ln \omega_i^2}{\tau^2} = 0$$

Donde  $\tau = 0.5 > 0$  regula la variación permitida de la volatilidad

La solución numérica se obtiene con un método de Illinois (regula falsi modificado) donde tenemos que hallar un valor de  $x^*$ , raíz de la ecuación  $f(x) = 0$ ; también el valor de  $\omega'_i$  que es la nueva volatilidad del estudiante después de la actualización, que cuantifica la variabilidad esperada en su desempeño futuro, se calcula a partir de  $x^*$ .

El método de Illinois es una variante del método de regula falsi que acelera la convergencia mediante la modificación de los valores de la función en los extremos del intervalo cuando uno de ellos se estanca. Los pasos son los siguientes:

- Se determinan las cotas iniciales  $[A, B]$  tales que  $f(A).f(B) < 0$ , garantizando que haya una raíz en el intervalo;
- Dado que  $f(x) > 0$  para  $x$  muy negativo y  $f(x) < 0$  para  $x$  muy positivo, se pueden definir las cotas como:

$$A = \ln \omega_i^2 - 10\tau \wedge B = \ln \omega_i^2 + 10\tau$$

Asegurando que  $f(A) > 0$  y  $f(B) < 0$  en la mayoría de los casos, si no se cumple, se debe ajustar  $A$  y  $B$  hasta obtener signos opuestos. A continuación, se necesita iterar encontrando el punto de interacción  $C$  con la fórmula de la secante:

$$C = \frac{A \cdot f(B) - B \cdot f(A)}{f(B) - f(A)}$$

Evaluamos  $f(C)$ , si  $|f(C)| < \varepsilon$  es muy pequeño se considera que  $C$  es la raíz o si  $|B - A| < \varepsilon$  : se detiene el algoritmo Illinois y se toma  $x^* = C$ . Caso contrario se actualiza el intervalo:

- Si  $f(A) \cdot f(C) < 0$ , la raíz está en  $[A, C]$  por lo que se actualiza  $B = C$  y  $f(B) = f(C)$ ;
- Si  $f(A) \cdot f(C) > 0$ , la raíz está en  $[C, B]$  por lo que se actualiza  $A = C$  y  $f(A) = f(C)$ .

En el método de Illinois para evitar la lentitud en la convergencia, se reduce a la mitad el valor de la función en el extremo que se actualiza consecutivamente: Si se actualiza el mismo extremo (izquierdo o derecho) dos veces seguidas, en la siguiente iteración se utiliza  $f(A)/2$  o  $f(B)/2$  en el cálculo de  $C$ . Este algoritmo se implementa llevando un registro del último extremo actualizado en la que si se actualiza el extremo opuesto se resetea y se usan los valores originales de  $f(A)$  o  $f(B)$ .

El criterio de parada del algoritmo en el cual se detiene es cuando la longitud del intervalo es menor que la tolerancia  $|B - A| < \varepsilon$ ; el valor de  $x^* = C$ . Una vez obtenido  $x^*$  calculamos  $\omega'_i$ :

$$\omega'_i = e^{\frac{1}{2}x^*}$$

Habiendo hallado la nueva volatilidad, se calcula el RD previo y posterior respectivamente:

$$\bar{\phi}_i^* = \sqrt{\bar{\phi}_i^2 + \omega'^2} , \quad \bar{\phi}'_i = \left[ \frac{1}{(\bar{\phi}_i^*)^2} + \frac{1}{v_i} \right]^{-\frac{1}{2}}$$

Con el RD posterior calculado se procede a hallar la media posterior lo que nos servirá para hallar el nuevo rating:

$$\bar{\mu}'_i = \bar{\mu}_i + (\bar{\phi}'_i)^2 \sum_j g(\bar{\phi}_j) (s_{ij} - E_{ij})$$

Por último, actualizamos el nuevo rating regresando a la escala original del curso:

$$\mu'_i = \kappa_c \bar{\mu}'_i + \mu_0^{(c)}, \sigma'_i = \kappa_c \bar{\phi}'_i, \omega'_i$$

Una vez obtenidos los parámetros finales en escala externa para cada estudiante  $i$  en el curso  $c$ , a saber  $(\mu'_i, \sigma'_i, \omega'_i)$ , se procede a definir el rating individual Glicko-2 de dicho estudiante en el curso. En la práctica, se utiliza como medida del rendimiento consolidado el valor final de  $\mu'_i$ , el cual representa la expectativa de desempeño ajustada a la incertidumbre propia del estudiante. De esta manera, para cada curso  $c$  se obtiene:

$$R_i^{G,c} = \mu'_i$$

Donde  $R_i^{G,c}$  constituye el rating Glicko-2 final del estudiante  $i$  en el curso  $c$ . Este valor representa tanto el rendimiento académico observado como las interacciones competitivas simuladas dentro del curso, ya que integra los ajustes derivados de la función logística, la reducción de la incertidumbre final  $\sigma'_i$  y la volatilidad asociada final  $\omega'_i$ . Consideramos, un semestre  $s$  del cual se genera para cada estudiante un conjunto de ratings individuales:

$$R_{i,s} = \{R_i^{G,c} \mid c \in C_s(i)\}$$

Donde  $C_s(i)$  representa el conjunto de cursos en los que el estudiante  $i$  se matriculó durante el semestre  $s$ . Este conjunto de ratings constituye la base para el cálculo del promedio ponderado por semestre y, en etapas posteriores, para la clasificación por percentiles y la asignación de

estrellas. Ya obtenidos los puntajes Glicko-2 individuales en cada curso y consolidados por estudiante en un semestre, se calcula un promedio ponderado que resume el rendimiento académico global. Formalmente, para un estudiante  $i$  matriculado en  $m$  cursos del semestre  $s$ , se define:

$$R_i^{G,Promedio} = \frac{\sum_{c=1}^m R_i^{G,c}}{m}$$

Los valores  $R_i^{G,Promedio}$  se transforman en percentiles  $P(x)$  donde  $x$  denota el puntaje promedio de cada estudiante y  $P(x) \in [0,100]$  representa su posición relativa dentro de la distribución de todo el semestre sobre esta base se asig categorías de estrellas que permiten clasificar el rendimiento en cinco niveles cualitativos:

**Tabla 6**

*Percentiles por rendimiento para asignación de estrellas Glicko-2*

Condición del Percentil Glicko-2	Estrellas (Stars)	Rendimiento
$P(x) \geq 90\%$	☆☆☆☆☆	Excelente
$75\% \leq P(x) < 90\%$	☆☆☆☆	Alto
$50\% \leq P(x) < 75\%$	☆☆☆	Promedio
$25\% \leq P(x) < 50\%$	☆☆	Malo
$P(x) < 25\%$	☆	Bajo

*Fuente: Elaboración personal*

En particular, los estudiantes ubicados en el cuartil inferior  $P(x) < 25\%$  son clasificados con una sola estrella (☆), lo que permite identificarlos como casos de bajo rendimiento académico

en el semestre. Se muestran a continuación los resultados obtenidos (Tabla 8) en tiempo de ejecución:

**Tabla 7**

*Resultados obtenidos después de la ejecución del algoritmo Glicko-2 en los semestres 2023-I, 2023-II, 2024-I y 2024-II.*

Semestre	Tiempo (s)	Estudiantes	Cursos
2023-1	1675.81	18540	3556
2023-2	1638.02	18204	3541
2024-1	1748.32	18607	3669
2024-2	1761.76	18221	3715

*Fuente: Elaboración personal*

Los resultados obtenidos tras la implementación del algoritmo Glicko-2 muestran una consistencia tanto en el tiempo de ejecución como en la capacidad de procesamiento frente a un volumen considerable de datos. En los cuatro semestres analizados, el número de estudiantes evaluados oscila entre 18204 y 18607, mientras que los cursos se sitúan en un rango de 3 541 a 3 715. A pesar de esta variabilidad, el tiempo de ejecución se mantiene estable, con valores entre 1 675 y 1 761 segundos aproximadamente, lo que evidencia la eficiencia del algoritmo para gestionar bases de datos de gran tamaño sin que se produzcan incrementos significativos en los recursos computacionales empleados.

Estos resultados confirman la escalabilidad del modelo Glicko-2 en contextos educativos con datos masivos, teniendo un buen desempeño y clasificación confiable. Para finalizar, se exportan las clasificaciones a un archivo llamado Alumnos\_Clasificados\_Glicko\_2.xlsx para disponer más adelante de los datos.

En la implementación del algoritmo Glicko-2, al determinar el rating de un estudiante  $i$ , se usan las calificaciones actuales de los oponentes (que pueden haberse actualizado si ya pasaron por el bucle). En Glicko-2 puro la actualización es simultánea por periodo usando los ratings pre-periodo; en cambio en la versión adaptada para registros de notas de la UNSAAC es una aproximación secuencial de un paso. Para cursos con  $n_c$  moderado esto suele ser indistinguible; si se desearía exactitud periódica pre-post, se podría computar todo respecto de los ratings iniciales y aplicar luego la actualización de manera simultánea, pero en este caso no se aplica debido a que la clasificación debe ser relativa por alumno, y al mismo tiempo absoluta por curso.

El algoritmo Elo, anteriormente descrito en esta investigación, presenta una estructura computacional sencilla, pues cada actualización se resuelve en tiempo constante mediante una función logística y un ajuste lineal; sin embargo, su simplicidad limita la sensibilidad del modelo, al no incorporar un tratamiento explícito de la incertidumbre y depender de una constante fija de ajuste. En contraste, Glicko-2 introduce un cálculo iterativo de volatilidad que, si bien aumenta levemente la complejidad teórica, resulta más eficiente en escenarios académicos tanto en tiempo de ejecución como en gestión de memoria. Ello se debe a que el algoritmo adapta de manera dinámica la magnitud de las actualizaciones según la estabilidad del desempeño de los estudiantes, lo que reduce oscilaciones, evita recalibraciones innecesarias y acelera la convergencia global del sistema, ofreciendo así una estimación más robusta y precisa con un costo computacional totalmente manejable.

### **3.10. Identificación de los Alumnos con Bajo Rendimiento después de la Clasificación por el Algoritmo Glicko-2**

Categorizamos los puntajes numéricos o MMR obtenidos del algoritmo Glicko-2 en una clasificación cualitativa del rendimiento académico, se implementa un sistema de estrellas (de 1 a

5) como se realizó anteriormente con el algoritmo ELO. El procedimiento se desarrolla en tres fases: cálculo del rating general ponderado Glicko-2, asignación de estrellas según percentiles de corte y clasificación cualitativa del rendimiento.

En primer lugar, para cada estudiante  $i$  se calcula el rating promedio Glicko-2 en un semestre  $n$ , denotado como  $R_i^{(n)}$ . Se pondera este valor con un peso de competencia definido como  $\omega_i = T_i \cdot C_i$ , donde  $T_i$  representa el número de estudiantes con los que compitió indirectamente y  $C_i$  el número de cursos llevados. De este modo, el rating ponderado individual resulta:

$$RP_i^{(n)} = R_i^{(n)} \cdot \omega_i$$

El rating general ponderado de Glicko-2 para cada estudiante se obtiene como:

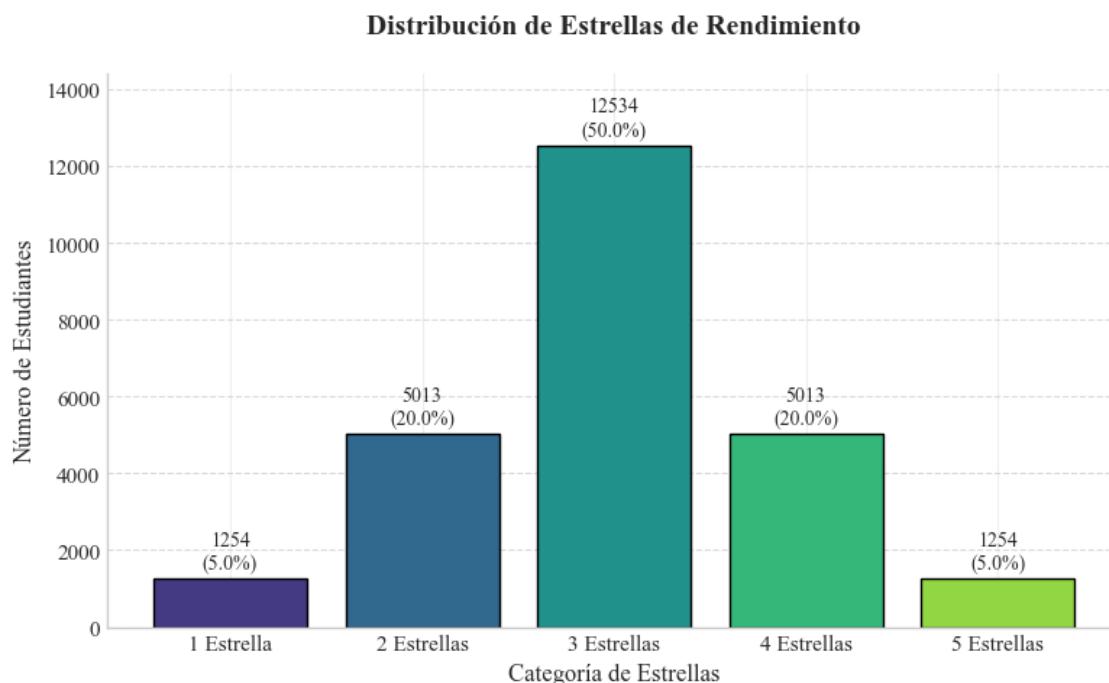
$$Glicko2\_GP_i = \frac{\sum_1^n RP_i^{(n)}}{\sum_1^n \omega_i}$$

Se emplea una distribución académica de cortes asimétricos (García Tárrago, 2020) (5%, 25%, 75% y 95%), lo que permite clasificar a los estudiantes en cinco categorías cualitativas generales, desde bajo (★) hasta excelente (★★★★★). Este criterio se justifica en entornos educativos por su mayor sensibilidad frente a distribuciones heterogéneas de desempeño. Con esto se logra identificar, de manera objetiva, a los estudiantes con bajo rendimiento, aquellos ubicados por debajo del percentil 5%.

Después de aplicar la distribución (Figura 13) se puede observar lo siguiente:

**Figura 11**

*Distribución y clasificación por Estrellas de Glicko-2 Ponderado General*



*Fuente: Elaboración personal*

Tras la clasificación general de los estudiantes mediante el algoritmo Glicko-2 y la posterior asignación de estrellas de rendimiento, se observa una distribución que replica de manera muy similar la obtenida bajo el sistema ELO. En particular, el 5% de los estudiantes se ubica en la categoría de 1 estrella (bajo) y un 20% en 2 estrellas (malo), lo que en conjunto representa al 25% de la población estudiantil considerada con desempeño por debajo de lo esperado. Por otro lado, el 50% se concentra en la categoría de 3 estrellas (rendimiento promedio), mientras que un 20% y un 5% alcanzan, respectivamente, 4 y 5 estrellas, asociadas a un nivel alto y excelente de rendimiento. Esta distribución asimétrica confirma que el modelo de percentiles aplicado genera una clasificación balanceada y coherente con los objetivos educativos del estudio.

Si bien la forma de la distribución es prácticamente equivalente a la del modelo ELO, el aspecto crítico para el desarrollo de la investigación radica en determinar si los estudiantes que aparecen en los percentiles más bajos son los mismos o difieren entre algoritmos. Identificar esta variación es fundamental, dado que el objetivo central es reconocer de manera precisa a los alumnos con bajo rendimiento y establecer si el cambio metodológico (de ELO a Glicko-2) repercute en la identificación de individuos específicos o únicamente en la magnitud de sus puntajes. En consecuencia, la comparación entre ambas clasificaciones más adelante en esta investigación permitirá evaluar la sensibilidad de cada enfoque y su pertinencia en entornos de gran escala educativa.

### **3.11. Correlación entre Glicko-2 General Ponderado y CGPA Final.**

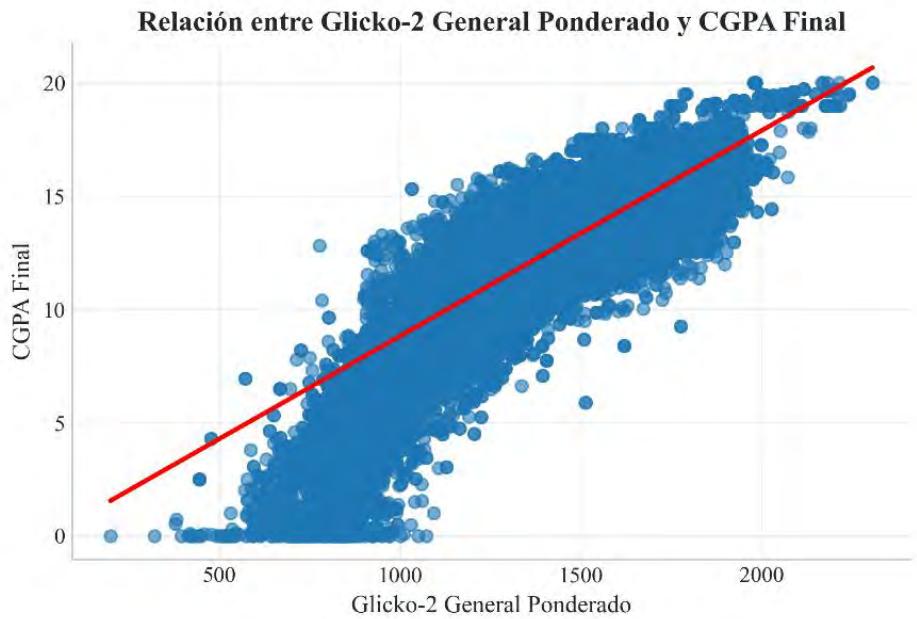
Los resultados de la correlación entre el CGPA final y el Glicko-2 General Ponderado muestran en la Figura 14 una asociación muy fuerte y estadísticamente significativa. El coeficiente de Pearson ( $r = 0.8213, p < 0.001$ ) indica una relación lineal positiva elevada, lo que significa que, en general, a medida que aumenta el rating Glicko-2 ponderado de los estudiantes, también tienden a mejorar sus calificaciones globales reflejadas en el CGPA. Esta relación se visualiza claramente en el diagrama de dispersión, donde los datos se agrupan en torno a la línea de tendencia, mostrando un patrón ascendente consistente.

Por su parte, el coeficiente de Spearman ( $\rho = 0.7948, p < 0.001$ ) refuerza este hallazgo desde una perspectiva no paramétrica, confirmando que el orden relativo de los estudiantes en el ranking Glicko-2 guarda una fuerte correspondencia con su posición en el CGPA, incluso si se relajan los supuestos de linealidad. En conjunto, se evidencia que el modelo Glicko-2 no solo permite clasificar a los estudiantes de manera robusta, sino que también logra capturar de forma

fiel la variabilidad de su desempeño académico real, siendo éste un sistema válido y consistente para la identificación del rendimiento de los estudiantes.

**Figura 12**

*Correlación entre Glicko-2 General Ponderado y CGPA Final de los alumnos clasificados*



*Fuente: Elaboración personal*

### **3.12. Clasificación de los estudiantes utilizando el algoritmo TrueSkill.**

El algoritmo TrueSkill Through Time (TTT) es una extensión del sistema de calificación bayesiano TrueSkill, implementado por Microsoft Research, diseñado para modelar la evolución de la habilidad de los jugadores a lo largo del tiempo. Mientras que TrueSkill clásico asume un nivel de habilidad relativamente estático, TTT incorpora un marco temporal que permite actualizar dinámicamente las distribuciones de habilidad conforme los participantes acumulan más partidas o evaluaciones. La habilidad de cada individuo se representa mediante distribuciones gaussianas que se propagan en el tiempo, aplicando técnicas de inferencia aproximada similares al filtrado

bayesiano. Esto lo convierte en una herramienta más realista en escenarios donde las capacidades no son fijas, sino que progresan o fluctúan según la experiencia y el contexto.

Para esta investigación, no se utilizó el modelo TTT estándar, sino su variante mejorada TTT-D (TrueSkill Through Time with Draws), que incorpora un tratamiento más sofisticado de los empates (draws). Este modelo ajusta de manera más precisa los draw margins (umbrales de empate), lo que permite una estimación más realista de la probabilidad de empates en partidas muy igualadas. Esta mejora es particularmente valiosa en entornos competitivos donde los empates son frecuentes y deben ser modelados con mayor fidelidad.

De esta manera, TTT-D no solo conserva su potencia en entornos competitivos, sino que también ofrece un enfoque más robusto en ambientes dinámicos y con alta incertidumbre. Este algoritmo es utilizado en videojuegos competitivos como Halo o Gears of War, donde se busca emparejar jugadores con habilidades similares y mantener partidas equilibradas.

La variante TTT-D (TrueSkill Through Time with Individual Draw Margins) introduce un refinamiento clave: márgenes de empate adaptativos para cada participante o enfrentamiento. En lugar de considerar los empates como un evento residual, el modelo incorpora explícitamente la posibilidad de que ocurran cuando las habilidades estimadas son cercanas dentro de un margen dinámico. En esta investigación, se adapta el modelo al contexto académico de la Universidad Nacional de San Antonio Abad del Cusco, donde permite analizar la evolución del rendimiento estudiantil a lo largo de los semestres y diferenciar con mayor precisión a los alumnos de rendimiento similar.

A cada estudiante  $i$  se le representa mediante una distribución normal de habilidades:

$$s_i \sim \mathcal{N}(\mu_i, \sigma_i^2)$$

Donde  $\mu_i$  es la estimación puntual de la habilidad y  $\sigma_i^2$  es la incertidumbre asociada a esa estimación. Un valor alto de  $\sigma_i^2$  indica que el sistema tiene menor confianza en la estimación actual del rating del estudiante, mientras que valores bajos sugieren que el rating está bien establecido basado en las comparaciones realizadas.

Para definir los parámetros iniciales de un alumno  $i$  en un curso  $c$  tales  $(\mu_i, \sigma_i^2)$  antes de las actualizaciones en nuestro sistema, se adaptan automáticamente los parámetros según las características estadísticas de cada curso  $c$ . Esto reconoce que diferentes cursos pueden tener diferentes niveles de dificultad, variabilidad en las evaluaciones, y distribuciones de notas.

La media inicial adaptativa alumno  $i$  en un curso  $c$  se calcula mediante:

$$\mu_i = \mu_0^{curso} = \lambda \cdot \bar{x}_c \cdot 100 + (1 - \lambda) \cdot \mu_g$$

Esta ecuación implementa un mecanismo de shrinkage bayesiano o fenómeno de encogimiento donde  $\lambda = n/(n + k)$  balancea entre la información específica del curso (promedio de notas  $\bar{x}_c$  del curso multiplicado por 100 para escalar) y una media global ( $\mu_g = 1500$ , mu global inicial estándar para TrueSkill). El parámetro  $k = 15$  actúa como un pseudoconteo que determina cuántas observaciones se necesitan para que la información del curso domine sobre el prior global. Cuando  $n$  es pequeño, el sistema confía más en la media global; cuando  $n$  es grande, utiliza principalmente el promedio del curso.

La desviación estándar inicial se adapta según:

$$\sigma_i^2 = (\sigma_0^{curso})^2 = \max(\sigma_{min}, \min(\sigma_{max}, 25 \cdot s_c))^2$$

Esta formulación escala la desviación estándar de las notas del curso  $s_c$  por un factor de 25, pero mantiene el resultado dentro de límites razonables  $[\sigma_{min}, \sigma_{max}]$  para evitar valores

extremos que podrían desestabilizar el sistema. Los límites típicos son  $\sigma_{min} = 120$  y  $\sigma_{max} = 400$ , asegurando que la incertidumbre inicial sea significativa pero no excesiva.

El parámetro adicional  $\beta$ , que controla la variabilidad del rendimiento, se ajusta basándose en el coeficiente de variación del curso:

$$\beta = \beta_0^{curso} = \max(\beta_{min}, \min(\beta_{max}, 100 + 200 \cdot CV_{curso}))$$

El coeficiente de variación  $CV_{curso} = s_c/\bar{x}_c$  mide la variabilidad relativa de las notas. Cursos con mayor variabilidad relativa reciben valores más altos de  $\beta$ , reconociendo que hay mayor aleatoriedad en las evaluaciones. Los límites típicos son  $\beta_{min} = 80$  y  $\beta_{max} = 300$ .

El parámetro adicional  $\varepsilon_i$ , margen de empate individualizado de cada estudiante, se ajusta basándose en su posición relativa en la distribución de notas del curso:

$$\varepsilon_i = \varepsilon_0^{curso} \cdot (1 + \alpha \cdot e^{-\beta_m \cdot |z_i|})$$

Donde representamos al z-score del estudiante  $z_i = (x_i - \bar{x}_c)/s_c$  mide cuántas desviaciones estándar se encuentra la nota del estudiante respecto a la media del curso. El factor multiplicativo  $(1 + \alpha \cdot e^{-\beta_m \cdot |z_i|})$  aumenta el margen de empate para estudiantes cerca de la media ( $|z_i|$  pequeño) y lo reduce para estudiantes en los extremos de la distribución. Los parámetros  $\alpha = 0.5$  y  $\beta_m = 1.0$  controlan la magnitud y la tasa de este ajuste. Esta formulación reconoce que es más difícil distinguir entre estudiantes con rendimiento promedio que entre aquellos en los extremos de la distribución.

Una de las contribuciones principales de la implementación es la adaptación del modelo original para manejar empates de manera sofisticada. En el contexto académico, los empates son frecuentes cuando las diferencias entre las notas de los estudiantes son mínimas. El modelo de comparación

asume que cuando dos estudiantes compiten, su rendimiento observable incluye tanto su habilidad latente como un componente de ruido aleatorio:

$$p_i = \theta_i + \epsilon_i, p_j = \theta_j + \epsilon_j$$

Los términos  $\epsilon_i$  y  $\epsilon_j$  representan las variaciones aleatorias en el rendimiento, modeladas como  $\epsilon_i, \epsilon_j \sim \mathcal{N}(0, \beta^2)$ . El parámetro  $\beta^2$  captura la variabilidad inherente en las evaluaciones académicas, reconociendo que el rendimiento de un estudiante en un examen específico puede no reflejar perfectamente su habilidad verdadera debido a factores externos como el estado de ánimo, la salud, o la familiaridad con los temas específicos evaluados.

La diferencia de rendimiento entre dos estudiantes sigue entonces una distribución normal:

$$\Delta = p_i - p_j \sim \mathcal{N}(\mu_i - \mu_j, c^2)$$

El parámetro  $c^2 = 2\beta^2 + \sigma_i^2 + \sigma_j^2$  representa la varianza total del sistema, incorporando tanto la incertidumbre en las estimaciones de habilidad de ambos estudiantes como la variabilidad del rendimiento. Este término es crucial para el cálculo de las probabilidades de diferentes resultados.

Esta mejora del algoritmo TTT-D utiliza márgenes de empate dinámicos que se adaptan a las características específicas de cada par de estudiantes. El margen de empate efectivo  $\epsilon_{ij}$  no es un valor fijo, sino que depende de múltiples factores:

$$\epsilon_{ij} = \frac{\epsilon_i + \epsilon_j}{2} \cdot \left(1 + \frac{\sigma_i + \sigma_j}{2}\right) + \max [0, \frac{\epsilon_i + \epsilon_j}{2} - 0.1|\Delta_{obs}|]$$

Esta formulación incorpora tres componentes importantes. Primero, el promedio de los márgenes individuales  $(\epsilon_i + \epsilon_j)/2$  establece una base personalizada para cada comparación.

Segundo, el factor de ajuste por incertidumbre  $\left(1 + \frac{\sigma_i + \sigma_j}{2}\right)$  aumenta el margen de empate cuando hay mayor incertidumbre en las estimaciones de los ratings, reconociendo que en estos casos es más difícil distinguir diferencias reales de habilidad. Tercero, el término de ajuste por diferencia de rendimiento observada reduce el margen de empate cuando las diferencias de notas son muy grandes, reflejando que diferencias sustanciales en el rendimiento son menos propensas a ser consideradas empates.

Después, el algoritmo calcula las probabilidades de victoria, empate y derrota utilizando la función de distribución acumulativa normal estándar  $\Phi$ . Para una comparación entre los estudiantes  $i$  y  $j$ , las probabilidades se definen como:

La probabilidad de que el estudiante  $i$  gane está dada por:

$$P(\text{win}_i) = \Phi\left(\frac{\mu_i - \mu_j - \varepsilon_{ij}}{c}\right)$$

Esta ecuación evalúa la probabilidad de que la diferencia de rendimiento supere el margen de empate  $\varepsilon_{ij}$ . El numerador  $(\mu_i - \mu_j - \varepsilon_{ij})$  representa la ventaja efectiva que necesita el estudiante  $i$  para ganar, mientras que el denominador  $c$  normaliza esta diferencia según la variabilidad total del sistema.

La probabilidad de empate se calcula como:

$$P(\text{draw}_i) = \Phi\left(\frac{\mu_i - \mu_j + \varepsilon_{ij}}{c}\right) - \Phi\left(\frac{\mu_i - \mu_j - \varepsilon_{ij}}{c}\right)$$

Esta expresión captura la probabilidad de que la diferencia de rendimiento caiga dentro del intervalo  $[-\varepsilon_{ij}, \varepsilon_{ij}]$ , representando situaciones donde las diferencias de rendimiento son lo suficientemente pequeñas para considerarse empates.

Finalmente, la probabilidad de derrota se obtiene por complemento:

$$P(\text{loss}_i) = 1 - \Phi\left(\frac{\mu_i - \mu_j + \varepsilon_{ij}}{c}\right)$$

Una vez definidas las probabilidades de victoria, empate y pérdida, el algoritmo TTT-D usa un mecanismo de actualización bayesiana, que ajusta las estimaciones de habilidad después de cada comparación. La actualización de la media  $\mu_i$  sigue la regla:

$$\mu_i^* = \mu_i + \frac{\sigma_i^2}{2} v$$

Donde el factor  $v$  conocido como el término de corrección, depende del resultado observado y representa cuánto debe ajustarse la estimación basándose en la sorpresa del resultado. Para una victoria o derrota, el término  $v$  se calcula como:

$$\pm v = \frac{\varphi\left(\frac{\mu_i - \mu_j - \varepsilon_{ij}}{c}\right)}{\Phi\left(\frac{\mu_i - \mu_j - \varepsilon_{ij}}{c}\right)}$$

Donde se asigna el operador (+) para victoria y el operador – para la derrota.

Donde  $\varphi$  representa la función de densidad de probabilidad normal estándar. Esta razón entre la densidad y la distribución acumulativa, conocida como la razón de Mills inversa, cuantifica la información ganada del resultado observado. Cuando el resultado es muy esperado (denominador grande), la actualización es pequeña. Cuando el resultado es sorprendente (denominador pequeño), la actualización es mayor. Para empates, el cálculo captura la diferencia entre las densidades en los límites del margen de empate, normalizada por la probabilidad total del empate:

$$v = \frac{\varphi\left(\frac{\mu_i - \mu_j - \varepsilon_{ij}}{c}\right) - \varphi\left(\frac{\mu_i - \mu_j + \varepsilon_{ij}}{c}\right)}{\Phi\left(\frac{\mu_i - \mu_j + \varepsilon_{ij}}{c}\right) - \varphi\left(\frac{\mu_i - \mu_j - \varepsilon_{ij}}{c}\right)}$$

Ahora necesitamos actualizar la nueva varianza en el cual término  $w$  incluye asegura que la incertidumbre disminuya con cada comparación, pero nunca desaparezca completamente (límite inferior de 0.1):

$$w = v^2 + \frac{\left(\frac{\mu_i - \mu_j + \varepsilon_{ij}}{c}\right)\varphi\left(\frac{\mu_i - \mu_j + \varepsilon_{ij}}{c}\right) - \left(\frac{\mu_i - \mu_j - \varepsilon_{ij}}{c}\right)\varphi\left(\frac{\mu_i - \mu_j - \varepsilon_{ij}}{c}\right)}{\Phi\left(\frac{\mu_i - \mu_j + \varepsilon_{ij}}{c}\right) - \Phi\left(\frac{\mu_i - \mu_j - \varepsilon_{ij}}{c}\right)}$$

$$\sigma_i^{2,*} = \sigma_i^2 \cdot \max [1 - \frac{\sigma_i^2}{2} \cdot w, 0.1]$$

Por consiguiente, continuamos con las comparaciones o rondas multiples all-for-all (todos contra todos) este proceso de clasificación dentro de cada curso sigue un algoritmo sistemático de comparaciones exhaustivas. Para un curso con  $n$  estudiantes, el algoritmo ejecuta los siguientes pasos:

Primero, se inicializan los ratings de todos los estudiantes usando los parámetros adaptativos calculados anteriormente para el curso. Cada estudiante recibe un rating inicial que refleja tanto su posición relativa en el curso como la incertidumbre asociada.

Luego, para cada par de estudiantes  $(i, j)$  donde  $i < j$ , se realiza una comparación. Se calcula la diferencia de rendimiento observada  $\Delta_{obs} = |x_i - x_j|$  basada en las notas reales. El umbral de empate efectivo se determina como:

$$\tau_{ij} = \frac{\varepsilon_i + \varepsilon_j}{2} \cdot (1 + 0.1 \cdot s_c)$$

Este umbral combina los márgenes individuales de ambos estudiantes y se ajusta por la variabilidad del curso. El resultado de la comparación se determina según:

- Si  $\Delta_{obs} < \tau_{ij}$ , se considera un empate;
- Si  $x_i - x_j > \tau_{ij}$ , el estudiante  $i$  gana;
- Si  $x_j - x_i > \tau_{ij}$ , el estudiante  $j$  gana.

Después de determinar el resultado, se actualizan los ratings de ambos estudiantes usando las ecuaciones bayesianas descritas anteriormente.

Este proceso resulta en  $n(n - 1)/2$  comparaciones totales, asegurando que cada estudiante sea comparado con todos los demás en el curso. La complejidad computacional es  $O(n^2)$  por curso, lo cual es manejable incluso para cursos con gran cantidad de estudiantes.

Por último, calculamos el rating promedio simple y promedio ponderado de cada alumno. El rating promedio simple se obtiene como:

$$\bar{\mu}_i = \frac{1}{K} \sum_{k=1}^K \mu_i^{(k)}$$

Donde  $K$  es el número de cursos en los que participó el estudiante y  $\mu_i^{(k)}$  es su rating en el curso  $k$ . Adicionalmente, se calcula un rating ponderado que da mayor peso a los cursos con más estudiantes:

$$\tilde{\mu}_i = \frac{\sum_{k=1}^K n_k \cdot \mu_i^{(k)}}{\sum_{k=1}^K n_k}$$

Esta ponderación reconoce que los ratings obtenidos en cursos más grandes pueden ser más confiables debido al mayor número de comparaciones realizadas. El sistema TTT-D también mantiene estadísticas sobre la incertidumbre promedio  $\bar{\sigma}_i$  y el margen de empate promedio  $\bar{\varepsilon}_i$  a través de todos los cursos.

Para la asignación final de rankings y categorías (representadas por estrellas), el sistema implementa un enfoque conservador que considera la incertidumbre en las estimaciones:

$$\mu_i^{**} = \mu_i^* - 2\sigma_i^*$$

Este rating conservador representa un límite inferior del intervalo de confianza del 95% para la habilidad del estudiante. Al usar este valor para la asignación de categorías, el sistema asegura que los estudiantes solo reciban clasificaciones altas cuando hay suficiente evidencia estadística de su rendimiento superior. Los percentiles se calculan sobre estos ratings conservadores, aquellos con percentiles iguales o superiores al 90% reciben cinco estrellas (☆☆☆☆☆) por rendimiento excelente, mientras que los ubicados entre el 75% y 90% obtienen cuatro (☆☆☆☆), y así sucesivamente hasta llegar al cuartil inferior ( $P(x) < 25\%$ ), donde se asigna una sola estrella (☆), identificando a los estudiantes con bajo rendimiento en el semestre. Se muestran a continuación los resultados obtenidos en tiempo de ejecución:

### **Tabla 8**

*Resultados obtenidos después de la ejecución del algoritmo TrueSkill (TTT-D) en los semestres 2023-I, 2023-II, 2024-I y 2024-II.*

Semestre	Tiempo (s)	Estudiantes	Cursos
2023-1	1550.62	18540	3556
2023-2	1532.71	18204	3541
2024-1	1689.78	18607	3669
2024-2	1711.73	18221	3715

*Fuente: Elaboración personal*

Los resultados de la ejecución del algoritmo TrueSkill (TTT-D) muestran una consistencia notable en el tiempo de procesamiento y en la magnitud de datos procesados entre los distintos semestres, con tiempos que varían entre 1532.71 s y 1711.73 s. En todos los semestres obtuvo un mejor tiempo de ejecución en comparación con el algoritmo Glicko-2 y ELO, lo que refleja una mejor eficiencia de TrueSkill en contextos con grandes cargas de cursos y estudiantes. Esto sugiere que el algoritmo es el más estable de los tres debido a su sensibilidad a la densidad e incertidumbre de los registros de notas de los alumnos de la UNSAAC. Para finalizar, se exportan los resultados a un archivo llamado Alumnos\_Clasicados\_TrueSkill.xlsx.

### **3.13. Identificación de los alumnos con bajo rendimiento después de la clasificación por el algoritmo TrueSkill**

Categorizamos los puntajes numéricos obtenidos del algoritmo TrueSkill (TTT-D) en una clasificación cualitativa del rendimiento académico, implementando un sistema de estrellas (de 1 a 5), similar al aplicado anteriormente con los algoritmos ELO y Glicko-2. El procedimiento se desarrolla en tres fases: cálculo del rating general ponderado TrueSkill, asignación de estrellas según puntos de corte percentiles y clasificación cualitativa del rendimiento.

En primer lugar, para cada estudiante  $i$  se calcula el promedio de TrueSkill en un semestre  $n$ , denotado como  $TS_i^{(n)}$ . Este valor se pondera con un peso de competencia definido como  $\omega_i = T_i \cdot C_i$ , donde  $T_i$  representa el número de estudiantes con los que compitió indirectamente y  $C_i$  el número de cursos llevados. De este modo, el rating ponderado individual resulta como:

$$TS\_P_i^{(n)} = TS_i^{(n)} \cdot \omega_i$$

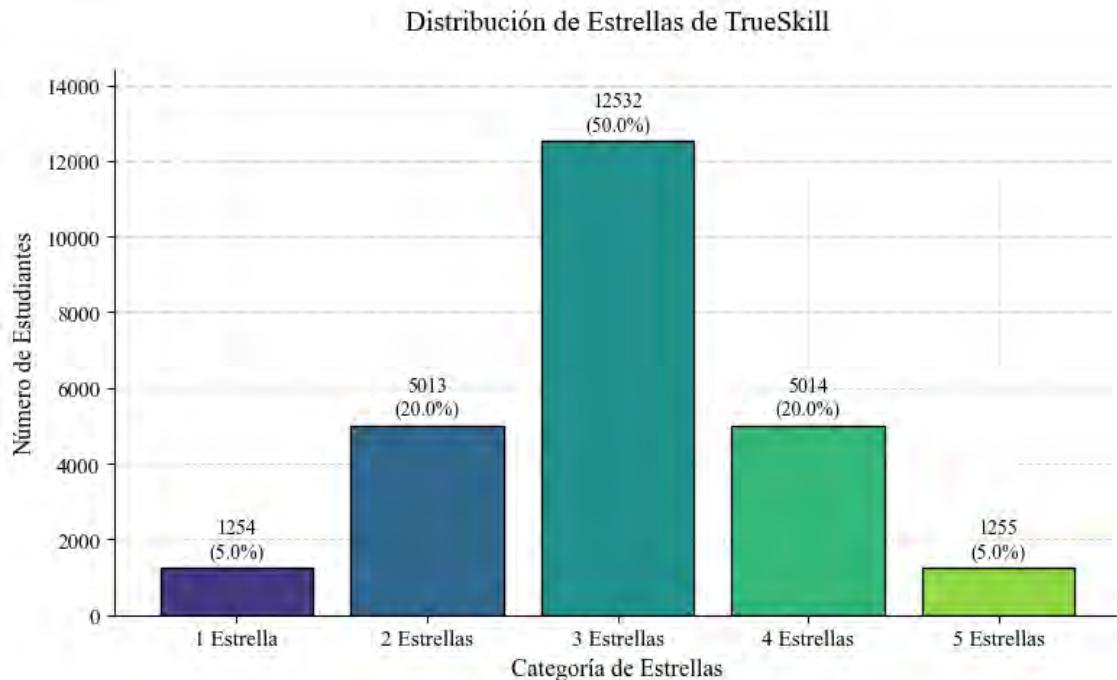
El rating general ponderado de Glicko-2 para cada estudiante se obtiene como:

$$TrueSkill\_GP_i = \frac{\sum_1^n TS\_P_i^{(n)}}{\sum_1^n \omega_i}$$

Posteriormente, se emplea una distribución académica de cortes asimétricos (García Tárrago, 2020), basada en los percentiles 5%, 25%, 75% y 95%, lo que permite clasificar a los estudiantes en cinco categorías cualitativas generales, desde bajo (★) hasta excelente (★★★★★), procediendo de la misma manera como se hizo en el algoritmo Elo y Glicko-2. Y obtenemos los siguientes resultados:

**Figura 13**

*Distribución y clasificación por Estrellas de TrueSkill Ponderado General*



*Fuente: Elaboración personal*

La clasificación con TrueSkill muestra una distribución fuertemente centrada en la categoría intermedia (3 estrellas), concentrando al 50% de los estudiantes en un rendimiento “promedio” y dejando solo a pequeños grupos en los extremos (1 y 5 estrellas, cada uno con 5%). Esto refleja una estratificación balanceada y simétrica, que es consistente con los principios de clasificación académica propuestos por García Tárrago (2020). Lo notable es que los resultados obtenidos con TrueSkill son prácticamente idénticos en distribución global a los alcanzados mediante los algoritmos ELO y Glicko-2, lo cual indica que, aunque difieren en la forma de calcular los ratings, las reglas de corte percentiles impuestas por la distribución académica dominan el resultado final de la clasificación. En otras palabras, el algoritmo subyacente impacta en los valores individuales de los ratings, pero la forma de asignación de estrellas fuerza a que la proporción de estudiantes en cada categoría sea la misma. Más allá de la distribución global, sería útil analizar si los mismos estudiantes son clasificados de manera consistente en bajo/alto rendimiento entre TrueSkill, ELO y Glicko-2. Esto daría una medida de robustez inter-algoritmo. Además de las estrellas, en la sección de resultados se podría considerar indicadores adicionales como la media, varianza y correlaciones entre algoritmos, para sustentar que TrueSkill no solo clasifica de manera similar, sino que mantiene coherencia con las otras métricas de desempeño académico.

### **3.14. Correlación entre TrueSkill General Ponderado y CGPA Final.**

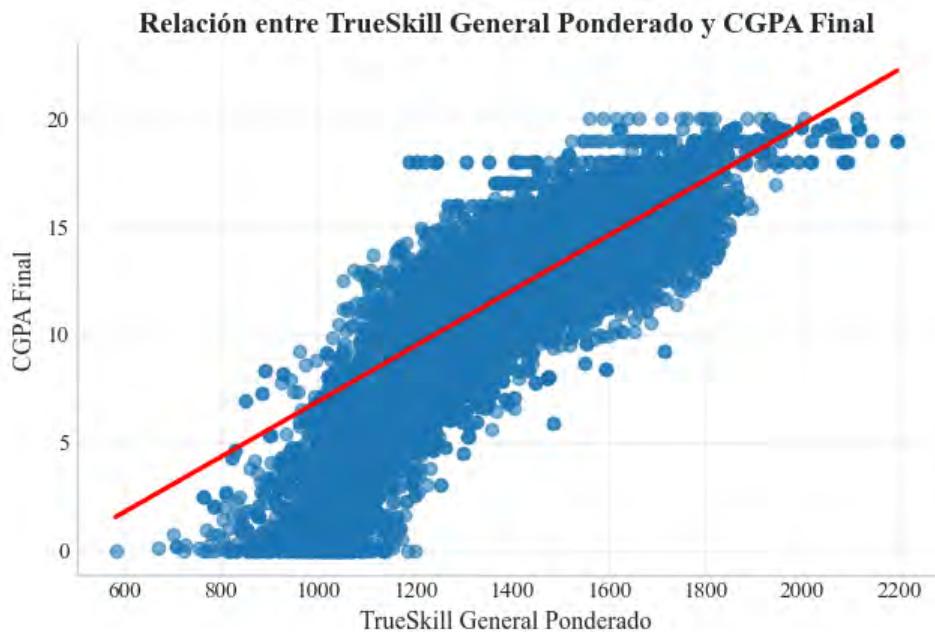
Los resultados de la correlación entre el CGPA final y el TrueSkill General Ponderado muestran en la Figura 16 una asociación muy fuerte y estadísticamente significativa. El coeficiente de Pearson ( $r = 0.8078$ ,  $p < 0.001$ ) indica una relación lineal positiva elevada, lo que significa que, en general, a medida que aumenta el rating TrueSkill ponderado de los estudiantes, también tienden a mejorar sus calificaciones globales reflejadas en el CGPA. Este comportamiento se aprecia en el

diagrama de dispersión, donde los puntos siguen un patrón ascendente en torno a la línea de tendencia, confirmando la consistencia del modelo.

Por su parte, el coeficiente de Spearman ( $\rho = 0.7783$ ,  $p < 0.001$ ) respalda este hallazgo desde una perspectiva no paramétrica, mostrando que el orden relativo de los estudiantes en el ranking de TrueSkill mantiene una correspondencia fuerte con su posición en el CGPA, incluso cuando se relajan los supuestos de linealidad. En conjunto, estos resultados evidencian que el modelo TrueSkill no solo clasifica de manera robusta a los estudiantes, sino que también captura de forma confiable la variabilidad de su desempeño académico real, lo que lo consolida como un sistema válido y consistente para la evaluación del rendimiento estudiantil.

**Figura 14**

*Correlación entre TrueSkill General Ponderado y CGPA Final de los alumnos clasificados*



*Fuente: Elaboración personal*

## CAPÍTULO IV: ANÁLISIS Y DISCUSIÓN DE RESULTADOS

En el presente capítulo se exponen los resultados obtenidos a partir de la implementación de los algoritmos de emparejamiento ELO, Glicko-2 y TrueSkill sobre los registros académicos de los estudiantes de la Universidad Nacional de San Antonio Abad del Cusco en el periodo postpandemia. Cada algoritmo fue aplicado siguiendo la misma secuencia metodológica: cálculo de los rankings, identificación de los estudiantes con bajo rendimiento y análisis de correlación con el CGPA final. De este modo, se buscó evidenciar el comportamiento de cada modelo en la clasificación estudiantil, así como sus diferencias.

Asimismo, los resultados se presentan con correspondencia con los objetivos específicos de la investigación, organizados de manera progresiva para facilitar la comparación. En primer lugar, se muestran los hallazgos obtenidos tras la imputación de datos inconsistentes; en segundo lugar, los resultados descriptivos generales y detección de valores atípicos en registros de notas; en tercer lugar, los tiempos de ejecución y la eficiencia de cada algoritmo y por último la comparación en la identificación de alumnos de bajo rendimiento después de la clasificación de los algoritmos ELO, Glicko-2 y TrueSkill.

### 4.1. Resultados en Inconsistencias en los Datos Conforme a la Resolución N.<sup>o</sup> CU-0359-2015-UNSAAC.

El proceso de validación y depuración de la base de datos de alumnos de la UNSAAC del semestre 2023-1 al 2024-II permitió identificar un total de 413791 registros académicos, de los cuales 413692 fueron consistentes y solo 99 presentaron inconsistencias, alcanzándose un porcentaje global de consistencia del 99.98 %. Estos resultados evidencian que la información proporcionada por el Centro de Cómputo de la UNSAAC presenta un alto grado de confiabilidad,

aunque fue necesario aplicar técnicas de imputación para garantizar la completitud de los datos antes de la implementación de los algoritmos de emparejamiento.

El análisis por semestre mostró un comportamiento estable en los niveles de consistencia, con porcentajes superiores al 99.7 % en todos los períodos evaluados. La mayor proporción de inconsistencias se concentró en los cursos de Medicina sin calificaciones parciales, los cuales fueron imputados en su mayoría (entre del 86.89 % hasta un 92.75%, según el semestre). En contraste, un porcentaje reducido de inconsistencias (7.25 % a 13.11 %) correspondió a casos en los que el examen subsanatorio no reemplazaba correctamente el promedio final, los cuales no pudieron ser imputados debido a que no fue necesario ya que para la implementación de los algoritmos de emparejamiento se utilizaron directamente los campos de exámenes parciales y sustitutorio. En conjunto, se obtuvo que el 90.04 % de los registros inconsistentes fueron imputados, consolidando así una base de datos depurada y confiable, adecuada para el posterior análisis con los algoritmos ELO, Glicko-2 y TrueSkill. A continuación, se muestran los resultados en la siguiente tabla:

**Tabla 9**

*Consistencia de registros de notas UNSAAC del semestre 2023-I al 2024-II por semestre*

Semestre	Total	Registros Cons.	% Cons.	Registros Incons.	Incons. Imp. (Cursos de Medicina sin parciales)	Incons. No Imp. (Subs. incorrecto)	% Imp.	% No Imp.
2023-1	104 001	103 734	99.74 %	267	232	35	86.89 %	13.11 %
2023-2	101 239	101 002	99.77 %	237	214	23	90.30 %	9.70 %
2024-1	104 760	104 463	99.72 %	297	270	27	90.91 %	9.09 %
2024-2	103 791	103 598	99.81 %	193	179	14	92.75 %	7.25 %
<b>TOTAL</b>	<b>413 791</b>	<b>412 797</b>	<b>99.76 %</b>	<b>994</b>	<b>895</b>	<b>99</b>	<b>90.04 %</b>	<b>9.96 %</b>

*Fuente: Elaboración personal*

#### **4.2. Resultado descriptivo general de los datos académicos y detección de valores atípicos en registros de notas.**

En el periodo comprendido entre los semestres 2023-I y 2024-II, la Universidad Nacional de San Antonio Abad del Cusco (UNSAAC) registró un total de 73746 matrículas a lo largo de los semestres; siendo 25162 alumnos únicos, distribuidos en 15150 asignaturas únicas. La carga académica promedio por estudiante se mantuvo relativamente estable, oscilando entre 5.55 y 5.68 asignaturas por alumno, lo que refleja una estructura académica homogénea a lo largo de los cuatro semestres. Este patrón permite sostener que, pese a las fluctuaciones propias de la matrícula, la distribución de cursos por estudiante presenta una consistencia en torno al promedio global de 5.61 asignaturas por alumno.

**Tabla 10**

*Descripción general de los registros de notas de la UNSAAC del semestre 2023-I al 2024-II*

<b>Semestre</b>	<b>Alumnos Matriculados</b>	<b>Asignaturas Únicas</b>	<b>Carga Académica Promedio por Alumno</b>
2023-1	18593	3677	5.59
2023-2	18240	3665	5.55
2024-1	18635	3881	5.62
2024-2	18278	3927	5.68
<b>TOTAL/PROMEDIO</b>	<b>73746</b>	<b>15150</b>	<b>5.61</b>

*Fuente: Elaboración personal*

El análisis descriptivo de las calificaciones parciales reveló la presencia de outliers asociados principalmente a notas bajas, lo que constituye un indicador relevante para la identificación de estudiantes en riesgo académico. La distribución de notas por semestre muestra valores de mediana cercanos a 15, con rangos intercuartílicos (IQR) entre 2.9 y 3.6 puntos, lo que evidencia una relativa estabilidad en la dispersión de las calificaciones. Sin embargo, al aplicar los

límites de detección de valores atípicos, se identificó que entre el 5 % y el 8 % de las notas correspondieron a outliers, concentrándose en valores por debajo del límite inferior de cada distribución (aproximadamente entre 7.5 y 9.0). Estos resultados indican que, pese a que la mayoría de los estudiantes se mantiene en un rango esperado de rendimiento, existe un subconjunto consistente de alumnos con calificaciones significativamente bajas en cada parcial, se observa que el segundo parcial concentra la mayor proporción de notas atípicamente bajas, con valores entre el 7 % y el 8.35 % en los distintos semestres analizados. Esto sugiere que esta evaluación representa un punto crítico en el rendimiento académico, probablemente porque coincide con una etapa de mayor exigencia y acumulación de carga académica para los alumnos. En contraste, el primer parcial muestra niveles intermedios (entre 5.5 % y 7.2 %), mientras que el tercer parcial mantiene proporciones más reducidas de outliers (cerca al 5 %), lo que indica que el mayor desafío para los estudiantes ocurre en la evaluación intermedia del semestre.

**Tabla 11**

*Valores atípicos en registros de notas por parcial de la UNSAAC de los algoritmos ELO, Glicko-2 y TrueSkill desde el semestre 2023-I al 2024-II*

Semestre	Parcial	Q1	Mediana (Q2)	Q3	IQR	Límite inferior	Límite superior	Nº Inliers	Nº Outliers (%)	Nº Outliers (%)
2023-1	1er	13.29	15	16.33	3.04	8.73	20	97419	93.67	6576
2023-1	2do	13.42	15.13	16.6	3.18	8.65	20	95907	92.22	8094
2023-1	3er	13	15	16.67	3.67	7.495	20	50180	48.25	5238
2023-2	1er	13.4	15	16.33	2.93	9.005	20	93938	92.79	7301
2023-2	2do	13.5	15.2	16.67	3.17	8.745	20	92787	91.65	8452
2023-2	3er	13.33	15	16.67	3.34	8.32	20	49844	49.23	5008
2024-1	1er	13.14	14.9	16.33	3.19	8.355	20	98717	94.23	6041
2024-1	2do	13.33	15	16.5	3.17	8.575	20	96694	92.3	8063

2024-1	3er	13.1	15	16.6	3.5	7.85	20	49806	47.54	5329	5.09
2024-2	1er	13.285	15	16.33	3.045	8.7175	20	97987	94.41	5804	5.59
2024-2	2do	13.4	15.05	16.6	3.2	8.6	20	96338	92.82	7450	7.18
2024-2	3er	13.2	15	16.58	3.38	8.13	20	49205	47.41	5045	4.86

*Fuente: Elaboración personal*

#### **4.3. Resultados comparativos en tiempos de ejecución de los algoritmos ELO, Glicko-2, TrueSkill**

Analizando comparativamente de los tiempos de ejecución, se revela disparidades significativas en la eficiencia computacional de los tres algoritmos evaluados. El modelo ELO, implementado en su configuración base, registró los mayores tiempos de procesamiento en todos los períodos analizados, con un valor máximo de 2091.27 segundos en el semestre 2024-II. Este desempeño se atribuye a su arquitectura determinista y a la falta de un mecanismo inherente para cuantificar la incertidumbre, lo que lo obliga a procesar cada interacción de manera secuencial e individual, haciéndolo sensible a un gran volumen de datos y a la incertidumbre.

En contraste, el algoritmo Glicko-2, cuya implementación incorpora una estructura adaptativa que considera la media y la desviación estándar del curso como el espacio competitivo, demostró una mejora sustancial en eficiencia, con tiempos fluctuando entre 1638.02 y 1761.76 segundos. Su superioridad sobre el modelo ELO radica en la integración de una desviación estándar (RD) que modela la certidumbre de cada estimación, permitiendo así actualizaciones de habilidad más inteligentes y con una convergencia más rápida.

Para finalizar, la variante del último algoritmo: TrueSkill Through Time with Draw Margins (TTT-D); se estableció como el más eficiente, reduciendo los tiempos de ejecución a un rango entre 1532.71 y 1711.73 segundos. Esta optimización es consecuencia directa de su

sofisticado marco teórico bayesiano probabilístico. La implementación de un draw margin bayesiano dinámico, junto con técnicas de inferencia aproximada sobre distribuciones gaussianas que se propagan en el tiempo análogas al filtrado bayesiano, permiten actualizar de manera cohesiva las habilidades de todos los participantes. Adicionalmente, el shrinkage bayesiano ajusta las estimaciones hacia la media del contexto (curso), estabilizando los resultados y reduciendo la necesidad de iteraciones onerosas, mientras que la integración del z-score de cada estudiante ofrece una métrica estandarizada sin un overhead computacional adicional.

**Tabla 12**

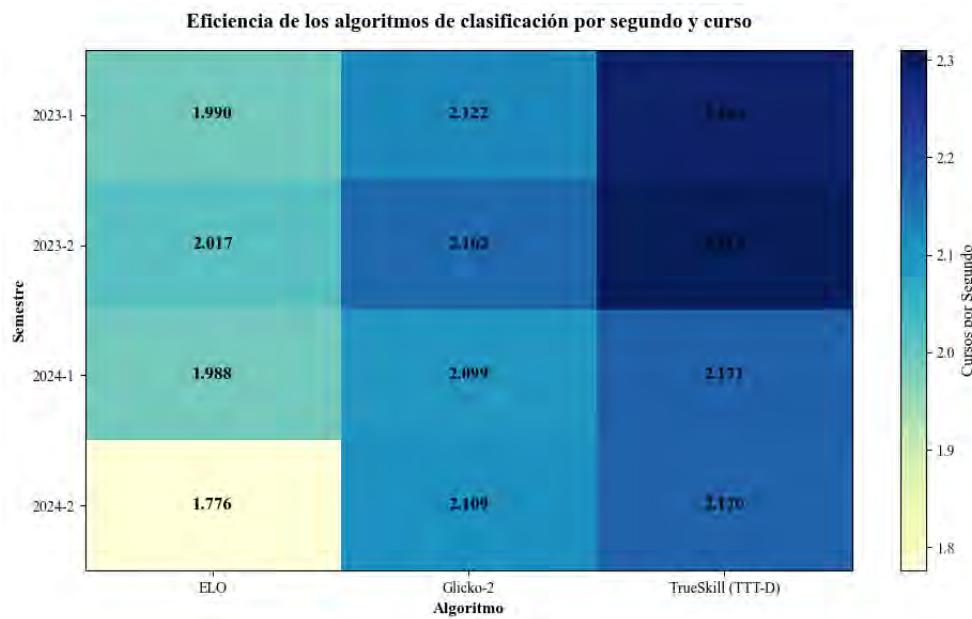
Tiempo de ejecución y promedio de cursos procesados por segundo de los algoritmos ELO, Glicko-2 y TrueSkill desde el semestre 2023-I al 2024-II

Sem.	Alums.	Cursos	t_ELO (s)	t_Glicko-2 (s)	t_TTT-D (s)	Cursos/s ELO	Cursos/s Glicko-2	Cursos/s TrueSkill
2023-1	18540	3556	1787.21	1675.81	1550.62	1.43	1.47	1.55
2023-2	18204	3541	1755.99	1638.02	1532.71	1.42	1.46	1.56
2024-1	18607	3669	1845.26	1748.32	1689.78	1.99	2.1	2.17
2024-2	18221	3715	2091.27	1761.76	1711.73	1.78	2.11	2.17

*Fuente: Elaboración personal*

**Figura 15**

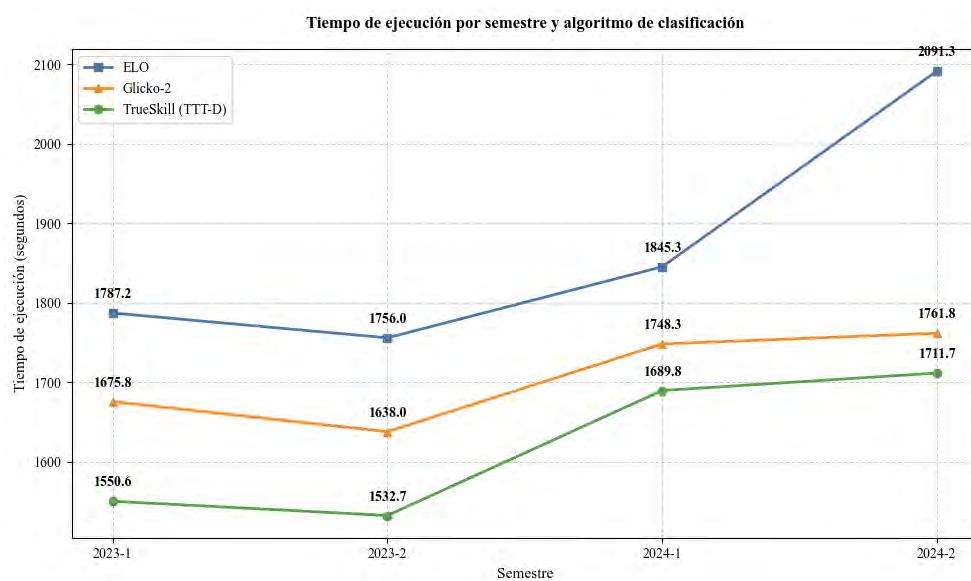
*Eficiencia de los algoritmos ELO, Glicko-2 y TrueSkill por segundo y curso desde el semestre 2023-I al 2024-II*



Fuente: Elaboración personal

**Figura 16**

*Tiempo de ejecución por semestre de los algoritmos de clasificación de los algoritmos ELO, Glicko-2 y TrueSkill desde el semestre 2023-I al 2024-II*



Fuente: Elaboración personal

#### **4.4. Comparación en la identificación de los alumnos con bajo rendimiento después de la clasificación por el algoritmo ELO, Glicko-2 y TrueSkill**

El análisis comparativo de los resultados obtenidos a partir de la clasificación de los estudiantes con bajo rendimiento (★) muestra que los tres algoritmos presentan niveles muy similares de detección global. Tanto TrueSkill (TTT-D) como Glicko-2 identificaron un total de 1254 alumnos (73.7 % del universo analizado), mientras que el modelo ELO detectó 1251 estudiantes (73.5 %). Estas cifras evidencian que, en términos agregados, las tres metodologías son consistentes en la delimitación de la población estudiantil en riesgo académico (Tabla 14).

**Tabla 13**

*Estadísticas comparativas de alumnos con bajo rendimiento identificados por los algoritmos ELO, Glicko-2 y TrueSkill desde el semestre 2023-I al 2024-II*

Algoritmo	Total de alumnos con bajo rendimiento (★)	Porcentaje universo
TrueSkill (TTT-D)	1254	73.70%
Glicko-2	1254	73.70%
ELO	1251	73.50%

*Fuente: Elaboración personal*

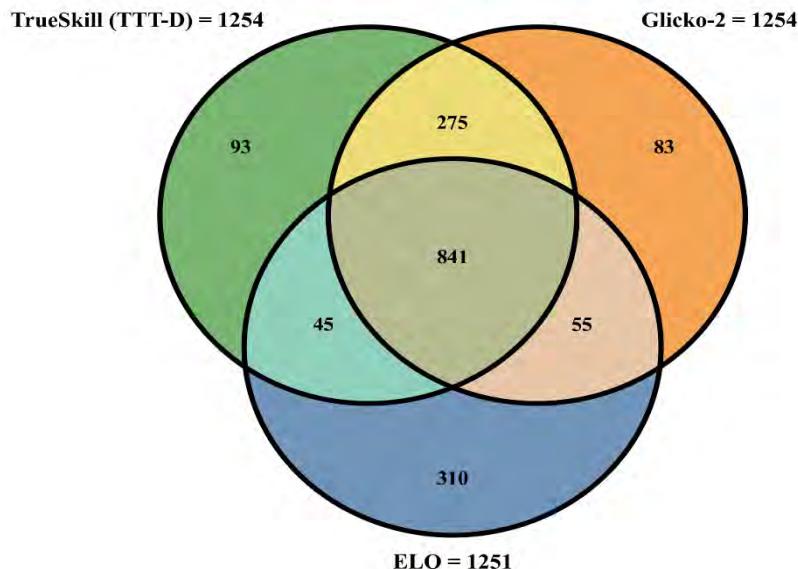
Al analizar las intersecciones y exclusividades (Figura 19), se observan diferencias significativas en la manera en que cada algoritmo discrimina casos específicos. Existen subconjuntos diferenciados: TrueSkill y Glicko-2 coincidieron de manera exclusiva en 275 alumnos (16.2 %), mientras que los pares TrueSkill–ELO y Glicko-2–ELO coincidieron en 45 (2.6 %) y 55 (3.2 %) estudiantes, respectivamente.

En cuanto a los casos exclusivos (Figura 19), se destaca que ELO identificó de manera individual a 310 estudiantes (18.2 %), proporción considerablemente mayor a los detectados únicamente por TrueSkill (93 alumnos, 5.5 %) o por Glicko-2 (83 alumnos, 4.9 %). Este hallazgo sugiere que el algoritmo ELO tiende a sobre clasificar en relación con los otros modelos, posiblemente por su estructura más rígida en el cálculo del rendimiento acumulado. En contraste, los algoritmos TrueSkill (TTT-D) y Glicko-2, al incorporar parámetros adaptativos y de variación, ofrecen una clasificación más convergente y precisa, disminuyendo el número de discrepancias respecto al grupo central de estudiantes.

**Figura 17**

*Diagrama de Venn de la intersección de alumnos con bajo rendimiento según ELO, Glicko-2 y TrueSkill (2023-I a 2024-II)*

**Nº de alumnos con Bajo Rendimiento identificados por algoritmo ELO, Glicko-2 y TruSkill**



*Fuente: Elaboración personal*

**Tabla 14**

*Análisis detallado de intersecciones y exclusividades en la identificación de alumnos con bajo rendimiento por los algoritmos ELO, Glicko-2 y TrueSkill*

Categoría	Cantidad	Porcentaje
Exclusivo TrueSkill (TTT-D)	93	5.50%
Exclusivo Glicko-2	83	4.90%
Exclusivo ELO	310	18.20%
TrueSkill $\cap$ Glicko-2 (exclusivo)	275	16.20%
TrueSkill $\cap$ ELO (exclusivo)	45	2.60%
Glicko-2 $\cap$ ELO (exclusivo)	55	3.20%
TrueSkill $\cap$ Glicko-2 $\cap$ ELO	841	49.40%
Universo (Total de alumnos encontrados con bajo rendimiento ★)	1702	100%

Fuente: Elaboración personal

En conjunto, los resultados (Tabla 15) muestran un alto consenso en la identificación de estudiantes con bajo rendimiento, pero también evidencian diferencias en los casos marginales según el algoritmo aplicado. Mientras que TrueSkill (TTT-D) y Glicko-2 ofrecen clasificaciones más estables y coherentes entre sí, el modelo ELO presenta un mayor número de discrepancias, al identificar de forma exclusiva a 310 alumnos (18.2 %). Esto sugiere que ELO tiende a sobredimensionar los casos de bajo rendimiento, a diferencia de los enfoques adaptativos de TrueSkill y Glicko-2, que reducen la dispersión en los resultados. El universo total de alumnos con bajo rendimiento identificados en la investigación asciende a 1702, donde se observa que el 49.4% (841 alumnos) fueron detectados de manera simultánea por los tres algoritmos (TrueSkill, Glicko-2 y ELO), lo que evidencia una alta coincidencia en la detección del bajo rendimiento académico.

Para complementar el análisis de intersecciones, se seleccionaron muestras aleatorias de alumnos exclusivos por conjunto, identificados por cada algoritmo. Estas muestras permiten observar de manera cualitativa el perfil académico que explica por qué un estudiante es clasificado solo y únicamente por ELO, Glicko-2 o TrueSkill como de bajo rendimiento. Las siguientes tablas a continuación muestran ejemplos representativos de cada grupo:

**Tabla 15**

*Muestra aleatoria de alumnos identificados exclusivamente por el algoritmo TrueSkill (TTT-D) como de bajo rendimiento académico (★)*

ID	Sem.	Código	Asignatura	P1	P2	P3	Sust.	Subs.	Promedio Final
12914	2023-1	ED122BEU	EDUCACIÓN COMUNITARIA INTERCULTURAL	16.33	16.67				17
12914	2023-1	ED123CEU	TUTORIA Y ORIENTACIÓN EDUCATIVA	15.75	16.75				16
12914	2023-1	LC422AEU	TEORÍA Y ANÁLISIS DEL TEXTO LITERARIO	16.33	11.33	13.33			14
12914	2023-1	ED121EEU	GESTIÓN DEL CURRÍCULO	14	11.75	11.5	8		12
12914	2023-1	LC430AEU	FONÉTICA Y FONOLOGÍA	13.33	12.67	1	9		10
12914	2023-1	LC415AEU	INTRODUCCIÓN A LA LINGÜÍSTICA	14	11.33	0	8		9
12914	2024-1	LC416AEU	MORFOLOGÍA DEL ESPAÑOL	7.2	0		0		4
12914	2024-1	LC423AEU	LITERATURA PERUANA I	4.67	0		0		2

12914	2024-1	LC415AEU	INTRODUCCIÓN A LA LINGÜISTICA	1	1	1	0	1
12914	2024-1	ED120CEU	GESTION DE LA EDUCACION	0	0	0	0	0
12914	2024-1	ED126CEU	ESTRATEGIAS DE ENSEÑANZA Y APRENDIZAJE	0	0	0		0
12914	2024-1	ED124BEU	PSICOLOGIA EDUCATIVA (EVOLUTIVA)	0	0	0	0	0
18545	2023-1	QU903AEN	QUÍMICA	13	12	5.25	0	10
508	2023-1	ED410AMI	MANIFESTACIONES ARTÍSTICAS Y DEPORTIVAS	11.67	17		0	14
508	2023-1	ML301AMI	TRATAMIENTO DE MINERALES	13	9.67	17.67	11	14
508	2023-1	IM221AMI	MECÁNICA DE ROCAS II	13.75	14.8	12.8	9	14
508	2023-1	IM120BMI	MAQUINARIA MINERA I	11	12	9.67	3	11
508	2023-1	IM313AMI	GEOESTADÍSTICA II	7.33	10.33		0	9
508	2023-2	IM120AMI	MAQUINARIA MINERA I	9.67	13.33	13.33	8	12
508	2023-2	IM313AMI	GEOESTADÍSTICA II	11.67	7.67		0	10
508	2023-2	IM105AMI	MÉTODOS DE EXPLOTACIÓN SUBTERRANEA	8	6	5	0	6
508	2023-2	IE350AMI	INGENIERÍA ELÉCTRICA	11	0			6

508	2023-2	IM223AMI	GEOMECANICA	0	0	0	0
508	2024-1	IM120BMI	MAQUINARIA MINERA I	7	0	0	2

Fuente: Elaboración personal

**Tabla 16**

*Muestra aleatoria de alumnos identificados exclusivamente por el algoritmo Glicko-2 como de bajo rendimiento académico (★)*

ID	Sem.	Código	Asignatura	P1	P2	P3	Sust.	Subs.	Promedio Final
21348	2023-1	FP901AFO	FILOSOFIA Y ETICA	2.5	2.5				3
21348	2023-1	AS901AFO	SOCIEDAD Y CULTURA	0	0				0
21348	2023-1	DE901AFO	CONSTITUCION POLITICA Y DERECHOS HUMANOS	14.5	14.25				14
21348	2023-1	ME901AFO	MATEMATICA I	10.4	8.2	0	0		6
21348	2023-1	LC901AFO	REDACCION DE TEXTOS	15	15	15			15
21348	2023-1	ED901AFO	ESTRATEGIAS DE APRENDIZAJE AUTONOMO	15	15	15			15
24097	2023-1	ME901BBI	MATEMATICA I	2.8	11	4.8	0		6
24097	2023-1	QU130BBI	BIOQUÍMICA DE LA CÉLULA	9.17	12	4.5	0		9
24097	2023-1	CB622BBI	ZOOLOGÍA SISTEMÁTICA I	12.67	9	0			7

---

24097	2023-1	CB201BBI	BOTANICA	14.33	9.33	0	0	8
24097	2023-1	CB102BBI	BIOLOGÍA DE LA CÉLULA I	6.5	8.5	0	0	5
24097	2024-1	ME901ABI	MATEMATICA I	14.3	14.9	15.6	0	15
24097	2024-1	CB102ABI	BIOLOGÍA DE LA CÉLULA I	13.2	11.6	14	15	14
24097	2024-2	ME902ABI	MATEMÁTICA II	12.2	11.8	0	0	8
24097	2024-2	CB103ABI	BIOLOGÍA DE LA CÉLULA II	9.4	11.6	10.3	0	10
24097	2024-2	CB902ABI	ECOLOGÍA Y MEDIO AMBIENTE	15.33	13.67	9.67	10	14
24097	2024-2	CB201ABI	BOTANICA	14.33	12.67	15	0	14
24097	2024-2	QU130ABI	BIOQUÍMICA DE LA CÉLULA	11.15	11.93	11.53	10	12
18616	2024-1	MEG02AMI	CÁLCULO I	8.4	9	6.2	0	8
18616	2024-1	HIG01AMI	HISTORIA CRÍTICA DEL PERÚ E IDENTIDAD NACIONAL	15.56	16.22			16
18616	2024-1	MEG01AMI	ÁLGEBRA Y GEOMETRÍA ANALÍTICA	6	10	2	10	8
18616	2024-1	QUG01AMI	QUÍMICA GENERAL	14.13	11.88	13.5	12	14

---

18616	2024-1	FIG01AMI	FÍSICA I	11.75	9	16.25	2	12
18616	2024-1	IMG01AMI	ECOLOGÍA Y MEDIO AMBIENTE	7.2	8.4		0	8
18616	2024-2	LCG01AMI	LINGÜÍSTICA Y COMUNICACIÓN HUMANA	0	0	0	0	0
18616	2024-2	MEG02BMI	CÁLCULO I	0	1.2	0	0	0
18616	2024-2	IMG02AMI	DIBUJO MINERO ASISTIDO POR EL COMPUTADOR	14	0		0	7
18616	2024-2	MEG03AMI	ESTADÍSTICA GENERAL	0	0	0	0	0
18616	2024-2	FIG01AMI	FÍSICA I	0	0	0	0	0
18616	2024-2	IFG01AMI	PENSAMIENTO COMPUTACIONAL E INTELIGENCIA ARTIF.	0	0		0	0

Fuente: Elaboración personal

**Tabla 17**

*Muestra aleatoria de alumnos identificados exclusivamente por el algoritmo ELO como de bajo rendimiento académico (★)*

ID	Sem.	Código	Asignatura	P1	P2	P3	Sust.	Subs.	Promedio Final
9725	2023-1	FP901ADR	FILOSOFIA Y ETICA	17	18				18
9725	2023-1	ED901ADR	ESTRATEGIAS DE APRENDIZAJE AUTONOMO	14.67	14	14	0		14
9725	2023-1	DE901ADR	CONSTITUCION POLITICA Y DERECHOS HUMANOS	2	12.5		0		7

9725	2023-1	AS901BDR	SOCIEDAD Y CULTURA	0	0	0	0
9725	2023-1	IF902ADR	TECNOLOGIAS DE LA INFORMACIÓN Y LA COMUNICACIÓN	0	0	0	0
9725	2023-2	DE857ADR	ACTIVIDAD CO-CURRICULAR I	17	12	15	15
9725	2023-2	AS901BDR	SOCIEDAD Y CULTURA	3.7	10.71	0	7
9725	2023-2	DE559ADR	DERECHO EMPRESARIAL	0	0	0	0
9725	2023-2	DE901BDR	CONSTITUCION POLITICA Y DERECHOS HUMANOS	5.1	0		3
9725	2023-2	DE461ADR	CONTROL GUBERNAMENTAL	0	6		3
9725	2023-2	DE675ADR	HISTORIA DEL DERECHO PERUANO	0	0	0	0
9725	2024-1	DE460ADR	DERECHO DE CONTRATACIONES DEL ESTADO	0	0	0	0
9725	2024-1	IF902BDR	TECNOLOGIAS DE LA INFORMACIÓN Y LA COMUNICACIÓN	0	0	0	0
9725	2024-1	DE563ADR	DERECHO COMERCIAL III (DERECHO BANCARIO, DERECHO...)	0	0		0
9725	2024-1	DE456ADR	DERECHO ELECTORAL	0	0		0
9725	2024-1	DE454ADR	DERECHO CONSTITUCIONAL ECONOMICO	0	0		0

9725	2024-2	DE461ADR	CONTROL GUBERNAMENTAL	11.4	0	0	0	6
9725	2024-2	DE252ADR	DERECHO PENAL II (PARTE ESPECIAL I)	0	0	0	0	0
9725	2024-2	DE460ADR	DERECHO DE CONTRATACIONES DEL ESTADO	0	0	0	0	0
14116	2023-1	FP901CEU	FILOSOFIA Y ETICA	16	15.5			16
14116	2023-1	DE901CEU	CONSTITUCION POLITICA Y DERECHOS HUMANOS	14.67	14.33			15
14116	2023-1	ED901CEU	ESTRATEGIAS DE APRENDIZAJE AUTONOMO	13.7	15.8	14.45	12	15
14116	2023-1	LC901CEU	REDACCION DE TEXTOS	16.67	15.33	13.67	0	15
14116	2023-1	AS901CEU	SOCIEDAD Y CULTURA	16	14.9		0	15
14116	2023-1	ME901FEU	MATEMATICA I	13.6	13.9	10.3	0	13
14116	2023-2	IF902CEU	TECNOLOGIAS DE LA INFORMACIÓN Y LA COMUNICACIÓN	5	0			3
14116	2023-2	FP902CEU	LIDERAZGO Y HABILIDADES SOCIALES	1	0			1
14116	2023-2	FP903CEU	PSICOLOGÍA GENERAL	0	0	1	0	0
14116	2023-2	ED118AEU	INTRODUCCIÓN A LAS CIENCIAS DE LA EDUCACIÓN	0	0	0	0	0

14116	2023-2	ED119AEU	EDUCACIÓN Y GESTIÓN AMBIENTAL	0	0	0		0
14116	2023-2	FP904CEU	INTRODUCCIÓN A LA EPISTEMOLOGÍA	0	0	0		0
14274	2023-1	AS733AAN	ETNOLOGÍA MUNDIAL COMPARADA	14.2	11.6	0	0	9
14274	2023-1	AS719AAN	SIMBOLOGÍA Y RITUAL	4	0	0.67	0	2
14274	2023-1	AS813BAN	ANTROPOLOGÍA Y DESARROLLO I: TEORÍAS EVOLUCIÓN	0	1	0	0	0
14274	2023-1	AS816AAN	BIOCULTURAL	0	0	0	0	0
14274	2024-1	AS808AAN	ANTROPOLOGÍA POLÍTICA I	0	0	0		0
14274	2024-1	AS818AAN	AMAZONIA REGIONAL	0	0	0		0
14274	2024-1	AS701AAN	ECOLOGÍA CULTURAL	0	0	0	0	0

*Fuente: Elaboración personal*

En el caso de TrueSkill (TTT-D), los estudiantes exclusivos presentan trayectorias mixtas: asignaturas aprobadas con calificaciones medias-altas conviven con otras con notas extremadamente bajas, incluso cercanas a cero. Ello sugiere que este algoritmo es sensible a vulnerabilidades focalizadas, identificando a quienes, pese a un desempeño aceptable en algunos cursos, arrastran debilidades críticas en otros.

Por su parte, los alumnos exclusivos de Glicko-2 exhiben un patrón de inestabilidad académica, con altibajos más pronunciados entre cursos de desempeño regular y otros con notas

mínimas, lo que refleja su capacidad de captar la irregularidad más que el bajo rendimiento absoluto.

Finalmente, los casos exclusivos de ELO corresponden mayormente a estudiantes con promedios finales extremadamente bajos o abandono académico, caracterizados por múltiples calificaciones de cero. Este comportamiento explica el mayor número de exclusividades detectadas por este algoritmo, aunque también evidencia una tendencia a la sobre clasificación, al incluir situaciones extremas que ya serían fácilmente detectables por los sistemas académicos tradicionales. En conjunto, estos hallazgos refuerzan que la intersección común de los tres algoritmos constituye el grupo más confiable de bajo rendimiento, mientras que los resultados exclusivos representan una clasificación que describen el comportamiento diferenciado de cada algoritmo de emparejamiento.

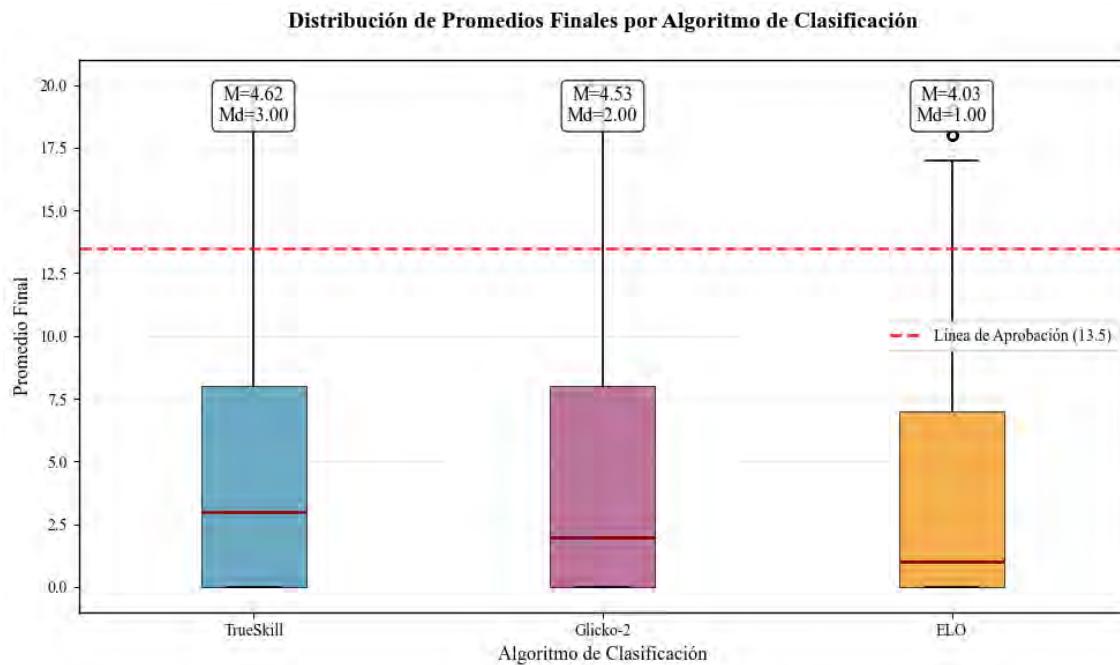
Los resultados descriptivos (Figura 20) muestran diferencias relevantes en la capacidad de los algoritmos para identificar estudiantes con bajo rendimiento. En términos generales, TrueSkill (TTT-D) y Glicko-2 presentan un comportamiento muy similar, ambos identificando poco más de 9300 estudiantes en cuatro semestres, con tasas de aprobación prácticamente idénticas (12.5% y 12.6%, respectivamente). Por su parte, el algoritmo ELO, bajo una estructura estándar, detecta un mayor número de estudiantes (9770), lo que se traduce en una tasa de aprobación ligeramente superior (14.0%). Esta diferencia sugiere que ELO posee un sesgo hacia una clasificación más amplia, lo que incrementa la probabilidad de incluir casos límite dentro del grupo de bajo rendimiento.

Al analizar los promedios finales, se observa que tanto TrueSkill (Media = 4.62; Desviación Estándar = 5.24) como Glicko-2 (Media = 4.53; Desviación Estándar = 5.22) mantienen distribuciones similares, con medias bajas que reflejan un rendimiento

consistentemente deficiente en los grupos detectados. En contraste, ELO presenta un promedio más reducido (Media = 4.03; Desviación Estándar = 5.43), lo que refuerza la idea de que este algoritmo tiende a concentrar casos de bajo rendimiento más extremo, aunque con un rango de aprobación relativamente mayor.

**Figura 18**

*Distribución de promedios finales de los estudiantes clasificados con bajo rendimiento por algoritmos de emparejamiento: ELO, Glicko-2 y TrueSkill*



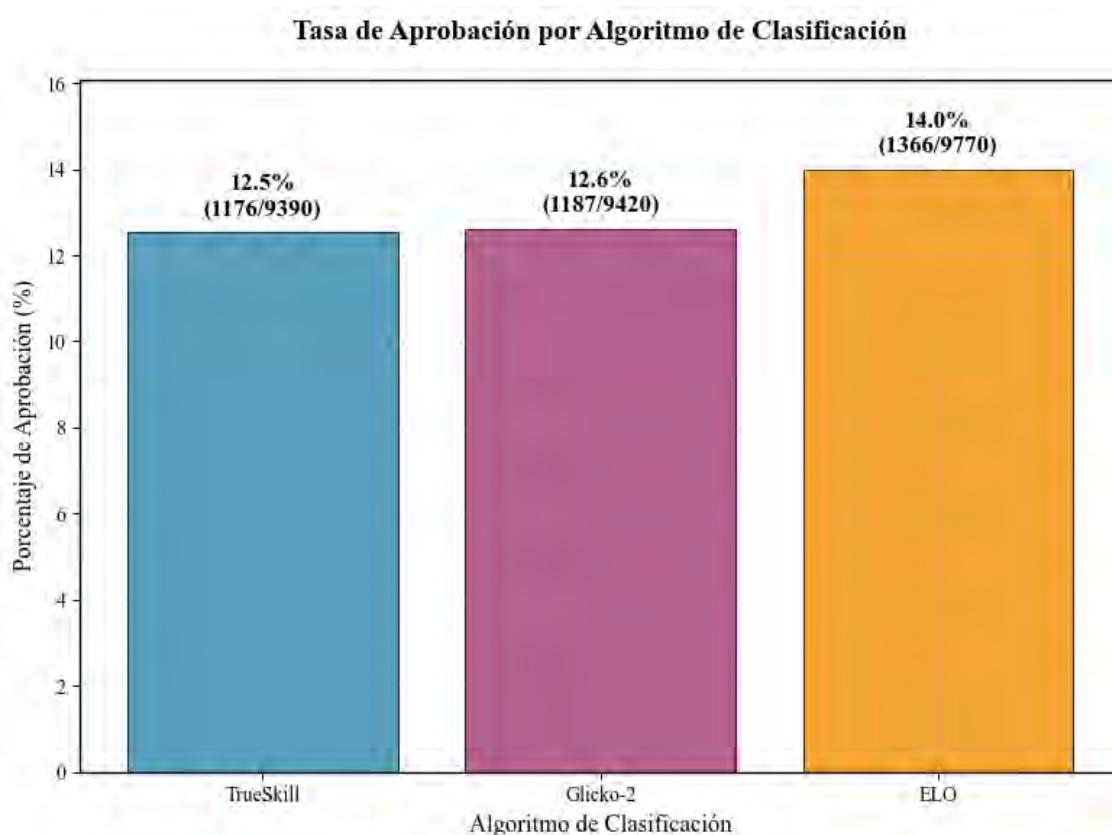
*Fuente: Elaboración personal*

El análisis de las tasas de aprobación (Figura 21), cursos aprobados sobre el total, muestra que ELO presenta la mayor efectividad con un 14.0% (1,366 de 9,770 estudiantes), seguido por Glicko-2 con 12.6% (1,187 de 9,420 estudiantes) y TrueSkill con 12.5% (1,176 de 9,390 estudiantes). Esto sugiere que ELO demuestra mayor precisión en la identificación de casos de estudiantes con cursos más recuperables, detectando estudiantes en riesgo moderado que mantienen potencial de recuperación académica, mientras que TrueSkill y Glicko-2 muestran

comportamientos similares en precisión de clasificación de estudiantes con mayor cantidad de cursos desaprobados.

**Figura 19**

*Tasa de aprobación de estudiantes clasificados con bajo rendimiento por algoritmos de emparejamiento ELO, Glicko-2 y TrueSkill.*



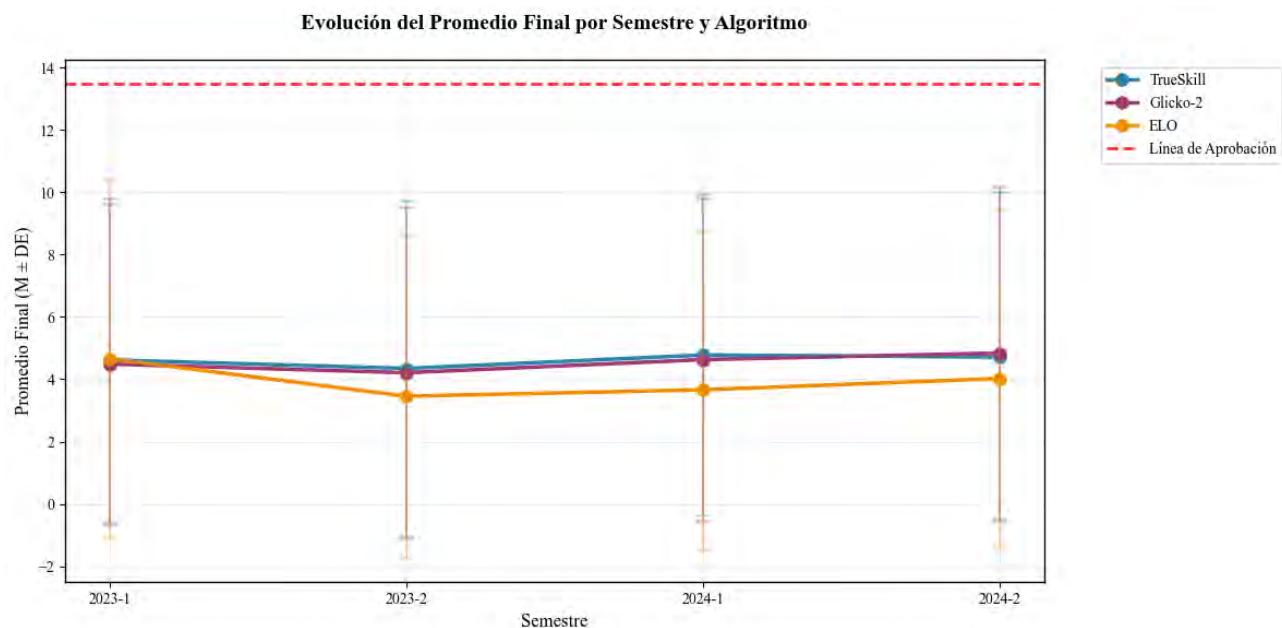
*Fuente: Elaboración personal*

Por otro lado, el análisis temporal (Figura 22) del período 2023-1 a 2024-2 revela patrones distintivos en el comportamiento temporal de cada algoritmo. TrueSkill mantiene estabilidad notable con promedios entre 4.5-5.0 puntos y consistencia diagnóstica temporal, mientras que Glicko-2 exhibe adaptabilidad algorítmica superior con evolución ascendente progresiva hacia los semestres recientes. ELO presenta el comportamiento más distintivo, manteniendo consistentemente los promedios más bajos (~4.0 puntos) con menor variabilidad temporal,

sugiriendo que a pesar que ELO clasifica a alumnos con mayor tasa de aprobación por curso, se observa que determina a alumnos con menor promedio general.

**Figura 20**

*Análisis temporal por promedio de estudiantes clasificados con bajo rendimiento por algoritmos de emparejamiento ELO, Glicko-2 y TrueSkill*



*Fuente: Elaboración personal*

Más adelante se realizó análisis exhaustivo de las asignaturas más desaprobadas (Tabla 19) que revela un patrón consistente de cursos recurrentes que afectan negativamente en el rendimiento académico de los alumnos. Entre ellas, destacan cursos como: Historia Crítica del Perú e Identidad Nacional, Ecología y Medio Ambiente, Sociedad y Cultura, Matemática I, Cálculo I, Física I y Química General, entre otras; todas estas asignaturas corresponden a cursos de Estudios Generales de la UNSAAC de primeros semestres. Dichos cursos presentan promedios extremadamente bajos (entre 3.12 y 4.29), con tasas de desaprobación superiores al 85% y un alto número de casos críticos (entre 340 y 513 estudiantes con calificaciones menores a 5.0). La persistencia de estas dificultades a lo largo de cuatro semestres analizados indica que se trata de problemáticas estructurales del plan de estudios primario de los alumnos de nuevo ingreso.

Las asignaturas de Estudios Generales más críticas son: Historia Crítica del Perú e Identidad Nacional, Ecología y Medio Ambiente y Sociedad y Cultura que exhiben índices problemáticos elevados, con tasas de desaprobación superiores al 73% y con alta recurrencia demostrada en todos los algoritmos. Estos resultados sugieren que no solo las áreas de ciencias exactas generan dificultades, sino que también existen brechas importantes en asignaturas que incluyen competencias de comprensión lectora, expresión escrita y reflexión crítica, que son transversales al desempeño académico.

Esto refuerza que el bajo rendimiento no responde únicamente a dificultades conceptuales en los alumnos, sino también supondría carencias en habilidades metacognitivas y hábitos de estudio en los primeros semestres o; probablemente métodos de calificación, rubrica y/o metodológica ineficiente.

**Tabla 18**

*Top 10 asignaturas más desaprobadas identificados por algoritmos de emparejamiento ELO, Glicko-2 y TrueSkill*

Asignatura	Total detecciones	Dist. (TS / Glicko-2 / ELO)	Prom(D esv.)	Tasa Reprobados (Promedio <9.5)	Casos Críticos (Promedi o < 5.0)	Sem.
Historia Crítica del Perú e Identidad Nacional	885	327 / 303 / 255	5.30 (6.02)	73.20%	476	2
Ecología y Medio Ambiente	788	290 / 282 / 216	4.17 (4.71)	83.90%	488	4
Sociedad y Cultura	724	227 / 226 / 271	3.81 (5.29)	83.80%	500	4
Matemática I	680	206 / 212 / 262	3.12 (4.38)	86.50%	513	4
Física I	707	272 / 261 / 174	4.13 (4.14)	87.40%	439	4

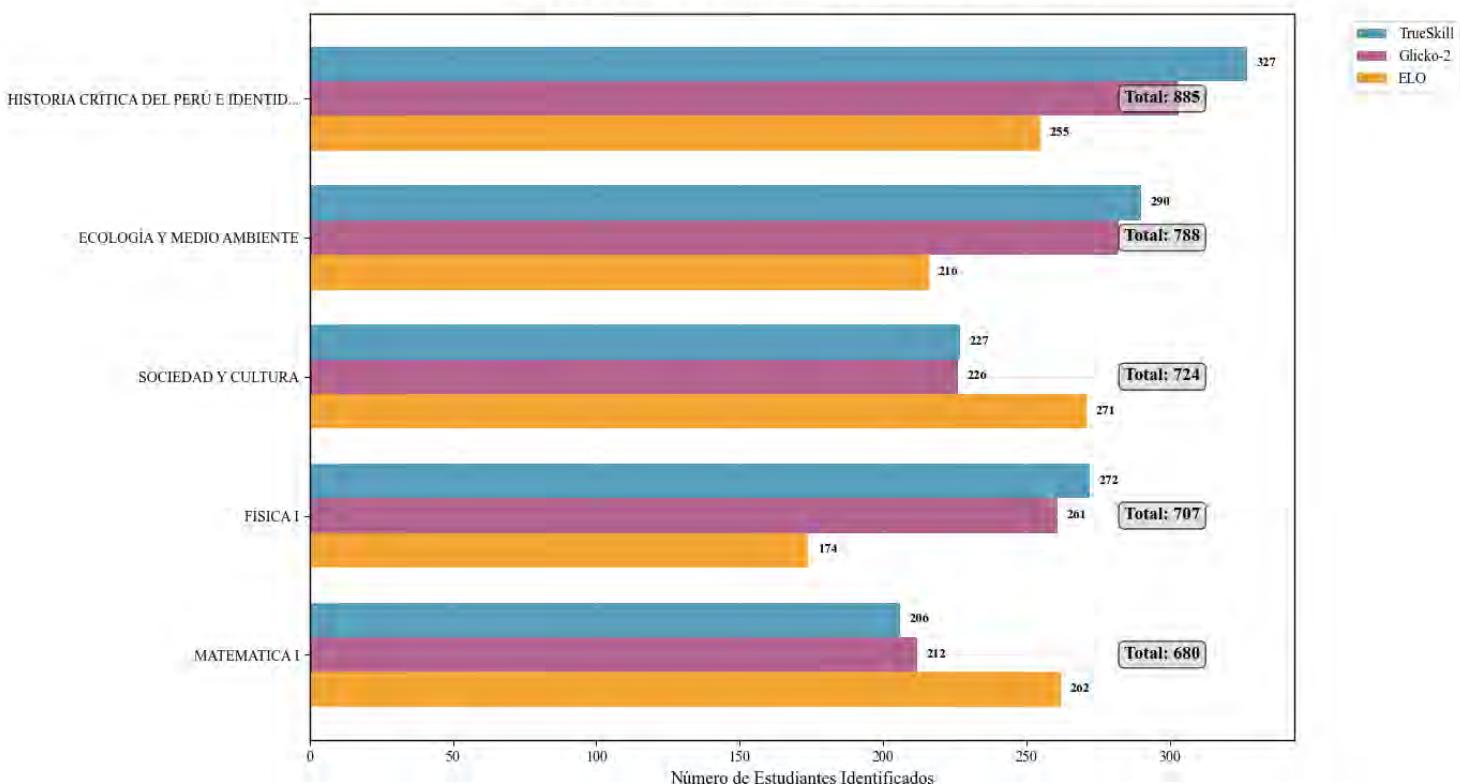
Filosofía y Ética	578	173 / 171 / 234	3.23 (5.43)	84.30%	426	4
Cálculo I	605	234 / 230 / 141	4.00 (4.27)	88.80%	386	4
Redacción de Textos	561	169 / 171 / 221	3.32 (5.10)	85.60%	403	4
Estrategias de Aprendizaje Autónomo	563	167 / 166 / 230	3.35 (5.28)	85.30%	395	4
Química General	590	230 / 223 / 137	4.29 (4.36)	86.60%	340	4

Fuente: Elaboración personal

**Figura 21**

*Top 5 asignaturas más críticas desaprobadas identificados por algoritmos de emparejamiento ELO, Glicko-2 y TrueSkill*

Top 5 Asignaturas más desaprobadas por algoritmo de clasificación



Fuente: Elaboración personal

Por último, se observó la distribución de cursos según el nivel de rendimiento académico y promedio (Tabla 19) lo que revela diferencias significativas en la capacidad de identificación de los tres algoritmos evaluados. El algoritmo ELO demuestra una superioridad significativa en la detección de casos críticos, identificando al 67.6% de estudiantes con promedios inferiores a 5.0 puntos, lo que representa una ventaja de 7.6 y 7.8 puntos porcentuales sobre Glicko-2 (60.8%) y TrueSkill (60.0%) respectivamente. Esto sugiere que ELO posee una mayor sensibilidad para detectar situaciones de fracaso académico extremo, constituyéndose como la herramienta más efectiva para sistemas de alerta temprana que requieren identificación prioritaria de estudiantes en riesgo crítico. Por el contrario, TrueSkill exhibe el perfil más equilibrado con una distribución uniforme a través de las categorías, mientras que Glicko-2 ocupa una posición intermedia con estabilidad diagnóstica moderada.

La distribución porcentual en las categorías intermedias revela estrategias algorítmicas complementarias que responden a diferentes enfoques de identificación. En el rango de casos reprobados (5.0-9.5 puntos), TrueSkill y Glicko-2 mantienen porcentajes similares (18.5% y 18.4% respectivamente), mientras que ELO registra una proporción significativamente menor (14.0%). Esta diferencia de aproximadamente 4.5 puntos porcentuales indica que ELO concentra su capacidad diagnóstica en los extremos de severidad, optimizando su funcionamiento para la detección de casos críticos, pero sacrificando efectividad en rangos intermedios. Para la categoría de casos desaprobados (9.5-13.5 puntos), TrueSkill identifica el mayor porcentaje (8.9%), confirmando su mayor sensibilidad para detectar estudiantes en rangos límitrofes de rendimiento académico. Esta complementariedad algorítmica sugiere la conveniencia de implementar sistemas híbridos que aprovechen las fortalezas específicas de cada algoritmo según el nivel de riesgo a identificar.

**Tabla 19**

Distribución de calificación cursos en alumnos con bajo rendimiento identificados por algoritmos de emparejamiento ELO, Glicko-2 y TrueSkill

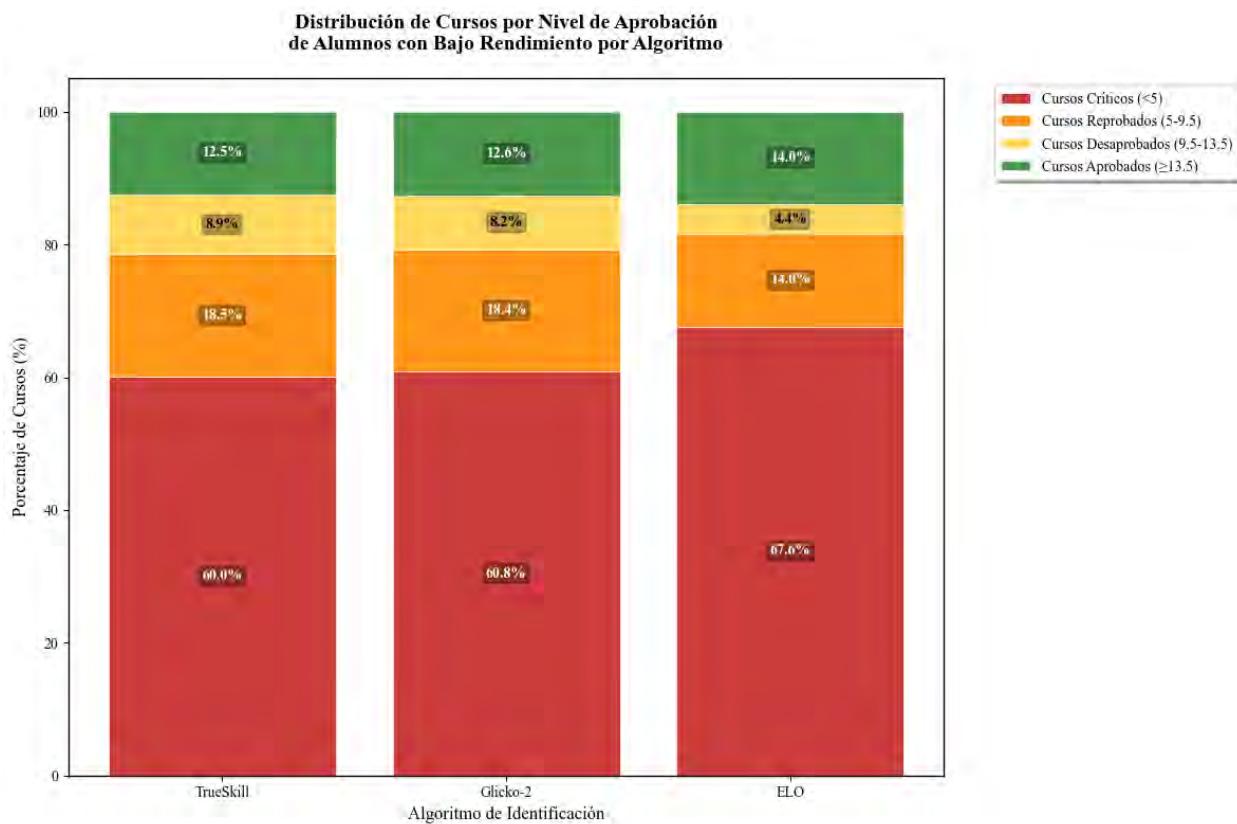
<b>Estado / Algoritmo</b>	<b>TrueSkill</b>	<b>Glicko-2</b>	<b>ELO</b>
<b>Cursos críticos (&lt;5.0)</b>	5635 (60.0%)	5730 (60.8%)	6607 (67.6%)
<b>Cursos reprobados (5.0-9.5)</b>	1740 (18.5%)	1731 (18.4%)	1365 (14.0%)
<b>Cursos desaprobados (9.5-13.5)</b>	839 (8.9%)	772 (8.2%)	432 (4.4%)
<b>Cursos aprobados (≥13.5)</b>	1176 (12.6%)	1187 (12.6%)	1366 (14.0%)
<b>Total, identificados</b>	9390	9420	9770

*Fuente: Elaboración personal*

En el rango de casos reprobados (5.0-9.5 puntos), TrueSkill y Glicko-2 mantienen porcentajes similares (18.5% y 18.4% respectivamente), mientras que ELO registra una proporción significativamente menor (14.0%). Esta diferencia de aproximadamente 4.5 puntos porcentuales indica que ELO concentra su capacidad diagnóstica en los extremos de severidad, optimizando su funcionamiento para la detección de casos críticos, pero sacrificando efectividad en rangos intermedios. Para la categoría de casos desaprobados (9.5-13.5 puntos), TrueSkill identifica el mayor porcentaje (8.9%), confirmando su mayor sensibilidad para detectar estudiantes en rangos limítrofes de rendimiento académico.

**Figura 22**

*Distribución por colores de cursos en alumnos con bajo rendimiento identificados por algoritmos de emparejamiento ELO, Glicko-2 y TrueSkill*



*Fuente: Elaboración personal*

## CONCLUSIONES

- En relación con el objetivo de la investigación, se determinó que los algoritmos de emparejamiento ELO, Glicko-2 y TrueSkill permiten clasificar con efectividad a los estudiantes de la UNSAAC en el periodo postpandemia (2023-I a 2024-II), mostrando correlaciones positivas y estadísticamente significativas con el Cumulative Grade Point Average (CGPA). Los coeficientes de Pearson y Spearman confirmaron la validez de los modelos: para ELO se obtuvo  $r = 0.7970$  y  $\rho = 0.4630$ ; en Glicko-2,  $r = 0.8213$  y  $\rho = 0.7948$ ; y en TrueSkill,  $r = 0.8078$  y  $\rho = 0.7783$ . Estos índices demuestran que sí existe correlación entre las clasificaciones generadas por los algoritmos y el desempeño académico real de los estudiantes, validando la pertinencia de su aplicación en entornos educativos.
- Respecto a la evaluación del algoritmo ELO, se concluye que este modelo, aunque carece de medidas explícitas de incertidumbre, fue el más efectivo en la identificación de casos críticos con promedios inferiores a 5.0, detectando al 67.6 % de los estudiantes en dicha condición. Sin embargo, sacrificó sensibilidad en rangos intermedios, lo que limita su capacidad de discriminación en estudiantes con rendimientos cercanos al promedio.
- En cuanto a la evaluación del algoritmo Glicko-2, se demostró que este alcanzó el mejor balance entre precisión y estabilidad, logrando correlaciones más altas con el CGPA ( $r = 0.82$  y  $\rho = 0.80$ ) y un buen tiempo de ejecución promedio por curso (1.76 cursos por segundo). Por ello, Glicko-2 se posiciona como el algoritmo más robusto y eficiente en término de correlación.
- En relación a la evaluación el algoritmo TrueSkill, los resultados evidenciaron que su enfoque bayesiano secuencial le otorgó una mayor sensibilidad en la detección de estudiantes en rangos

límitrofes (9.5–13.5), alcanzando un 8.9 % de identificación en este segmento, además el mejor tiempo de ejecución promedio por curso (1.86 cursos por segundo) aunque su correlación CGPA ( $r = 0.8078$  y  $\rho = 0.7783$ ) sea un poco menor a Glicko-2. Esto demuestra que TrueSkill es particularmente eficaz en discriminar casos en zonas de incertidumbre académica, aportando un valor diferencial frente a ELO y Glicko-2. Por tanto, TrueSkill se posiciona como el algoritmo más robusto y eficiente en término de tiempo de ejecución.

- Finalmente, comparando los tres los algoritmos: se observó que, aunque los tres identificaron tasas globales de identificación de alumnos de bajo rendimiento (73.5 % a 73.7 %). Un total de 841 alumnos de bajo rendimiento (el 49.4% del total) fueron detectados de manera unánime por los tres algoritmos: ELO mostró la mayor capacidad de identificación exclusiva (310 alumnos, 18.2%), en contraste, TrueSkill y Glicko-2 presentaron una alta coincidencia mutua exclusiva (275 alumnos, 16.2%)
- , cada uno presenta fortalezas complementarias: ELO es más eficaz en casos extremos, TrueSkill es el mejor en determinar rangos intermedios de emparejamiento y en tiempo de ejecución; y Glicko-2 tiene la mejor correlación con respecto CGPA. Además, el análisis de asignaturas mostró que las mayores tasas de desaprobación en alumnos con bajo rendimiento se concentran en cursos de Estudios Generales (Historia Crítica del Perú e Identidad Nacional, Ecología y Medio Ambiente, Matemática I, Física I, Química General), con promedios extremadamente bajos y tasas de reprobación superiores al 85 %, lo que evidencia tanto dificultades en ciencias exactas como en competencias de comprensión lectora y análisis crítico en los primeros semestres. Concluyendo, Glicko-2 y TrueSkill demuestran ser los mejores algoritmos en la detección de estudiantes con bajo rendimiento en el periodo post-covid de la Universidad Nacional de San Antonio Abad del Cusco

## RECOMENDACIONES

- Se recomienda que la UNSAAC y otras universidades integren algoritmos de emparejamiento como ELO, Glicko-2 y TrueSkill en sus sistemas de gestión académica, con el fin de identificar tempranamente a estudiantes en riesgo con bajo rendimiento y facilitar una mejor implementación de tutorías, programas de nivelación y estrategias pedagógicas focalizadas. Dado que los tres modelos presentan fortalezas complementarias —ELO en la detección de casos críticos, TrueSkill en la discriminación de rangos intermedios y Glicko-2 en la robustez global de sus correlaciones—, resulta conveniente explorar sistemas híbridos que combinen sus capacidades para maximizar la precisión diagnóstica y reducir sesgos.
- Asimismo, se recomienda profundizar en la optimización de hiperparámetros para mejorar la estabilidad y precisión de los modelos. En el caso de ELO, el valor de K determina la sensibilidad de las actualizaciones; en TrueSkill, los márgenes de empate y las distribuciones iniciales pueden generar sesgos; mientras que en Glicko-2 los parámetros de volatilidad ( $\sigma$ ) y ajuste ( $\tau$ ) suelen definirse experimentalmente. Por ello, se sugiere realizar pruebas adicionales, análisis de sensibilidad y validaciones cruzadas que permitan perfeccionar su desempeño en distintos contextos académicos.
- Finalmente, se recomienda garantizar la ética y confidencialidad en el uso de estas técnicas, asegurando que la clasificación no se convierta en una etiqueta sancionadora, sino en un insumo para la mejora del aprendizaje y la permanencia estudiantil. Para futuros trabajos, se sugiere ampliar la aplicación de estos algoritmos a otras universidades y niveles educativos,

así como su integración con técnicas de inteligencia artificial y análisis predictivo que permitan ajustar los modelos en tiempo real y fortalecer su implantación en entornos académicos.

## BIBLIOGRAFÍA

- Ahmad, N., Adibah, N., Hasan, A., Aina, N.', Razak, A., Mara, U. T., Kelantan, C., Teknologi, U., Cawangan, M., & Pinang, P. (2017). Factors that Significantly Affect College Students' CGPA Factors that Significantly Affect College Students' CGPA Article Information. In *International Academic Research Journal of Social Science* (Vol. 3, Issue 1). <https://www.researchgate.net/publication/334558121>
- Alberto Caycho-Valencia, F., Norbert Wiener, U., & Perú, L. (2023). *HACIA UN MODELO DE ENSEÑANZA HÍBRIDO TOWARDS A HYBRID TEACHING MODEL*. 38(1), 159–176. <https://orcid.org/0000-0001-8241-5506>
- Amor Pulido, R., Aguilar Peña, C., & Morales Luque, A. (2009). *Inferencia Estadística*. Grupo Editorial Universitario.
- Azlinah, M. W. B., Bee, M., & Yap, W. (2020). *Supervised and Unsupervised Learning for Data Science*. <http://www.springer.com/series/15892>
- Cañihua Florez, D. J., Aguirre Carbajal, D. S., & Huamán Aguilar, C. (2022). *Registro de Graduados y Titulados 2022 - UNSAAC*. [www.unsaac.edu.pe](http://www.unsaac.edu.pe)
- Cañihua Florez, D. J., Huamán Aguilar, C., & Aguirre Carbajal, D. S. (2023). *Compendio Estadístico 2022 N°37 - UNSAAC*.
- Cañihua Florez, D. J., & Meza de Loayza, O. (2018). *Compendio Estadístico 2018 N°33 - UNSAAC*.
- Davidson, R. R. (1970). On extending the Bradley-Terry model to accommodate ties in paired comparison experiments. *Journal of the American Statistical Association*, 65(329), 317-328. <https://doi.org/10.1080/01621459.1970.10481082>
- Dangauthier, P., Herbrich, R., Minka, T., & Graepel, T. (2007). *TrueSkill Through Time: Revisiting the History of Chess*.
- Dehpanah, A., Gemmell, J., Mobasher, B., & Faizan Ghori, M. (2021). *Evaluating Team Skill Aggregation in Online Competitive Games*. <https://www.researchgate.net/publication/352573508>
- García Tárrago, M. J. (2020). Development of a Python algorithm to simulate and analyze the reliability of multiple choice tests to evaluate the student knowledge. *Advances in Building Education*, 4(2), 20. <https://doi.org/10.20868/abe.2020.2.4461>

- Glickman, M. E. (2022). Example of the Glicko-2 system.
- How It Works — Camelot 1.0.9 documentation. (n.d.). <https://camelot-py.readthedocs.io/en/stable/user/how-it-works.html>
- Instituto Nacional de Estadística e Informática. (2023). *Cusco Compendio Estadístico 2023*.
- Kekare, A., Jachak, A., Gosavi, A., Hanwate, S., & Student, B. E. (2020). Techniques for Detecting and Extracting Tabular Data from PDFs and Scanned Documents: A Survey. *International Research Journal of Engineering and Technology*. [www.irjet.net](http://www.irjet.net)
- Youngjin, L. (2019). Estimating student ability and problem difficulty using item response theory (IRT) and TrueSkill. *Information Discovery and Delivery*, 47(2), 67–75. <https://doi.org/10.1108/IDD-08-2018-0030>
- Ministerio de Educación Peruano, & Diario El Peruano. (2020). *Decreto Supremo N° 012-2020-MINEDU: Aprueban la Política Nacional de Educación Superior y Técnico-Productiva*.
- Reddick, R. (2019). *Using a Glicko-based Algorithm to Measure In-Course Learning*.
- Ruiperez-Valiente, J. A., Kim, Y. J., Baker, R. S., Martinez, P. A., & Lin, G. C. (2023). The Affordances of Multivariate Elo-Based Learner Modeling in Game-Based Assessment. *IEEE Transactions on Learning Technologies*, 16(2), 152–165. <https://doi.org/10.1109/TLT.2022.3203912>
- Szczecinski, L., & Djebbi, A. (2020). Understanding draws in Elo rating algorithm. *Journal of Quantitative Analysis in Sports*, 16(3), 211–220. <https://doi.org/10.1515/jqas-2019-0102>
- Vesin, B., Mangaroska, K., Akhuseyinoglu, K., & Giannakos, M. (2022). Adaptive Assessment and Content Recommendation in Online Programming Courses: On the Use of Elo-rating. *ACM Transactions on Computing Education*, 22(3). <https://doi.org/10.1145/3511886>
- VICERRECTORADO ACADEMICO UNSAAC. (2017). *RESOLUCIÓN NRO. CU-093-2017-UNSAAC*.
- Wild, C. (2006). The Concept of Distribution. *STATISTICS EDUCATION RESEARCH JOURNAL*, 5(2), 10–26. <https://doi.org/10.52041/serj.v5i2.497>
- What is Cumulative GPA? How do you calculate it? (2024, 25 de marzo). *CollegeVine Blog*. <https://blog.collegevine.com/what-is-cumulative-gpa/>
- Zafari, M., Sadeghi-Niaraki, A., Choi, S. M., & Esmaeily, A. (2021). A Practical Model for the Evaluation of High School Student Performance Based on Machine Learning. *Applied Sciences (Switzerland)*, 11(23). <https://doi.org/10.3390/app112311534>

## ANEXOS

**Figura 23**

*Anexo 1: Solicitud de acceso a registros académicos dirigida al Centro de Cómputo*

Cusco, 27 de marzo del 2025

**Ing. Aguedo Huamani Huayhua**  
 Director de la Unidad de Centro de Cómputo  
 Universidad Nacional de San Antonio Abad del Cusco

**Asunto: Solicitud Datos de Estudiantes para Trabajo de Investigación de Tesis**

Estimado Ingeniero Aguedo Huamani, reciba un cordial saludo. Yo, Sebastian Israel Macedo Gheiler, egresado de la Universidad Nacional de San Antonio Abad del Cusco (UNSAAC) en el semestre 2024-I. Actualmente, me encuentro desarrollando mi trabajo de investigación para la elaboración de mi Tesis de grado, titulada: **"Modelos de Algoritmos de Emparejamiento Aplicados en la Detección de Estudiantes con Bajo Rendimiento en el Período Post-COVID de la Universidad Nacional de San Antonio Abad del Cusco"**.

Para el desarrollo de este estudio, solicito amablemente el acceso a los registros académicos de los estudiantes desde el semestre **2022-II hasta el 2024-II**, asegurando en todo momento el tratamiento **anónimo** de los datos personales, en cumplimiento con la normativa peruana de protección de datos sensibles.

**Datos requeridos:**

- Cod\_estudiante (código anonimizado)
- Carrera Profesional
- Semestre Académico
- Nombre del Curso
- Notas Parciales
- Promedio Final

Agradeceré que la información sea proporcionada en formato **CSV**, de manera estructurada y en cumplimiento de las políticas institucionales de manejo de datos.

Quedo atento a su pronta respuesta y agradezco de antemano su apoyo en el desarrollo de esta investigación.

Atentamente,



**Sebastian Israel Macedo Gheiler**

DNI: 72796109

Teléfono: 984711851

Correo: 164243@unsaac.edu.pe

## Figura 24

Anexo 2: Pseudocódigo de implementación de algoritmo ELO por Alumno

**Algorithm 1** Sistema de Ranking ELO para calcular Rendimiento por Alumno

---

```

1: procedure CALCULARPUNTAJEESPERADO( $R_A, R_B$ )
2:    $E_A \leftarrow \frac{1}{1+10^{(R_B-R_A)/400}}$ 
3:   return  $E_A$ 
4: end procedure
5: procedure CALCULARNOTATOTAL(estudiante, columnas_notas)
6:   parciales  $\leftarrow$  columnas sin sufijo  $_S_-$ 
7:   sustitutorios  $\leftarrow$  columnas con sufijo  $_S_-$ 
8:   notas_validas  $\leftarrow \{\}$ 
9:   for cada parcial en parciales do
10:    if estudiante[parcial] no es nulo then
11:      agregar estudiante[parcial] a notas_validas
12:    end if
13:   end for
14:   if notas_validas está vacío then return NaN
15:   end if
16:   suma_parciales  $\leftarrow \sum \text{notas\_validas}$ 
17:   bonificacion  $\leftarrow 0$ 
18:   for cada sustitutorio en sustitutorios do
19:    if estudiante[sustitutorio] no es nulo then
20:      bonificacion  $\leftarrow \text{bonificacion} + \text{estudiante}[\text{sustitutorio}] \times 0,3$ 
21:    end if
22:   end for
23:   nota_total  $\leftarrow \frac{\text{suma_parciales}+\text{bonificacion}}{|\text{notas\_validas}|}$ 
24:   return nota_total
25: end procedure
```

## Figura 25

Anexo 3: Pseudocódigo de implementación de algoritmo ELO por Curso y Ejecución

---

**Algorithm 2** Sistema de Ranking ELO por Curso y Ejecución

---

```

1: procedure CALCULARELOCURSO(datos_curso, id_estudiante, columnas_notas)
2:   Calcular nota_total para cada estudiante
3:   Eliminar estudiantes sin nota_total válida
4:   if cantidad de estudiantes < 2 then
5:     Asignar ELO_inicial a todos
6:   return datos con ELOs iniciales
7: end if
8: desviacion  $\leftarrow$  desviación estándar de notas_totales
9: K  $\leftarrow K_{base} \times \frac{\min(desviacion\_5)}{5}$ 
10: Inicializar todos los ELOs con valor inicial
11: for cada estudiante i en el curso do
12:   puntos_reales  $\leftarrow 0$ , puntos Esperados  $\leftarrow 0$ , ponentes  $\leftarrow 0$ 
13:   for cada estudiante j  $\neq i$  en el curso do
14:     diferencia  $\leftarrow nota_i - nota_j$ 
15:     if diferencia  $> 0$  then
16:       resultado_real  $\leftarrow 0.5 + \frac{diferencia}{40}$ 
17:     else if diferencia  $< 0$  then
18:       resultado_real  $\leftarrow 0.5 - \frac{|diferencia|}{40}$ 
19:     else
20:       resultado_real  $\leftarrow 0.5$ 
21:     end if
22:     resultado Esperado  $\leftarrow$  CALCULARPUNTAJEESPERADO(ELOi, ELOj)
23:     puntos_reales  $\leftarrow$  puntos_reales + resultado_real
24:     puntos Esperados  $\leftarrow$  puntos Esperados + resultado Esperado
25:     ponentes  $\leftarrow$  ponentes + 1
26:   end for
27:   if ponentes  $> 0$  then
28:     cambio  $\leftarrow K \times \frac{puntos\_reales - puntos\_esperados}{ponentes}$ 
29:     ELOi  $\leftarrow ELO_i + cambio$ 
30:   end if
31: end for
32: return datos con ELOs actualizados
33: end procedure
34: procedure EJECUTARALGORITMOELO(datos_notas, Kbase, ELO_inicial)
35:   Identificar y agrupar columnas por código de curso
36:   cursos_validos  $\leftarrow \{\}$ 
37:   for cada curso en cursos identificados do
38:     Calcular estudiantes con notas válidas
39:     if estudiantes válidos  $\geq$  mínimo requerido then
40:       Agregar curso a cursos_validos
41:     end if
42:   end for
43:   resultados  $\leftarrow \{\}$ 
44:   for cada curso en cursos_validos do
45:     datos_curso  $\leftarrow$  extraer datos del curso actual
46:     elo_curso  $\leftarrow$  CALCULARELOCURSO(datos_curso)
47:     Agregar metadatos del curso
48:     Agregar elo_curso a resultados
49:   end for
50:   if resultados está vacío then return DataFrames vacíos
51:   end if
52:   Consolidar todos los resultados en dataset unificado
53:   Calcular estadísticas agregadas por estudiante
54:   Ordenar ranking por ELO promedio descendente
55:   return ranking global, datos detallados por curso
56: end procedure

```

---

## Figura 26

Anexo 4: Pseudocódigo de implementación de algoritmo Glicko-2 adaptativo por Alumno

---

### Algorithm 1 Sistema de Ranking Glicko-2 Adaptativo para calcular Rendimiento por Alumno

---

```

1: procedure CALCULARPARAMETROSADAPTATIVOS(datos_curso, columnas_notas)
2:   Calcular nota_promedio y nota_std del curso
3:    $\mu \leftarrow \text{nota\_promedio} \times 100$ 
4:    $\sigma \leftarrow \max(200, \min(500, \text{nota\_std} \times 30))$ 
5:   volatilidad  $\leftarrow \max(0,04, \min(0,1, \text{nota\_std} \times 0,005))$ 
6:   return  $\mu, \sigma, \text{volatilidad}$ 
7: end procedure
8: procedure REDUCIRIMPACTO(rating)
9:   return  $\frac{1}{\sqrt{1 + \frac{3 \times \text{rating} \cdot \sigma^2}{\pi^2}}}$ 
10: end procedure
11: procedure CALCULARPUNTAJEESPERADO(rating, otro_rating, impacto)
12:   return  $\frac{1}{1 + e^{-\text{impacto} \times (\text{rating} - \mu - \text{otro\_rating} - \mu)}}$ 
13: end procedure
14: procedure DETERMINARVOLATILIDAD(rating, diferencia, varianza)
15:    $a \leftarrow \ln(\text{rating.volatilidad}^2)$ 
16:   Definir función  $f(x)$  para algoritmo Illinois
17:    $A \leftarrow a$ ,  $B \leftarrow$  valor inicial según condiciones
18:   while  $|B - A| > \epsilon$  do
19:      $C \leftarrow A + (A - B) \times f(A)/(f(B) - f(A))$ 
20:     Actualizar  $A, B$  según algoritmo Illinois
21:   end while
22:   return  $e^{A/2}$ 
23: end procedure
24: procedure ACTUALIZARRATING(rating, serie)
25:   varianza_inv  $\leftarrow 0$ , diferencia  $\leftarrow 0$ 
26:   for cada (puntaje, oponente) en serie do
27:     impacto  $\leftarrow$  REDUCIRIMPACTO(oponente)
28:     esperado  $\leftarrow$  CALCULARPUNTAJEESPERADO(rating, oponente, impacto)
29:     varianza_inv  $\leftarrow$  varianza_inv + impacto2  $\times$  esperado  $\times (1 - \text{esperado})$ 
30:     diferencia  $\leftarrow$  diferencia + impacto  $\times (\text{puntaje} - \text{esperado})$ 
31:   end for
32:   if varianza_inv = 0 then
33:      $\sigma_{\text{nuevo}} \leftarrow \sqrt{\text{rating} \cdot \sigma^2 + \text{rating.volatilidad}^2}$ 
34:   return nuevo rating con  $\sigma_{\text{nuevo}}$ 
35:   end if
36:   varianza  $\leftarrow 1,0 / \text{varianza\_inv}$ 
37:   diferencia  $\leftarrow \text{diferencia} \times \text{varianza}$ 
38:   vol_nueva  $\leftarrow$  DETERMINARVOLATILIDAD(rating, diferencia, varianza)
39:    $\sigma_{\text{estrella}} \leftarrow \sqrt{\text{rating} \cdot \sigma^2 + \text{vol}_n^2}$ 
40:    $\sigma_{\text{nuevo}} \leftarrow 1,0 / \sqrt{1,0 / \sigma_{\text{estrella}}^2 + 1,0 / \text{varianza}}$ 
41:    $\mu_{\text{nuevo}} \leftarrow \text{rating} \cdot \mu + \sigma_{\text{nuevo}}^2 \times (\text{diferencia} / \text{varianza})$ 
42:   return nuevo rating con  $\mu_{\text{nuevo}}, \sigma_{\text{nuevo}}, \text{vol}_{\text{nueva}}$ 
43: end procedure

```

---

## Figura 27

Anexo 5: Pseudocódigo de implementación de algoritmo Glicko-2 por Curso y Ejecución

---

**Algorithm 2** Sistema Glicko-2 Adaptativo por Curso y Ejecución
 

---

```

1: procedure CALCULARGLICKO2CURSO(datos_curso, id_estudiante, columnas_notas)
2:   Calcular nota_total para cada estudiante
3:   Eliminar estudiantes sin nota válida
4:   if cantidad estudiantes < 2 then return datos iniciales
5:   end if
6:    $\mu, \sigma, vol \leftarrow$  CALCULARPARAMETROSADAPTATIVOS(datos_curso, columnas_notas)
7:   Inicializar sistema Glicko-2 con parámetros adaptativos
8:   ratings  $\leftarrow \{\}$ ; nota_promedio  $\leftarrow$  promedio del curso
9:   for cada estudiante en el curso do
10:    ajuste  $\leftarrow (nota - nota\_promedio) \times 20$ 
11:     $\mu_{ajustado} \leftarrow \mu + ajuste$ 
12:    Inicializar rating con  $\mu_{ajustado}, \sigma, vol$ 
13:   end for
14:   n  $\leftarrow$  cantidad de estudiantes; nota_std  $\leftarrow$  desviación estándar
15:   for cada estudiante i en el curso do
16:     serie  $\leftarrow []$ 
17:     for cada estudiante j  $\neq i$  en el curso do
18:       diferencia  $\leftarrow |nota_i - nota_j|$ 
19:       umbral  $\leftarrow$  máx(0,3, nota_std  $\times$  0,1)
20:       if diferencia < umbral then resultado  $\leftarrow$  EMPATE
21:       else if notai > notaj then resultado  $\leftarrow$  VICTORIA
22:       else resultado  $\leftarrow$  DERRROTA
23:       end if
24:       Agregar (resultado, ratingj) a serie
25:     end for
26:     if serie no está vacía then
27:       ratings[i]  $\leftarrow$  ACTUALIZARRATING(ratings[i], serie)
28:     end if
29:   end for
30:   Asignar ratings finales a datos del curso
31:   return datos con ratings Glicko-2
32: end procedure
33: procedure EJECUTARALGORITMOGLICKO2(datos_notas)
34:   Identificar y agrupar cursos por código
35:   cursos_validos  $\leftarrow \{\}$ 
36:   for cada curso en cursos identificados do
37:     if estudiantes válidos  $\geq$  mínimo requerido then
38:       Agregar curso a cursos_validos
39:     end if
40:   end for
41:   resultados  $\leftarrow []$ 
42:   for cada curso en cursos_validos do
43:     datos_curso  $\leftarrow$  extraer datos del curso
44:     resultado_curso  $\leftarrow$  CALCULARGLICKO2CURSO(datos_curso)
45:     Agregar metadatos del curso
46:     Agregar resultado_curso a resultados
47:   end for
48:   if resultados vacío then return DataFrame vacío
49:   end if
50:   Consolidar todos los resultados
51:   Calcular estadísticas agregadas por estudiante
52:   Ordenar ranking por rating promedio descendente
53:   return ranking global
54: end procedure
  
```

---

## Figura 28

Anexo 6: Pseudocódigo de implementación de algoritmo TrueSkill TTT-D por Alumno

**Algorithm 1** Sistema de Ranking TTT-D (TrueSkill Through Time con Draw Margins) para calcular Rendimiento por Alumno

---

```

1: procedure CALCULARPARAMETROSADAPTATIVOSTT(Datos.curso, columnas.notas)
2:   Calcular nota_promedio y nota_std del curso
3:    $\mu \leftarrow \text{nota\_promedio} \times 100$ 
4:    $\sigma \leftarrow \max(150, \min(400, \text{nota\_std} \times 25))$ 
5:    $\beta \leftarrow \max(100, \min(300, 100 + \text{nota\_std} \times 15))$ 
6:   margen_empate  $\leftarrow \max(0.2, \min(1.5, \text{nota\_std}/\text{nota\_promedio}))$ 
7:   return  $\mu, \sigma, \beta, \text{margen\_empate}$ 
8: end procedure
9: procedure CALCULARMARGENEMPATEINDIVIDUAL(nota, nota_promedio, nota_std, margen_curso)
10:  if nota_std  $\leq 0$  then return margen_curso
11:  end if
12:   $z \leftarrow |\text{nota} - \text{nota\_promedio}|/\text{nota\_std}$ 
13:  factor  $\leftarrow 1 + 0.5 \times e^{-z}$ 
14:  margen_individual  $\leftarrow \text{margen\_curso} \times \text{factor}$ 
15:  return  $\max(0.1, \min(2.0, \text{margen\_individual}))$ 
16: end procedure
17: procedure CALCULARMARGENEMPATEEFFECTIVO(rating1, rating2, diferencia_rendimiento)
18:  margen_combinado  $\leftarrow (\text{rating1.margen} + \text{rating2.margen})/2$ 
19:  factor_incertidumbre  $\leftarrow (\text{rating1}\sigma + \text{rating2}\sigma)/(2 \times \sigma_{\text{sistema}})$ 
20:  ajuste_rendimiento  $\leftarrow \max(0, \text{margen\_combinado} - |\text{diferencia\_rendimiento}| \times 0.1)$ 
21:  return  $\max(0.1, \text{margen\_combinado} \times \text{factor\_incertidumbre} + \text{ajuste\_rendimiento})$ 
22: end procedure
23: procedure CALCULARPROBABILADESTT(rating1, rating2, diferencia_rendimiento)
24:   $\Delta\mu \leftarrow \text{rating1}\mu - \text{rating2}\mu$ 
25:   $c^2 \leftarrow 2 \times \beta^2 + \text{rating1}\sigma^2 + \text{rating2}\sigma^2$ 
26:   $c \leftarrow \sqrt{c^2}$ 
27:   $\epsilon \leftarrow \text{CALCULARMARGENEMPATEEFFECTIVO}(\text{rating1}, \text{rating2}, \text{diferencia\_rendimiento})$ 
28:   $P_{\text{ganar}} \leftarrow \Phi((\Delta\mu - \epsilon)/c)$ 
29:   $P_{\text{empate}} \leftarrow \Phi((\Delta\mu + \epsilon)/c) - \Phi((\Delta\mu - \epsilon)/c)$ 
30:   $P_{\text{perder}} \leftarrow 1 - \Phi((\Delta\mu + \epsilon)/c)$ 
31:  Normalizar probabilidades para suma = 1
32:  return  $P_{\text{ganar}}, P_{\text{empate}}, P_{\text{perder}}$ 
33: end procedure
34: procedure ACTUALIZARATINGTTD(rating, oponente, resultado, diferencia_rendimiento)
35:   $c^2 \leftarrow 2 \times \beta^2 + \text{rating}\sigma^2 + \text{oponente}\sigma^2$ 
36:   $c \leftarrow \sqrt{c^2}$ 
37:   $\Delta\mu \leftarrow \text{rating}\mu - \text{oponente}\mu$ 
38:   $\epsilon \leftarrow \text{CALCULARMARGENEMPATEEFFECTIVO}(\text{rating}, \text{oponente}, \text{diferencia\_rendimiento})$ 
39:  Calcular factores v y w según resultado:
40:  case resultado of
41:    VICTORIA:  $v \leftarrow \varphi(\text{arg})/\Phi(\text{arg}), w \leftarrow v \times (v + \text{arg})$ 
42:    DERROTA:  $v \leftarrow -\varphi(\text{arg})/\Phi(\text{arg}), w \leftarrow v \times (v + \text{arg})$ 
43:    EMPATE:  $v \leftarrow (\varphi_{\text{inf}} - \varphi_{\text{sup}})/(\Phi_{\text{sup}} - \Phi_{\text{inf}})$ 
44:  end case
45:   $\mu_{\text{nuevo}} \leftarrow \text{rating}\mu + (\text{rating}\sigma^2/c) \times v$ 
46:   $\sigma_{\text{nuevo}}^2 \leftarrow \text{rating}\sigma^2 \times \max(1 - (\text{rating}\sigma^2/c^2) \times w, 0.1)$ 
47:  factor_rendimiento  $\leftarrow 1 + |\text{diferencia\_rendimiento}| \times 0.05$ 
48:  margen_nuevo  $\leftarrow \text{rating}\text{margen} \times (0.95 + 0.05/\text{factor\_rendimiento})$ 
49:  margen_nuevo  $\leftarrow \max(0.1, \min(2.0, \text{margen\_nuevo}))$ 
50:  return nuevo rating con  $\mu_{\text{nuevo}}, \sqrt{\sigma_{\text{nuevo}}^2}, \text{margen\_nuevo}$ 
51: end procedure
52: procedure ACTUALIZARENFRENTAMIENTO1VS1(rating1, rating2, resultado, diferencia_rendimiento)
53:  nuevo_rating1  $\leftarrow \text{ACTUALIZARATINGTTD}(\text{rating1}, \text{rating2}, \text{resultado}, \text{diferencia\_rendimiento})$ 
54:  if resultado = VICTORIA then resultado2  $\leftarrow$  DERROTA
55:  else if resultado = DERROTA then resultado2  $\leftarrow$  VICTORIA
56:  else resultado2  $\leftarrow$  EMPATE end if
57:  nuevo_rating2  $\leftarrow \text{ACTUALIZARATINGTTD}(\text{rating2}, \text{rating1}, \text{resultado2}, \text{diferencia\_rendimiento})$ 
58:  return nuevo_rating1, nuevo_rating2
59: end procedure

```

---

## Figura 29

Anexo 7: Pseudocódigo de implementación de algoritmo TrueSkill TTT-D por Curso y Ejecución

---

### Algorithm 2 Sistema TTT-D por Curso y Ejecución

---

```

1: procedure CALCULARTTTDCURSO(datos_curso, id_estudiante, columnas_notas)
2:   Calcular nota_total para cada estudiante
3:   Eliminar estudiantes sin nota válida
4:   if cantidad estudiantes < 2 then return datos iniciales
5:   end if
6:    $\mu, \sigma, \beta, \text{margen} \leftarrow \text{CALCULARPARAMETROSADAPTATIVOSTTTD}(\text{datos\_curso}, \text{columnas\_notas})$ 
7:   Inicializar sistema TTT-D con parámetros adaptativos
8:   ratings  $\leftarrow \{\}$ ; nota_promedio  $\leftarrow$  promedio; nota_std  $\leftarrow$  desviación
9:   for cada estudiante en el curso do
10:    ajuste  $\leftarrow (\text{nota} - \text{nota\_promedio}) \times 15$ 
11:     $\mu_{\text{ajustado}} \leftarrow \mu + \text{ajuste}$ 
12:    margen_individual  $\leftarrow \text{CALCULARMARGENEMPATEINDIVIDUAL}(\text{nota}, \text{nota\_promedio}, \text{nota\_std}, \text{margen})$ 
13:    Inicializar rating con  $\mu_{\text{ajustado}}, \sigma, \text{margen\_individual}$ 
14:   end for
15:   n  $\leftarrow$  cantidad de estudiantes
16:   for i  $\leftarrow 0$  to n – 1 do
17:    for j  $\leftarrow i + 1$  to n – 1 do
18:      diferencia  $\leftarrow \text{nota}_i - \text{nota}_j$ ; diff_abs  $\leftarrow |\text{diferencia}|$ 
19:      umbral  $\leftarrow (\text{rating}_i.\text{margen} + \text{rating}_j.\text{margen})/2 \times (1 + \text{nota\_std} \times 0,1)$ 
20:      if diff_abs < umbral then resultado  $\leftarrow \text{EMPATE}$ 
21:      else if diferencia > 0 then resultado  $\leftarrow \text{VICTORIA}$ 
22:      else resultado  $\leftarrow \text{DERROTA}$ 
23:      end if
24:      ratingi, ratingj  $\leftarrow \text{ACTUALIZARENFRENTAMIENTO1VS1}(\text{rating}_i, \text{rating}_j, \text{resultado}, \text{diff\_abs})$ 
25:    end for
26:   end for
27:   Asignar ratings finales a datos del curso
28:   return datos con ratings TTT-D
29: end procedure
30: procedure EJECUTARALGORITMOTTTD(datos_notas)
31:   Identificar y agrupar cursos por código
32:   cursos_validos  $\leftarrow \{\}$ 
33:   for cada curso en cursos identificados do
34:     if estudiantes válidos  $\geq$  mínimo requerido then
35:       Agregar curso a cursos_validos
36:     end if
37:   end for
38:   resultados  $\leftarrow []$ 
39:   for cada curso en cursos_validos do
40:     datos_curso  $\leftarrow$  extraer datos del curso
41:     resultado_curso  $\leftarrow \text{CALCULARTTTDCURSO}(\text{datos\_curso})$ 
42:     Agregar metadatos del curso
43:     Agregar resultado_curso a resultados
44:   end for
45:   if resultados vacío then return DataFrame vacío
46:   end if
47:   Consolidar todos los resultados
48:   Calcular estadísticas agregadas por estudiante
49:   Ordenar ranking por rating promedio descendente
50:   return ranking global TTT-D
51: end procedure

```

---