UNIVERSIDAD NACIONAL PEDRO RUIZ GALLO

FACULTAD DE INGENIERÍA CIVIL, DE SISTEMAS Y DE ARQUITECTURA

ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS

TESIS

Sistema tutor inteligente para apoyar el aprendizaje de matemáticas del grado tercero de secundaria en la institución educativa peruano canadiense.

PARA OPTAR EL TÍTULO PROFESIONAL DE:

Ingeniero de sistemas

Autores

Bach. Maldonado Vasquez Alexei Billy

Bach. Vasquez Vasquez Braulio Walter

Asesor

Ma. Ing. Arteaga Lora Roberto Carlos

Lambayeque, Perú

2025

UNIVERSIDAD NACIONAL PEDRO RUIZ GALLO

FACULTAD DE INGENIERÍA CIVIL, DE SISTEMAS Y DE ARQUITECTURA

ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS

TESIS

Sistema tutor inteligente para apoyar el aprendizaje de matemáticas del grado tercero de secundaria en la institución educativa peruano canadiense.

PARA OPTAR EL TÍTULO DE INGENIERO DE SISTEMAS

APROBADO POR EL SIGUIENTE JURADO:

DR. ING. VILLEGAS CUBAS JUAN ELÍAS
PRESIDENTE

MSC. ING. OMAR WILTON SAAVEDRA SALAZAR
SECRETARIO

MSC. ING. OSCAR EFRAIN CAPUÑAY UCEDA
VOCAL

MSC. ING. ROBERTO CARLOS ARTEAGA LORA
ASESOR

UNIVERSIDAD NACIONAL PEDRO RUIZ GALLO
FACULTAD DE INGENIERÍA CIVIL DE SISTEMAS Y DE ARQUITECTURA
UNIDAD DE INVESTIGACIÓN



ACTA DE SUSTENTACIÓN N°623-2025-UI-FICSA



Siendo las 12:30 pm horas del día 17 de diciembre del 2025, se reunieron de manera presencial los miembros de jurado de la tesis titulada: **"SISTEMA TUTOR INTELIGENTE PARA APOYAR EL APRENDIZAJE DE MATEMÁTICAS DEL GRADO TERCERO DE SECUNDARIA EN LA INSTITUCIÓN EDUCATIVA PERUANO CANADIENSE", con código N° IS_V_2025_007,** designado por **Resolución Decanal N° 261-2025-UNPRG-FICSA** con la finalidad de Evaluar y Calificar la sustentación de la tesis antes mencionada, conformado por los siguientes docentes:

DR. ING. VILLEGAS CUBAS JUAN ELÍAS	**PRESIDENTE**
MSC. ING. OMAR WILTON SAAVEDRA SALAZAR	**SECRETARIO**
MSC. ING. OSCAR EFRAIN CAPUÑAY UCEDA	**VOCAL**

Asesorado por MSC. ING. ROBERTO CARLOS ARTEAGA LORA.

El acto de sustentación fue autorizado por **OFICIO VIRTUAL N° 299-2025-UIFICSA**, la tesis fue presentada y sustentada por los Bachilleres: **MALDONADO VASQUEZ ALEXEI BILLY Y VASQUEZ VASQUEZ BRAULIO WALTER,** tuvo una duración de 60 minutos. Después de la sustentación, y absueltas las preguntas y observaciones de los miembros del jurado; se procedió a la calificación respectiva:

	NUMERO	LETRAS	CALIFICATIVO
MALDONADO VASQUEZ ALEXEI BILLY	18	DIECIOCHO	MUY BUENO
VASQUEZ VASQUEZ BRAULIO WALTER	18	DIECIOCHO	MUY BUENO

Por lo que quedan **APTOS** para obtener el Título Profesional de **INGENIERO DE SISTEMAS** de acuerdo con la Ley Universitaria 30220 y la normatividad vigente de la Facultad de Ingeniería Civil De Sistemas y de Arquitectura de la Universidad Nacional Pedro Ruiz Gallo.

Siendo las 13:30. Se dio por concluido el presente acto académico, dándose conformidad al presente acto, con la firma de los miembros del jurado.

DR. ING. VILLEGAS CUBAS JUAN ELÍAS
PRESIDENTE

MSC. ING. OMAR WILTON SAAVEDRA SALAZAR
SECRETARIO

MSC. ING. OSCAR EFRAIN CAPUÑAY UCEDA
VOCAL

MSC. ING. ROBERTO CARLOS ARTEAGA LORA
ASESOR

UNIVERSIDAD NACIONAL PEDRO RUIZ GALLO
FACULTAD DE INGENIERÍA CIVIL, DE SISTEMAS Y DE ARQUITECTURA
UNIDAD DE INVESTIGACIÓN

"Año de la recuperación y consolidación de la economía peruana"

CONSTANCIA DE VERIFICACIÓN DE ORIGINALIDAD DE TESIS

Según Res. N° 659-2020-R

Yo, ARTEAGA LORA ROBERTO CARLOS, asesor de la Tesis de los bachilleres:

MALDONADO VASQUEZ ALEXEI BILLY
VASQUEZ VASQUEZ BRAULIO WALTER

TITULADA:

"SISTEMA TUTOR INTELIGENTE PARA APOYAR EL APRENDIZAJE DE MATEMÁTICAS DEL GRADO TERCERO DE SECUNDARIA EN LA INSTITUCIÓN EDUCATIVA PERUANO CANADIENSE"

Luego de la revisión exhaustiva del documento constato que la misma tiene un índice de similitud de 7% verificable en el reporte de similitud del programa TURNITIN.

El suscrito analizó dicho reporte y concluyó que, cada una de las coincidencias detectadas NO CONSTITUYEN PLAGIO. A mi leal saber y entender, la tesis cumple con todas las normas para el uso de citas y referencias establecidas por la Universidad Nacional Pedro Ruíz Gallo. Se expide la presente según lo dispuesto en la Resolución N° 659-2020-R, de fecha 8 de setiembre de 2020, que aprueba la directiva para la evaluación de originalidad de los documentos académicos, de investigación formativa y para la obtención de Grados y Títulos de la UNPRG.

Lambayeque, 31 de octubre de 2025

Atentamente,

MA. ING. ROBERTO CARLOS ARTEAGA LORA
DNI 16755764

Se adjunta:

- Recibo digital
- Revisión de informe en Turnitin

MA. ING. ROBERTO CARLOS ARTEAGA LORA
ASESOR
DNI 16755764

UNIVERSIDAD NACIONAL PEDRO RUIZ GALLO
FACULTAD DE INGENIERÍA CIVIL, DE SISTEMAS Y DE ARQUITECTURA
UNIDAD DE INVESTIGACIÓN

SISTEMA TUTOR INTELIGENTE PARA APOYAR EL APRENDIZAJE DE MATEMÁTICAS DEL GRADO TERCERO DE SECUNDARIA EN LA INSTITUCIÓN EDUCATIVA PERUANO CANADIENSE

INFORME DE ORIGINALIDAD

7%	7%	3%	3%
INDICE DE SIMILITUD	FUENTES DE INTERNET	PUBLICACIONES	TRABAJOS DEL ESTUDIANTE

FUENTES PRIMARIAS

1	hdl.handle.net Fuente de Internet	1%
2	repositorio.unprg.edu.pe Fuente de Internet	1%
3	revistas.udistrital.edu.co Fuente de Internet	<1%
4	Submitted to Universidad Cesar Vallejo Trabajo del estudiante	<1%
5	tesis.usat.edu.pe Fuente de Internet	<1%
6	juandomingofarnos.wordpress.com Fuente de Internet	<1%
7	Submitted to Universidad TecMilenio Trabajo del estudiante	<1%
8	repositorio.uta.edu.ec Fuente de Internet	<1%
9	docplayer.es Fuente de Internet	<1%
10	Submitted to Moodle ISFODOSU Trabajo del estudiante	<1%
11	Zulma Cataldi, Fernando J. Lage. "Sistemas tutores inteligentes orientados a la enseñanza para la comprensión", Edutec. Revista Electrónica de Tecnología Educativa, 2009 Publicación	<1%

MA. ING. ROBERTO CARLOS ARTEAGA LORA
DNI 16755764

turnitin

Recibo digital

Este recibo confirma que su trabajo ha sido recibido por Turnitin. A continuación podrá ver la información del recibo con respecto a su entrega.

La primera página de tus entregas se muestra abajo.

Autor de la entrega:	Alexei Billy Maldonado Vasquez
Título del ejercicio:	Informe de Tesis
Título de la entrega:	SISTEMA TUTOR INTELIGENTE PARA APOYAR EL APRENDIZAJE ...
Nombre del archivo:	INFORME_FINAL_ALEXEI.docx
Tamaño del archivo:	2.32M
Total páginas:	97
Total de palabras:	15,115
Total de caracteres:	90,457
Fecha de entrega:	31-oct-2025 05:10p.m. (UTC-0500)
Identificador de la entrega:	2771045886



UNIVERSIDAD NACIONAL PEDRO RUIZ GALLO
FACULTAD DE INGENIERÍA CIVIL, DE SISTEMAS Y DE ARQUITECTURA
ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS

TESIS

SISTEMA TUTOR INTELIGENTE PARA APOVAR EL APRENDIZAJE EN MATEMÁTICAS EN EL GRADO TERCERO DE SECUNDARIA EN LA INSTITUCIÓN EDUCATIVA PERUANO CANADIENSE.

PARA OPTAR EL TÍTULO DE
INGENIERO DE SISTEMAS

AUTORES
BACH. MALDONADO VASQUEZ ALEXEI BILLY
BACH. VASQUEZ VASQUEZ BRENDA WILMER

ASESOR
MA. ING. ARTEAGA LORA ROBERTO CARLOS

LAMBAYEQUE – PERÚ
2025

Derechos de autor 2025 Turnitin. Todos los derechos reservados.

MA. ING. ROBERTO CARLOS ARTEAGA LORA
DNI 16755764

RESUMEN

El propósito de este estudio fue implementar un Sistema Tutor Inteligente (STI) para fortalecer el aprendizaje de matemáticas en estudiantes de tercer grado de secundaria, mediante una herramienta tecnológica capaz de personalizar la enseñanza y focalizar contenidos según las áreas con mayor dificultad. Para ello, se aplicó un diseño cuasi-experimental con enfoque cuantitativo, trabajando con una muestra de 82 estudiantes que tenían el área de matemáticas desaprobada, seleccionados mediante muestreo no probabilístico por conveniencia. La implementación técnica del STI utilizó Python 3.10, bibliotecas especializadas como Pandas, NumPy y Scikit-learn, así como el algoritmo Random Forest Regressor para predecir la efectividad de los ejercicios según el historial de desempeño de cada estudiante, generando retroalimentación inmediata, ejercicios adaptativos y un ranking temático personalizado. Los resultados indicaron que el porcentaje de estudiantes en nivel bajo disminuyó de 63.41% a 17.07%, mientras que los estudiantes en nivel alto aumentaron de 4.88% a 35.37%, y los errores por tema se redujeron considerablemente en Exponentes (19,5%), Matrices (25,3%) y Factorización (35,7%). La evaluación evidenció que la integración de los algoritmos adaptativo y de análisis de desempeño permitió reforzar eficazmente las áreas con mayor dificultad y favorecer la progresión académica de los estudiantes. En conclusión, el STI demostró ser una solución eficaz, escalable y adaptable para el aprendizaje de matemáticas en la Institución Educativa Peruano Canadiense, mejorando la personalización de la enseñanza, la progresión académica y la adquisición de competencias algebraicas de manera progresiva y adaptativa.

Palabras clave: Aprendizaje, matemáticas, algoritmo, evaluación, desempeño

ABSTRACT

The purpose of this study was to implement an Intelligent Tutoring System (ITS) to strengthen mathematics learning for third-grade secondary students, using a technological tool capable of personalizing instruction and focusing content on areas of greatest difficulty. A quasi-experimental design with a quantitative approach was applied, working with a sample of 82 students who had failed mathematics, selected through non-probabilistic convenience sampling. The technical implementation of the ITS utilized Python 3.10, specialized libraries such as Pandas, NumPy, and Scikit-learn, as well as the Random Forest Regressor algorithm to predict the effectiveness of exercises based on each student's performance history, generating immediate feedback, adaptive exercises, and a personalized thematic ranking. The results indicated that the percentage of students at a low level decreased from 63.41% to 17.07%, while high-level students increased from 4.88% to 35.37%, and errors by topic were considerably reduced in Exponents (19.5%), Matrices (25.3%), and Factorization (35.7%). The evaluation showed that the integration of the adaptive and performance analysis algorithms effectively reinforced the most challenging areas and promoted students' academic progression. In conclusion, the ITS proved to be an effective, scalable, and adaptable solution for mathematics learning at the Peruano Canadiense Educational Institution, improving personalized instruction, academic progression, and the acquisition of algebraic competencies progressively and adaptively.

***Keywords**:* Learning, mathematics, algorithm, assessment, performance

ÍNDICE

RESUMEN 7

ABSTRACT 8

ÍNDICE 9

ÍNDICE DE FIGURAS 12

ÍNDICE DE TABLAS 14

I. INTRODUCCIÓN 15

II. MARCO TEÓRICO 18

2.1. Antecedentes 18

2.2. Bases Teóricas 21

2.2.1. Sistema Tutor Inteligente 21

2.2.2. Modelos de Sistemas Tutores Inteligentes 22

2.2.3. Arquitectura del Sistema Tutor Inteligente 24

2.2.4. Inteligencia Artificial 25

2.2.5. Enfoques de Inteligencia Artificial 26

2.2.6. Aprendizaje matemático 26

2.2.7. Bloques curriculares del aprendizaje matemático 28

2.3. Bases Conceptuales 30

2.3.1. Operacionalización de variables 30

III. DISEÑO METODOLÓGICO 31

3.1. Población y muestra 32

3.2. Técnicas, instrumentos, equipos y materiales 33

IV. RESULTADOS 37

4.1. Diagnóstico Inicial 37

4.1.1. Nivel de conocimiento 37

4.1.2. Análisis de desempeño por tema 38

4.1.3. Análisis de diagnóstico general....43

4.2. Diseño del Sistema Tutor Inteligente....43

4.2.1. Arquitectura general del sistema....44

4.2.2. Interfaz web del estudiante....46

4.2.3. API para Gestión de Exámenes....46

4.2.4. Análisis de rendimiento y ML....47

4.2.5. Base de Datos Relacional....48

4.2.6. Preparación del Entorno de Trabajo....48

4.2.7. Comunicación entre Módulos....50

4.3. Diseño del algoritmo adaptativo....50

4.3.1. Requerimientos del algoritmo adaptativo....50

4.3.2. Desarrollo del algoritmo....50

4.4. Desarrollo del algoritmo de análisis de rendimiento....56

4.4.1. Análisis del algoritmo de rendimiento....56

4.4.2. Análisis del algoritmo de ranking y generación de examen....59

4.4.3. Análisis del algoritmo de la retroalimentación continua....60

4.5. Implementación de los algoritmos en el STI....61

4.5.1. Configuración del STI....61

4.5.2. Estructura y Diseño modular....61

4.5.3. Endpoints principales....62

4.5.4. Modelos de Datos y Lógica....68

4.5.5. Desarrollo del Backend....72

4.5.6. Interfaces del Sistema Experto....74

4.6. Evaluación de la efectividad del STI....81

4.6.1. Nivel de conocimiento....81

4.6.2. Análisis de desempeño por tema....82

4.6.3. Análisis de diagnóstico final....85

V. DISCUSIÓN DE RESULTADOS.....86

VI. CONCLUSIONES 89

VII. RECOMENDACIONES 90

VIII. REFERENCIAS 91

IX. ANEXOS.....95

ÍNDICE DE FIGURAS

Figura 1 Interacción de los módulos del Sistema Tutor Inteligente 22
Figura 2 Agentes distribuidos en módulos 23
Figura 3 Clasificación de reglas 23
Figura 4 Arquitectura del STI 24
Figura 5 Fases del Aprendizaje Matemático 27
Figura 6 Bloques Matemáticos 29
Figura 7 Distribución porcentual de estudiantes por nivel de conocimiento 37
Figura 8 Comparación de respuestas correctas e incorrectas – Tema 1 (Exponentes)...38
Figura 9 Comparación de respuestas correctas e incorrectas – Tema 2 (Matrices) 39
Figura 10 Comparación de respuestas correctas e incorrectas – Tema 3 (Polinomios) .40
Figura 11 Comparación de respuestas correctas e incorrectas – Tema 4 (Productos Notables).....41
Figura 12 Comparación de respuestas correctas e incorrectas – Tema 5 (Factorización) 42
Figura 13 Módulo Fronted Angular (web_tesis_lt).....44
Figura 14 Módulo Fronted Angular (ws_wa_tesis).....45
Figura 15 Módulo Fronted Angular (ws_wa_tesis).....45
Figura 16 Diagrama de flujo del diagnóstico de desempeño del sistema tutor inteligente 51
Figura 17 Diagrama de flujo del algoritmo de análisis de rendimiento del sistema tutor inteligente 59
Figura 18 Diagrama de flujo del algoritmo de ranking y generación de examen del SIT 59
Figura 19 Diagrama de flujo del algoritmo de retroalimentación continua del SIT 60
Figura 20 Ejemplo de recuperación de ejercicios de un examen 63
Figura 21 Entrenamiento de modelo diagnóstico con datos históricos 64
Figura 22 Generador de modelos genérico.....65
Figura 23 Generación de examen personalizado.....66
Figura 24 Pantalla de Login del Estudiante.....75
Figura 25 Dashboard inicial del Alumno 76
Figura 26 Generación de Exámenes Personalizados 76
Figura 27 Panel de Examen Personalizado con Estadísticas.....77

Figura 28 Retroalimentación de Desempeño Académico 78
Figura 29 Historial de Intentos por Tema 79
Figura 30 Ejecución de un Examen Personalizado 80
Figura 31 Resultados de examen personalizado.....80
Figura 32 Registro de Intentos de Exámenes 81

ÍNDICE DE TABLAS

Tabla 1 Corriente del Aprendizaje....28
Tabla 2 Tabla de operacionalización de variable....30
Tabla 3 Registro de estudiantes por sección....32
Tabla 4 Registro de estudiantes desaprobados por sección....33
Tabla 5 Distribución del Nivel de Conocimiento según Calificaciones Obtenidas....37
Tabla 6 Desempeño del Tema 1: Exponentes I, II y III....38
Tabla 7 Desempeño del Tema 2: Matrices....39
Tabla 8 Desempeño del Tema 3: Polinomios....40
Tabla 9 Desempeño del Tema 4: Productos Notables....41
Tabla 10 Desempeño del Tema 5: Factorización....42
Tabla 11 Endpoint de Recuperación de Ejercicios....62
Tabla 12 Endpoint de Diagnóstico....63
Tabla 13 Endpoint de Modelos Genéricos....64
Tabla 14 Endpoint de Generación de Examen Personalizado....65
Tabla 15 Endpoint de Generación de Examen Diagnóstico....66
Tabla 16 Endpoint de Envío y Calificación....67
Tabla 17 Endpoints de Ranking Dinámico....67
Tabla 18 Endpoints de Reclasificación....68
Tabla 19 Comparación entre DTOs y Entidades....68
Tabla 20 Repositorios y método....69
Tabla 21 Estructura de carpetas....73
Tabla 22 Distribución del Nivel de Conocimiento según Calificaciones Obtenidas....82
Tabla 23 Desempeño del Tema 1: Exponentes I, II y III....82
Tabla 24 Desempeño del Tema 2: Matrices....83
Tabla 25 Desempeño del Tema 3: Polinomios....83
Tabla 26 Desempeño del Tema 4: Productos Notables....84
Tabla 27 Desempeño del Tema 5: Factorización....84

I. INTRODUCCIÓN

Las matemáticas han sido una de las asignaturas más desafiantes en el ámbito educativo, generando dificultades en la comprensión y aplicación de sus conceptos, estos factores como la ansiedad matemática, la falta de motivación y la percepción de complejidad han llevado a que un gran número de estudiantes experimente un bajo rendimiento en esta área (Villamizar et al., (2020). Además, la enseñanza tradicional basada en clases magistrales limita la participación activa del estudiante, dificultando el desarrollo del pensamiento lógico y la resolución de problemas (Chacón et al., (2020). Estos problemas no solo afectan el desempeño académico, sino que también pueden influir en la elección de carreras y en las oportunidades laborales futuras.

A nivel internacional, los problemas en el aprendizaje de matemáticas siguen siendo evidentes, según la prueba PISA de la OCDE, muchos estudiantes no logran el nivel básico en esta materia, lo que puede afectar su futuro académico y laboral en un mundo cada vez más digital (OCDE, 2019). Mientras tanto, países como Japón y Corea del Sur han mejorado sus resultados aplicando nuevas metodologías y usando tecnología en el aula (Espinosa, (2024). En este sentido, los Sistemas de Tutoría Inteligente (ITS) con inteligencia artificial se han convertido en una opción efectiva para hacer que el aprendizaje sea más personalizado y dinámico (Guaman et al., (2025). Estos sistemas ajustan el contenido según las necesidades del estudiante y brindan retroalimentación inmediata. Además, Rodríguez (2021) menciona que los ITS sirven como apoyo para los docentes, ofreciendo explicaciones y ejercicios adaptados tanto en clases presenciales como virtuales. Sin embargo, su aplicación todavía enfrenta desafíos, como la formación de los profesores y la falta de infraestructura adecuada en algunas instituciones.

A nivel nacional, Ocaña et al. (2019) destacan la necesidad de que la educación universitaria se adapte a los avances tecnológicos, incorporando inteligencia artificial para personalizar el aprendizaje y fortalecer las competencias digitales. Por otro lado, Paz (2021) identifica dificultades en la enseñanza de la geometría en una institución de Tumán, proponiendo un Sistema Tutor Inteligente (STI) como solución para mejorar el rendimiento académico. Asimismo, Cespedes (2021) evidencia deficiencias en el aprendizaje del inglés en un colegio de Ferreñafe, sugiriendo la implementación de un STI interactivo para optimizar la enseñanza del idioma, así reflejando la urgencia de integrar herramientas tecnológicas en el sistema educativo nacional para mejorar la calidad del aprendizaje.

A nivel local, en la I.E. Peruano Canadiense, los estudiantes de tercer grado de secundaria presentan dificultades en el curso de álgebra, lo que ha llevado a que 82 alumnos de todas las secciones se encuentren desaprobados. Esto puede deberse a diversos factores, como el nivel de dificultad del curso, la falta de bases sólidas en matemáticas y la ausencia de herramientas complementarias que refuercen su aprendizaje. Por esta razón, se propone el desarrollo de un sistema tutor inteligente que brinde apoyo personalizado a los estudiantes, permitiéndoles mejorar su rendimiento académico y fortalecer sus conocimientos en álgebra.

A partir de lo planteado, se formuló la siguiente pregunta de investigación: ¿De qué manera puede apoyarse el aprendizaje de matemáticas en los estudiantes de tercero de secundaria de la Institución Educativa Peruano Canadiense? Para responder esta interrogante, se propuso como objetivo general implementar un sistema tutor inteligente que contribuya al fortalecimiento del aprendizaje matemático en los estudiantes de dicho nivel educativo.

Con el propósito de alcanzar este objetivo, se definieron varias metas específicas.

En primer lugar, se planteó el diseño de un algoritmo adaptativo capaz de ajustar el nivel de enseñanza en función del rendimiento individual de cada estudiante. Asimismo, se consideró el desarrollo de un algoritmo de análisis de desempeño que permita generar un ranking temático personalizado, facilitando así la selección automatizada de contenidos destinados al refuerzo académico. Posteriormente, ambos algoritmos serían integrados dentro del entorno funcional del tutor inteligente, posibilitando la generación dinámica de ejercicios y contenidos de aprendizaje. Finalmente, se propuso evaluar la efectividad del sistema mediante una prueba posterior que permita medir su impacto en el rendimiento académico de los estudiantes.

II. MARCO TEÓRICO

2.1. Antecedentes

A nivel internacional, Romero (2024) evaluó el impacto de un ChatBot basado en inteligencia artificial en la enseñanza de Física para estudiantes de tercero de bachillerato en la Unidad Educativa González Suárez durante el refuerzo académico. Se aplicó un diseño cuasiexperimental con enfoque cuantitativo, utilizando pruebas y cuestionarios para medir su eficacia y usabilidad. Con una muestra de 68 estudiantes divididos en grupo de control y experimental, los resultados reflejaron mejoras notorias en el aprendizaje, reduciendo de 26 a solo 3 los alumnos en la categoría de "No alcanza el aprendizaje". La evaluación con el Sistema de Usabilidad del Software (SUS) arrojó una puntuación de 81.4, superando el umbral de aceptabilidad. Se concluye que el ChatBot es una herramienta efectiva para mejorar la comprensión de conceptos complejos, recomendando su adaptación a distintos entornos educativos para maximizar su impacto.

Mora (2022) tuvo como finalidad diseñar un prototipo de sistema tutor inteligente basado en web para el aprendizaje de Lenguajes de Programación, específicamente en lenguajes imperativos. Se empleó la metodología de prototipado rápido, desarrollando módulos específicos para el estudiante, dominio, tutor e interfaz. Como resultado, se logró un sistema adaptativo de evaluación del conocimiento que facilitó el aprendizaje autónomo y brindó a docentes y estudiantes herramientas para monitorear el progreso académico. Su implementación demostró que el uso de tecnologías inteligentes en la educación puede optimizar la enseñanza, favoreciendo un aprendizaje más eficiente y accesible.

Mamani (2020) tuvo como objetivo mejorar el aprendizaje de matemáticas en estudiantes de sexto de secundaria mediante un tutor web inteligente. El sistema identificó el estilo de aprendizaje de cada alumno (visual, auditivo o kinestésico) y adaptó la

enseñanza de forma personalizada, además de monitorear su progreso con evaluaciones automatizadas. La metodología empleada incluyó la construcción del tutor con la Metodología de Desarrollo de Tutor Inteligente y el desarrollo del software bajo RUP, utilizando herramientas como Java, PostgreSQL y GWT en NetBeans. Los resultados evidenciaron una mejora en el rendimiento académico y una aceptación del 87.5%, confirmando la efectividad del sistema como apoyo educativo sin sustituir al docente.

A nivel nacional, Benavides (2019) desarrolló un Sistema Tutor Inteligente basado en inteligencia artificial para mejorar el aprendizaje en estudiantes de secundaria, especialmente en matemáticas. El objetivo fue proporcionar una enseñanza personalizada mediante una plataforma web estructurada en tres módulos: estudiante, tutor y dominio. La metodología aplicada incluyó cinco fases: análisis de requisitos, diseño, desarrollo, pruebas y despliegue, implementadas en la Institución Educativa Flores de Villa con alumnos de tercero y cuarto grado. Como resultado, el sistema optimizó el aprendizaje, reduciendo la necesidad de clases particulares y fortaleciendo las habilidades cognitivas y técnicas de los estudiantes. En conclusión, la integración de inteligencia artificial en la educación permitió un aprendizaje más eficiente, autónomo y accesible, contribuyendo al desarrollo académico y personal de los alumnos.

Rosas y Sagastegui (2022) tuvieron como objetivo mejorar el aprendizaje musical en el Conservatorio Carlos Valderrama de Trujillo mediante un tutor inteligente en 2021. Se usó un enfoque aplicado con diseño pre-experimental, validando una encuesta con los coeficientes V de Aiken y Alfa de Cronbach. El tutor se desarrolló con la metodología Mobile-D, siguiendo fases de Exploración, Inicialización, Producción, Estabilización y Pruebas. Tras su implementación, el conocimiento de técnicas de afinación aumentó un 55%, el de tonalidades un 5.6%, la lectura de partituras un 22.2% y la simbología musical un 33.3%. Se aplicaron pruebas estadísticas como Shapiro-Wilk, Wilcoxon y T-Student

para el análisis. En conclusión, el tutor inteligente mejoró significativamente el aprendizaje musical en los estudiantes del programa preparatorio FOBAS-I.

Montenegro y Rios (2020) buscó mejorar la comprensión lectora de los estudiantes de cuarto grado de la I.E. El Divino Maestro en Cartavio en 2020 mediante un tutor inteligente con realidad aumentada. Se aplicó un diseño pre-experimental con una muestra de 26 alumnos, utilizando cuestionarios y fichas de registro validados con expertos y analizados en SPSS. El tutor se desarrolló con la metodología Mobile-D y se programó en Kotlin, integrando Google Firebase, DialogFlow y ARcore. Tras su implementación, el interés por la lectura aumentó un 50%, la identificación de elementos en tareas de comprensión lectora un 46.2% y la realización correcta de estas tareas un 69.3%, resultados validados con pruebas estadísticas. Se concluye que el tutor inteligente con realidad aumentada mejoró notorio la comprensión lectora de los estudiantes, fortaleciendo su interés y desempeño en la lectura.

A nivel local, Saavedra (2024) tuvo como objetivo mejorar la comprensión lectora en estudiantes de segundo grado de primaria de la I.E. 11050 Rómulo Cabrera Chilcón mediante un Sistema Tutor Inteligente (STI). Su desarrollo se basó en SCRUM y la metodología CRISP-DM, permitiendo clasificar a los alumnos en distintos niveles de aprendizaje (inicio, proceso y satisfactorio). Para ello, se aplicaron los modelos Alicia y ECE, los cuales identificaron áreas de mejora y brindaron estrategias de retroalimentación. Además, el sistema incluyó un chatbot con redes neuronales multicapa, que proporcionó definiciones y aclaraciones de palabras según la RAE. La validación del STI, mediante juicios de expertos, reflejó un 98% de satisfacción en adecuación funcional según la norma IEC 25010:2015. Se concluye que el uso de los modelos Alicia y ECE permitió una evaluación más precisa y personalizada, ayudando a los estudiantes a mejorar su comprensión lectora según su nivel (literal, inferencial y crítico).

Cespedes (2021) tuvo como propósito mejorar el aprendizaje del idioma inglés en estudiantes de tercer grado de secundaria en la I.E. “Santa Lucía” de Ferreñafe mediante un Sistema Tutor Inteligente (STI). Para su desarrollo, se empleó la lógica difusa de Mamdani, lo que permitió diagnosticar el nivel de conocimiento de los estudiantes y proporcionar recomendaciones personalizadas. Esta metodología se integró con la Metodología de Software Educativo (MEISE) para crear una aplicación interactiva. La validación del sistema se realizó mediante una escala Likert de 7 puntos, obteniendo una evaluación favorable por parte de expertos. Se concluye que la implementación del STI, basada en un modelo de lógica difusa, facilita la orientación y retroalimentación de los estudiantes, permitiendo conocer sus expectativas, así como la percepción de docentes y usuarios sobre su efectividad.

Paz (2021) tuvo como objetivo mejorar el aprendizaje de geometría en los estudiantes de la I.E.P. Internacional “Jeshua” de Tumán, enfocándose en el tema de triángulos. Se empleó un enfoque cuasiexperimental y se desarrolló un Sistema Tutor Inteligente (STI) siguiendo la metodología MeISE. Para estructurar las pruebas, se utilizó el modelo de aprendizaje de Van Hiele, y se implementó un algoritmo de lógica difusa para mejorar la precisión de las respuestas. Tras su aplicación, los resultados mostraron un aumento en los promedios de calificación: 31.59% en el grupo de control y 21.14% en el experimental. Se concluye que la organización de las preguntas con el modelo de Van Hiele y la aplicación del STI contribuyeron al progreso académico de los estudiantes en geometría.

2.2. Bases Teóricas

2.2.1. Sistema Tutor Inteligente

Son simulaciones de enseñanzas, aprendizaje y comunicación de un experto o especialista para incrementar el entendimiento del estudiante en un tema determinado. El

propósito principal es brindar un ambiente más agradable y flexible al alumno. Algunos mencionan que captan el conocimiento de los expertos en el área de pedagogía transformando el ambiente en una enseñanza dinámica, este sistema fomenta la toma de decisiones como lo haría un docente en una clase presencial (Rios y Portugal, 2019).

2.2.2. Modelos de Sistemas Tutores Inteligentes

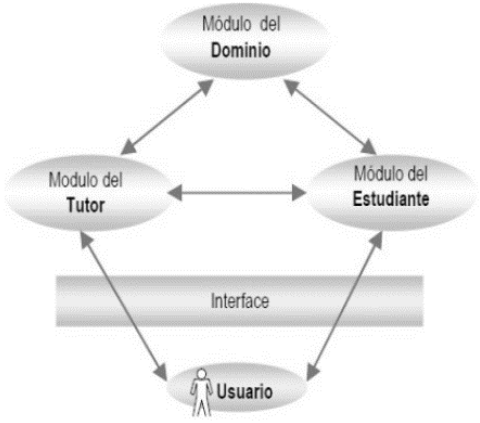
2.2.2.1. Modelo de Agente

Cada agente cuenta con características que lo definen estos mayormente son capacidad lógica, sensores, habilidades del servicio, efectores. Por tal motivo se clasifican en dos tipos, estos son (Rios y Portugal, 2019):

- **Agentes Humanos:** También conocidos como agentes externos, estos son el aprendiz o estudiante y el experto o el administrador.
- **Agentes software:** Se les conoce como los programas desarrollados con un fin último de enseñanza y aprendizaje, estos están configurados para cumplir con acciones tutor, diseño entendible, software colaborativo de manera pedagógica y evaluador.

Figura 1

Interacción de los módulos del Sistema Tutor Inteligente



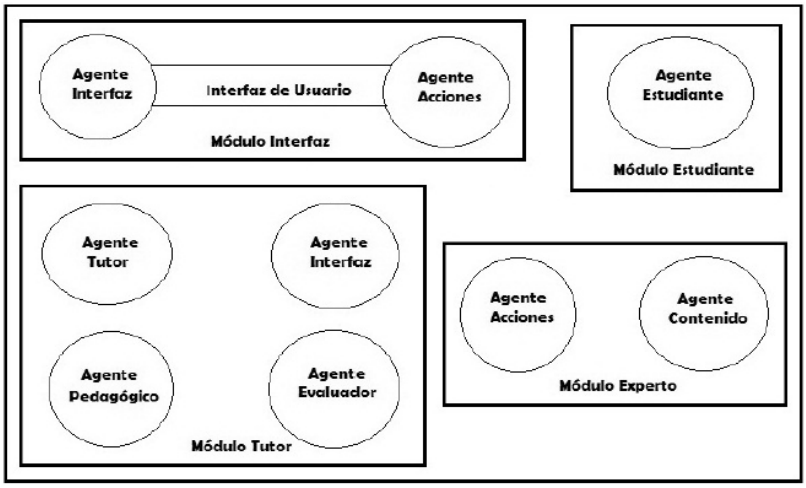
Nota. Tomado de (Rios & Portugal, 2019)

2.2.2.2. Modelo de la Experiencia

Son las habilidades del experto y su experiencia en el diseño del STI, relacionando la estructura del conocimiento con las diferentes tareas (Rios y Portugal, 2019).

Figura 2

Agentes distribuidos en módulos



Nota. Tomado de (Rios & Portugal, 2019)

2.2.2.3. Modelo del Tutor

Es el análisis de acciones que el estudiante puede llevar a cabo, así como los conocimientos requeridos al realizar diversas acciones de manera correcta (Sierra, García, Cataldi, y Hossian, 2015).

Figura 3

Clasificación de reglas

Regla	Descripción	Clasificación
R1	Inferir acerca de conocimiento sobre el mecanismo en base a una sola acción del educando	Conocimiento sobre el mecanismo
R2	Inferir acerca de conocimiento sobre el mecanismo en base a una serie de acciones del educando	Conocimiento sobre el mecanismo
R3	¿Está el estudiante utilizando una secuencia de dividir para conquistar en la resolución del problema?	Conocimiento sobre resolución de problemas
R4	¿Está el estudiante utilizando una secuencial en la resolución del problema?	Conocimiento sobre resolución de problemas
R5	¿Comprende el estudiante la operación de los componentes?	Conocimiento sobre componentes
R6	¿Está usando el estudiante la guía de resolución de problemas?	Conocimiento sobre resolución de problemas
R7	¿Está llevando a cabo el estudiante acciones repetitivas?	Conocimiento sobre el mecanismo / Conocimiento sobre resolución de problemas

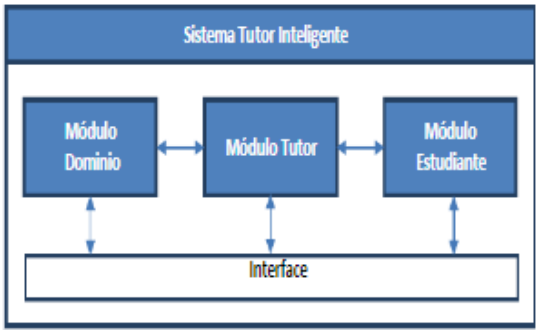
Nota. Tomado de (Sierra y otros, 2015)

2.2.3. Arquitectura del Sistema Tutor Inteligente

También llamados como procedimientos distribuidos con la finalidad del uso de una red de manera eficiente ya sea por medio del internet o algún otro. El Sistema tutor inteligente está conformado por cuatro módulos como se manifiesta en la Figura 4 (Molina et al., 2015).

Figura 4

Arquitectura del STI



Nota. Tomado de Molina et al. (2015).

2.2.3.1. Modulo Dominio

Es el encargado de almacenar todos los conocimientos del campo del sistema. Dentro de los submódulos se tienen los siguientes:

- Parámetros básicos del sistema: Almacenan su información en una base de datos.
- Conocimientos: Es la información que debe estar en el sistema para que el diagnóstico sea más exitoso, estos podrían ser ejercicios problemas y relaciones.
- Elementos didácticos: Son los adjuntos del sistema inteligente tales como sonidos, videos, imágenes u otros materiales multimedia facilitando el aprendizaje del conocimiento pedagógico.

2.2.3.2. Modulo Tutor

Es el módulo donde se define y a la vez aplica estrategias pedagógicas de enseñanza, mostrando el avance del alumno. Se exponen a la vez los siguientes submódulos:

- Protocolos pedagógicos: Acopiados en una base de datos, teniendo un gestor para estos mismos.
- Planificador de lección: Es el que organiza el mismo contenido.
- Analizador de perfil: Estudia las características del estudiante, escogiendo la estrategia pedagógica más provechoso.

2.2.3.3. Modulo Estudiante

Es donde se realiza el diagnóstico cognitivo del alumno. Sirve como orientación del alumno, así que cuando un alumno comete un error el sistema busca orientarlo. Dentro de los submódulos se tiene lo siguiente:

- Estilos de aprendizaje: Son datos almacenados con disponibles aprendizajes en el sistema, así como métodos de selección de diferentes estilos y peculiaridades en cada uno de ellos.
- Estado de conocimientos: Son los conocimientos adquiridos dentro del módulo del dominio y que está expuesto a modificaciones frente a diversos resultados obtenidos por el tutor.
- Interface: Es el que permite que los usuarios intervengan con el sistema, diseñado para su entendimiento, contando con flexibilidad y teniendo en cuenta diversas variables ante su creación para cumplir con las perspectivas del alumno, así como del docente.

2.2.4. Inteligencia Artificial

La Inteligencia Artificial se manifiesta de manera parecida al estado de funcionamiento del cerebro siendo parte de la Ciencia Computacional, relacionadas a las conductas humanas brindados por un conjunto de métodos, técnicas y herramientas para modelar y resolver problemas de manera más rápida y eficaz (Acaña, et al., 2019).

2.2.5. Enfoques de Inteligencia Artificial

Según lo que manifiesta Russell y Norving (2014), la inteligencia artificial se manifiesta en cuatro enfoques, estos son los siguientes:

- **Sistemas que piensan como humanos:** Se refiere a la automatización de las actividades que se asignan con los pensamientos humanos, así como actividades para la toma de decisiones, aprendizaje, etc.
- **Sistemas que piensan racionalmente:** Se piensa alcanzar no solo el comportamiento humano, sino también el razonamiento mediante el uso de modelos computacionales.
- **Sistemas que actúan como humanos:** Realizado para que las máquinas que puedan ejecutar acciones humanas por medio de la inteligencia. Pudiéndolos reemplazar perfectamente.
- **Sistemas que actúan racionalmente:** Son las acciones que las maquinas pueden hacer de manera racional.

2.2.6. Aprendizaje matemático

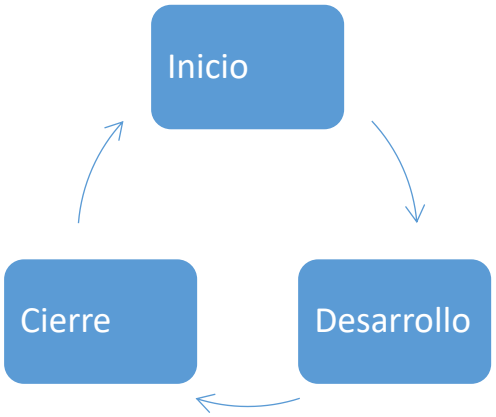
Es fundamental en la formación de los estudiantes, ya que el docente guía el aprendizaje mediante diversas estrategias didácticas, facilitando no solo la transmisión de conocimientos, sino también la construcción activa del saber, considerando factores sociales, culturales y pedagógicos que influyen en su desarrollo integral (Ampuero, 2022).

2.2.6.1. Fases del aprendizaje matemático

Normalmente el aprendizaje matemático pasa por un proceso de fases para que el aprendizaje sea dado de manera profunda, cumpliendo con cada uno de ellos a manera de ciclo, estos son los que se manifiestan en la Figura 5 (Devia & Pinilla, 2012).

Figura 5

Fases del Aprendizaje Matemático



Nota. Tomado de (Devia & Pinilla, 2012)

- Inicio: Influye mucho la comunicación para transmitir exitosamente la información, sin que este sea alterado, destacando la importancia, así como utilidad del mismo.
- Desarrollo: Se da a conocer los nuevos objetivos para alcanzar previamente usando los ya adquiridos. El experto pedagógico imparte estrategias creativas para llegar a una enseñanza más enriquecedora.
- Cierre: Muestra los resultados finales al aplicar la evaluación posterior al alumno midiendo su nivel de conocimientos adquiridos y el análisis de ellos, si es que manifiesta confusiones dentro de los estudiantes, se procede al método de aclaramientos de dudas, estos pueden ser dadas por medio de videos de reforzamiento, preguntas y resolución de dudas de los alumnos.

2.2.6.2. Corrientes del Aprendizaje

Para que el aprendizaje sea dado de manera exitosa existen dos corrientes que explican los procesos en el ámbito escolar, estos son la empírica o transmisiva y la constructivista. Estos son detallados en la siguiente tabla (Arce y otros, 2019).

Tabla 1

Corriente del Aprendizaje

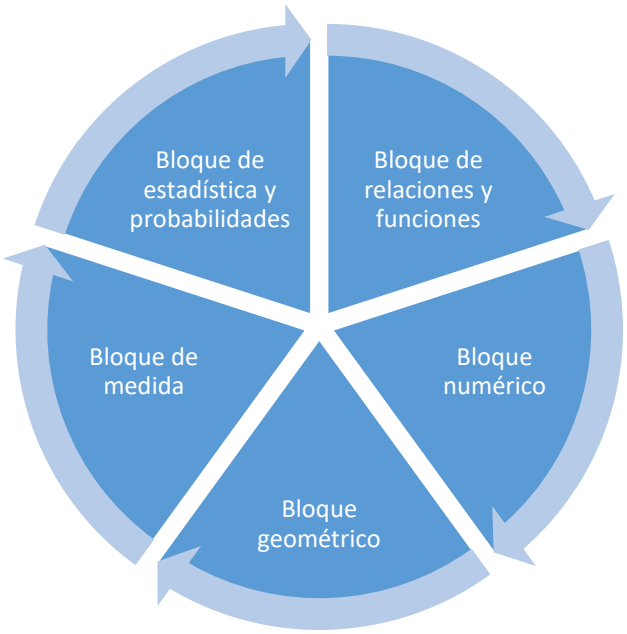
C. Empírica o transmisiva	C. Constructiva
El alumno capta lo que el docente explaya en la clase.	El estudiante aprende por sus acciones realizada dentro de la clase.
Los conocimientos son adquiridos por la transmisión del docente que lo van recolectando.	Los conocimientos son adquiridos por medio de conflictos presentados durante la clase y que el estudiante debe superar.
Rol principal de docente de impartir conocimientos.	Rol principal del estudiante como ente principal, ya sea por los aprendizajes de unos alumnos a otros.
Rol pasivo del estudiante como captador de conocimientos dados por el docente que después deben ser expresados.	Rol destacado del docente, ya que es el creador de situaciones del aprendizaje orientando la acción del alumno.

2.2.7. Bloques curriculares del aprendizaje matemático

Dentro de los bloques curriculares dentro del área matemáticas se encuentran los manifestados en la Figura 6 (Ministerio de educación Ecuador, 2018):

Figura 6

Bloques Matemáticos



Nota. Tomado de (Ministerio de educación Ecuador, 2018)

- Bloque de relaciones y funciones: Inicializa en los primeros años de construcciones patrones, objetos y figuras. Incrementa la complejidad cada año que pasa en los estudiantes siendo fundamentos básicos para la solución de ecuaciones, funciones y sucesiones matemáticas.
- Bloque numérico: Se detalla la forma de presentación de los números y la relación geométrica, así como las especializaciones de localizaciones, así como espaciales apoyando el razonamiento fluido de diversas estimaciones.
- Bloque geométrico: Se realiza el análisis de figuras con dos o tres dimensiones, además del desarrollo de diversos argumentos numéricos logrando comprender las operaciones y problemas matemáticos.
- Bloque de medida: Es el que analiza los diversos atributos de manera medible tanto como longitud, capacidad y peso logrados aprender desde los primeros

años de básica, con el objetivo de resolver problemas manifestados en tiempo real en el entorno.

- Bloque de estadística y probabilidades: La finalidad es que el alumno pueda formular problemas para la recopilación y organización de diagramas y muestra de datos pertinentes.

2.3. Bases Conceptuales

2.3.1. Operacionalización de variables

Se establece la siguiente operacionalización de variables, enfocada en el sistema tutor inteligente y el aprendizaje de matemáticas, las cuales son fundamentales para el objetivo de la investigación, detalladas a continuación:

Tabla 2

Tabla de operacionalización de variable

Variables	Definición Conceptual	Dimensiones	Indicadores
Sistema tutor inteligente	Es una herramienta basada en inteligencia artificial que personaliza la enseñanza mediante la adaptación de contenidos y estrategias pedagógicas según el desempeño y las necesidades de cada estudiante.	Funcionalidad del sistema	Usabilidad
Aprendizaje de matemáticas	Es el proceso mediante el cual un estudiante adquiere, comprende y aplica conceptos, procedimientos y habilidades matemáticas para resolver problemas y desarrollar el pensamiento lógico.	Nivel de conocimiento	Registro de notas
		Evaluación del aprendizaje	Prueba de base

Nota. Elaboración propia.

III. DISEÑO METODOLÓGICO

La presente investigación se enmarca dentro del tipo aplicada, dado que su finalidad es brindar una solución concreta a un problema específico: mejorar el aprendizaje de matemáticas en estudiantes de tercero de secundaria mediante la implementación de un sistema tutor inteligente. Este tipo de estudio permite trasladar conocimientos teóricos al contexto práctico, generando un impacto directo en el entorno educativo de la Institución Educativa Peruano Canadiense.

El enfoque adoptado fue cuantitativo, ya que se trabajó con datos numéricos recolectados a partir de la interacción de los estudiantes con el sistema. Este enfoque permitió obtener resultados objetivos, medibles y replicables. Se evaluaron variables como el rendimiento académico antes y después del uso del tutor inteligente, así como la precisión del sistema en la adaptación del contenido y su efectividad en la resolución de dudas en función del nivel de desempeño de cada estudiante.

En cuanto al diseño metodológico, se optó por un diseño cuasi-experimental, debido a que los participantes no fueron asignados aleatoriamente a grupos de control y experimental. Se trabajó con grupos ya existentes dentro de las secciones del nivel educativo, realizando comparaciones del rendimiento académico previo y posterior a la intervención con el tutor inteligente. Este diseño permitió analizar los efectos del sistema sobre un entorno real sin alterar la estructura organizativa original de los grupos.

El alcance de la investigación fue explicativo, pues no solo se buscó describir o correlacionar variables, sino también explicar el efecto del sistema tutor inteligente sobre el rendimiento académico de los estudiantes. A través del análisis de los resultados obtenidos en pruebas aplicadas antes y después de la intervención, se examinó la

influencia del sistema en los logros de aprendizaje, permitiendo establecer relaciones causales entre su uso y los resultados académicos observados.

Respecto a las hipótesis formuladas en el estudio, se propuso como hipótesis nula que no existe una diferencia en la efectividad del sistema tutor inteligente respecto a los métodos tradicionales de enseñanza de matemáticas en estudiantes de tercero de secundaria de la Institución Educativa Peruano Canadiense. En contraposición, la hipótesis alternativa plantea que el sistema tutor inteligente resulta más efectivo que las metodologías convencionales para fortalecer el aprendizaje matemático en dicho grupo estudiantil.

3.1. Población y muestra

Población

El estudio se llevó a cabo en la Institución Educativa Peruano Canadiense, enfocándose en los estudiantes del tercer grado de secundaria en ambos turnos (mañana y tarde), con un total de 202 estudiantes distribuidos en seis secciones, como se mostró en la siguiente tabla:

Tabla 3

Registro de estudiantes por sección

Turno	**Mañana**			**Tarde**			**Total**
Sección	A	B	C	D	E	F	
Estudiantes	34	34	33	34	34	33	202

Nota. Elaboración propia.

Muestra

Se tomó como muestra a 82 estudiantes que tienen el área de matemáticas desaprobada, distribuidos en las seis secciones del tercer grado de secundaria en la Institución Educativa Peruano Canadiense. La selección se realizó mediante un muestreo

no probabilístico por conveniencia, ya que estos estudiantes requieren mayor refuerzo en la materia y representan el grupo ideal para evaluar el impacto del Sistema Tutor Inteligente.

Tabla 4

Registro de estudiantes desaprobados por sección

Turno	**Mañana**				**Tarde**		**Total**
Sección	A	B	C	D	E	F	
Estudiantes desaprobados	12	10	14	16	13	17	82

Nota. Elaboración propia.

3.2. Técnicas, instrumentos, equipos y materiales

Base de datos estructurada

Para la recolección de datos se aplicaron dos instrumentos principales, diseñados para medir las variables del estudio: Sistema Tutor Inteligente (V1) y Aprendizaje de Matemáticas (V2). Ambos fueron elaborados en función de la operacionalización de variables, garantizando que cada dimensión e indicador se tradujera en ítems medibles y alineados con los objetivos de la investigación.

El primer instrumento correspondió al Cuestionario de Usabilidad del Sistema Tutor Inteligente, orientado a evaluar la funcionalidad del sistema, su facilidad de uso, claridad de la información y utilidad percibida por los estudiantes. Este cuestionario estuvo conformado por 8 ítems, estructurados en una escala tipo Likert de 5 puntos (1 = Totalmente en desacuerdo a 5 = Totalmente de acuerdo), permitiendo medir la percepción de los usuarios respecto a la experiencia de interacción con el sistema.

El segundo instrumento fue el Cuestionario de Conocimiento del Curso de Álgebra, diseñado para evaluar el nivel de aprendizaje de los estudiantes antes y después

de la implementación del sistema tutor. Este cuestionario incluyó 20 ítems de opción múltiple, distribuidos en cinco temas fundamentales: Exponentes, Matrices, Polinomios, Productos Notables y Factorización. Cada pregunta tuvo una única respuesta correcta, codificada como 1 (correcta) y 0 (incorrecta), lo que permitió su análisis cuantitativo y la comparación de resultados en las pruebas diagnósticas y posteriores.

Ambos instrumentos fueron revisados por tres expertos en educación matemática y sistemas inteligentes, quienes validaron la pertinencia, claridad y coherencia de los ítems, garantizando su validez de contenido. A partir de sus observaciones, se realizaron ajustes en la redacción de algunos ítems para mejorar su precisión técnica y su correspondencia con las competencias del curso de álgebra.

En cuanto a la confiabilidad, se aplicó el coeficiente Alfa de Cronbach para determinar la consistencia interna de los instrumentos. El Cuestionario de Usabilidad del Sistema Tutor Inteligente obtuvo un valor de $\alpha = 0.876$, lo que indica una alta confiabilidad y coherencia entre los ítems relacionados con la percepción funcional y pedagógica del sistema. Por su parte, el Cuestionario de Conocimiento del Curso de Álgebra presentó un coeficiente de $\alpha = 0.892$, evidenciando una muy alta consistencia interna entre los ítems que evalúan las competencias matemáticas de los estudiantes.

Estos resultados confirman que ambos instrumentos son válidos, confiables y adecuados para medir las dimensiones planteadas en esta investigación: la funcionalidad del sistema tutor inteligente y la evaluación del aprendizaje matemático, sustentando así el rigor metodológico del estudio y la fiabilidad de los datos obtenidos

Base de datos estructurada

Para la implementación del sistema tutor inteligente, se utilizó una base de datos que almacenó la información relevante de los estudiantes, como los intentos realizados,

los resultados obtenidos y las categorías de dificultad de los ejercicios. Esta base de datos también almacenó el desempeño histórico, que fue esencial para alimentar tanto el modelo de diagnóstico como el proceso de generación de exámenes personalizados. Además, el sistema permitió el registro y actualización continua del desempeño académico, generando una retroalimentación que se incorporó en futuras iteraciones del modelo.

Herramientas de procesamiento de datos

Para el procesamiento de los datos estudiantiles, se utilizó el lenguaje de programación Python junto con bibliotecas especializadas. La biblioteca Pandas (import pandas as pd) fue empleada para la carga (pd.read_csv()), transformación (df.apply(), df.groupby()) y limpieza de los datos, facilitando la estructuración del historial académico de los estudiantes y la preparación de las variables predictoras. NumPy (import numpy as np) se utilizó para operaciones matemáticas y vectorización de cálculos necesarios en el análisis del desempeño. Además, se aplicaron técnicas como codificación categórica (pd.get_dummies()), normalización (StandardScaler() de Scikit-learn), y separación de conjuntos de datos (train_test_split() de sklearn.model_selection) para preparar los datos antes del entrenamiento del modelo de aprendizaje automático.

Algoritmos de Machine Learning

Se implementó el algoritmo Random Forest Regressor (from sklearn.ensemble import RandomForestRegressor) como modelo principal para predecir la efectividad de cada ejercicio en función del historial de desempeño del estudiante. Este modelo fue entrenado con variables como el número de intentos, el tiempo de resolución, la dificultad del ejercicio y la precisión previa. Su objetivo fue estimar la utilidad esperada de un ejercicio, definiéndola como su potencial para reforzar áreas débiles. El modelo fue

ajustado mediante validación cruzada (cross_val_score()), y los parámetros óptimos se determinaron con búsqueda en malla (GridSearchCV()).

Software de programación y análisis de datos

El desarrollo del sistema tutor inteligente se realizó en Python 3.10, utilizando la biblioteca Scikit-learn (sklearn) para la implementación y evaluación del modelo Random Forest. Se trabajó en el entorno Jupyter Notebook, que permitió una integración eficaz entre el código, las salidas gráficas (usando matplotlib.pyplot y seaborn) y la documentación del proceso. La visualización del desempeño del modelo se apoyó en gráficas de error (plt.plot()) y matrices de importancia de características (model.feature_importances_). La combinación de código interactivo, documentación embebida y visualización contribuyó a una validación transparente y reproducible del sistema.

Computadora y acceso a Internet

El desarrollo del sistema se realizó en una computadora con las especificaciones adecuadas para ejecutar los algoritmos de Machine Learning y procesar los grandes volúmenes de datos. En particular, se utilizó una computadora con procesador Core i7, 16 GB de RAM, y una conexión a Internet de 100 MB, lo que permitió una adecuada ejecución de los procesos de análisis, así como la descarga de bibliotecas, recursos y acceso a plataformas en línea para documentación técnica durante el desarrollo del proyecto.

IV. RESULTADOS

4.1. Diagnóstico Inicial

4.1.1. Nivel de conocimiento

Para establecer el nivel de conocimiento previo en álgebra de los estudiantes de tercer grado de secundaria, se aplicó una prueba diagnóstica inicial, cuyo instrumento se detalla en el Anexo 2. Esta evaluación constó de 20 preguntas distribuidas en cinco temas clave: exponentes, matrices, polinomios, productos notables y factorización. La prueba fue aplicada a un total de 82 estudiantes y permitió obtener un registro base del desempeño individual y colectivo, identificando tanto el nivel general de aprendizaje como las áreas con mayor dificultad, con el fin de orientar el diseño adaptativo del sistema tutor inteligente.

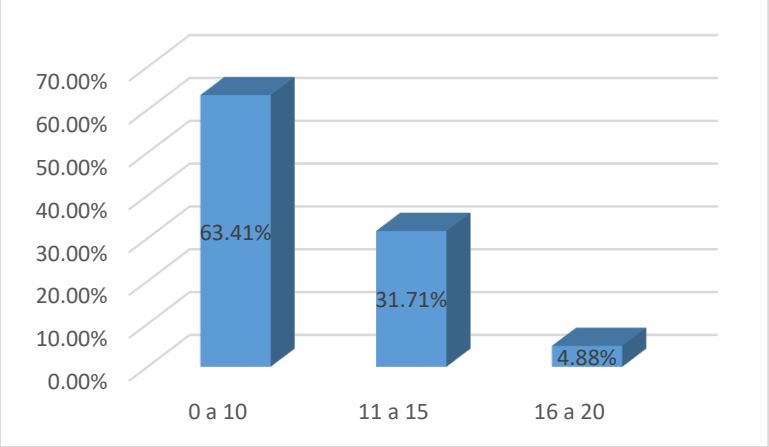
Tabla 5

Distribución del Nivel de Conocimiento según Calificaciones Obtenidas

Rango	Cantidad	%	Nivel
0 a 10	52	63.41%	Bajo
11 a 15	26	31.71%	Medio
16 a 20	4	4.88%	Alto
Total	82	100%	

Figura 7

Distribución porcentual de estudiantes por nivel de conocimiento



El 63.41% de los estudiantes se encuentra en un nivel bajo (calificaciones entre 0 y 10), lo que evidencia una comprensión deficiente de los contenidos matemáticos evaluados. Solo el 4.88% alcanzó un nivel alto (16 a 20), mientras que un 31.71% mostró desempeño medio. Esta distribución refleja una clara necesidad de intervención pedagógica, especialmente orientada a fortalecer las competencias básicas, ya que más del 95% no alcanza el nivel óptimo de rendimiento.

4.1.2. Análisis de desempeño por tema

A continuación, se presenta el análisis detallado del número de respuestas correctas e incorrectas por tema, con el propósito de identificar las áreas con mayores dificultades.

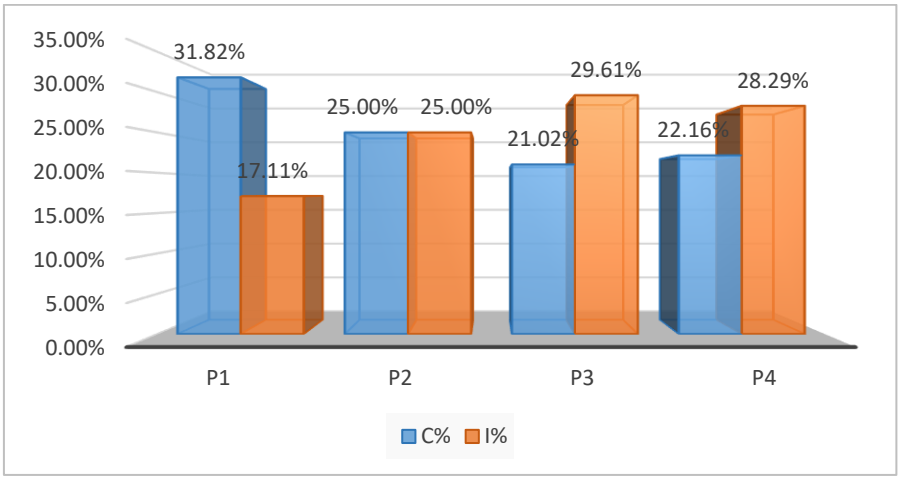
Tabla 6

Desempeño del Tema 1: Exponentes I, II y III

Tema I	Correctas	C%	Incorrectas	I%
P1	56	31.82%	26	17.11%
P2	44	25.00%	38	25.00%
P3	37	21.02%	45	29.61%
P4	39	22.16%	43	28.29%
Total	176	100%	152	100%

Figura 8

Comparación de respuestas correctas e incorrectas – Tema 1 (Exponentes)



En el tema de exponentes, aunque se registraron más respuestas correctas (176) que incorrectas (152), el 48.90% de errores revela una comprensión inestable del tema. Las preguntas P3 y P4 concentraron los mayores porcentajes de fallos (29.61% y 28.29%, respectivamente), lo que sugiere dificultades específicas en la aplicación de leyes combinadas y el manejo de exponentes negativos o fraccionarios. Esta situación refleja una apropiación superficial de los conceptos, que limita el desarrollo de competencias algebraicas más complejas.

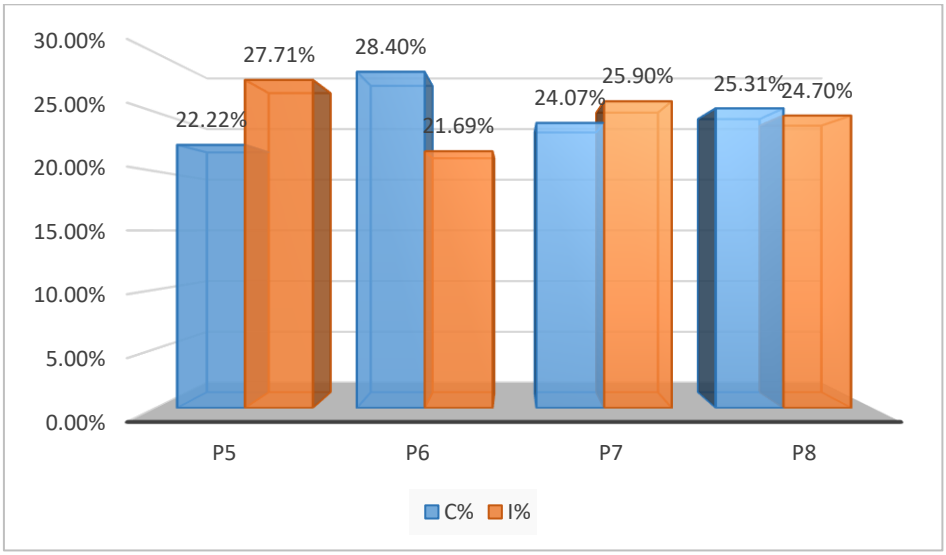
Tabla 7

Desempeño del Tema 2: Matrices

Tema II	Correctas	C%	Incorrectas	I%
P5	36	22.22%	46	27.71%
P6	46	28.40%	36	21.69%
P7	39	24.07%	43	25.90%
P8	41	25.31%	41	24.70%
Total	162	100%	166	100%

Figura 9

Comparación de respuestas correctas e incorrectas – Tema 2 (Matrices)



En el tema de matrices, la proximidad entre respuestas correctas (162) e incorrectas (166), con un 51.39% de error, evidencia un aprendizaje superficial. Las

mayores dificultades se concentraron en las preguntas P5 (27.71%) y P7 (25.90%), lo que indica problemas en la ejecución de operaciones fundamentales como suma, resta y multiplicación de matrices. Esta distribución sugiere que los estudiantes aplican procedimientos de forma mecánica, pero enfrentan limitaciones al integrar conceptos o resolver situaciones que exigen razonamiento estructurado.

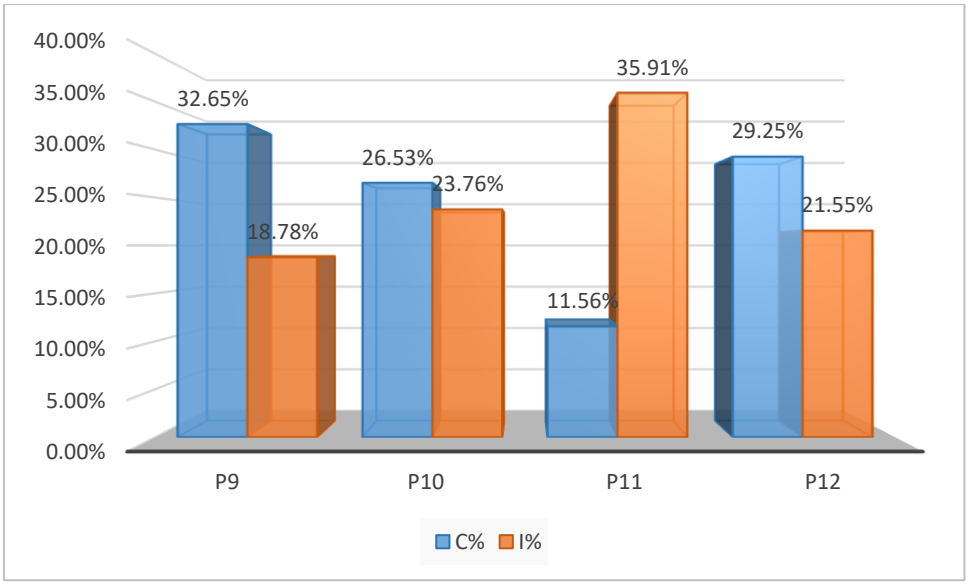
Tabla 8

Desempeño del Tema 3: Polinomios

Tema III	Correctas	C%	Incorrectas	I%
P9	48	32.65%	34	18.78%
P10	39	26.53%	43	23.76%
P11	17	11.56%	65	35.91%
P12	43	29.25%	39	21.55%
Total	147	100%	181	100%

Figura 10

Comparación de respuestas correctas e incorrectas – Tema 3 (Polinomios)



En el tema de polinomios se observa un descenso notable en el rendimiento, con 147 respuestas correctas frente a 181 incorrectas, lo que representa un 55.17% de error. La pregunta P11 destaca como la de mayor dificultad, con un 35.91% de errores y apenas un 11.56% de aciertos, reflejando problemas severos en la identificación de términos

semejantes, el desarrollo y la factorización de expresiones. Este bajo desempeño evidencia que los estudiantes no dominan las estructuras algebraicas básicas, lo que compromete su capacidad para abordar problemas de mayor complejidad en este tema.

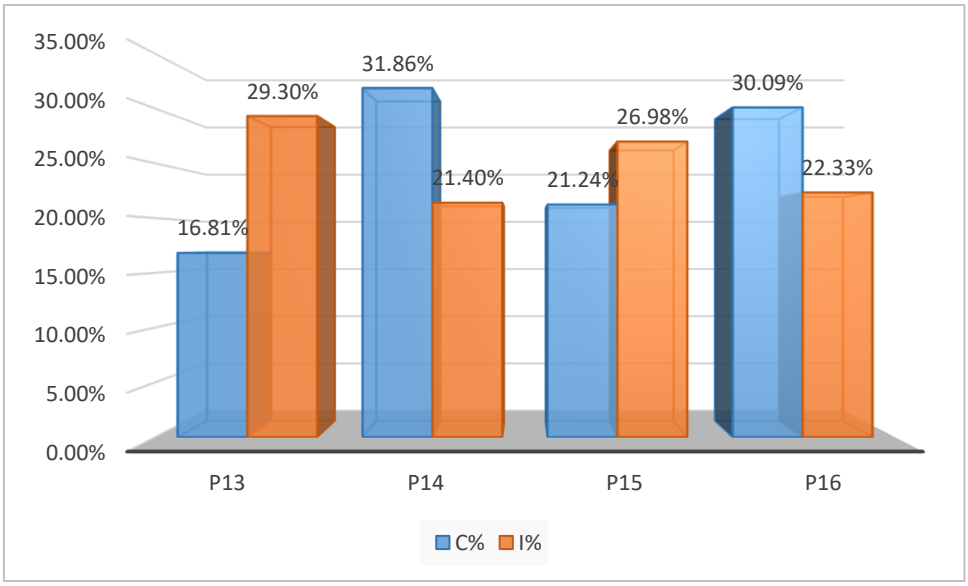
Tabla 9

Desempeño del Tema 4: Productos Notables

Tema IV	Correctas	C%	Incorrectas	I%
P13	19	16.81%	63	29.30%
P14	36	31.86%	46	21.40%
P15	24	21.24%	58	26.98%
P16	34	30.09%	48	22.33%
Total	113	100%	215	100%

Figura 11

Comparación de respuestas correctas e incorrectas – Tema 4 (Productos Notables)



En el tema de productos notables se identificó una de las mayores tasas de error del diagnóstico, con 215 respuestas incorrectas frente a 113 correctas, reflejando un 65.54% de equivocaciones. Las preguntas P13 y P15 concentraron los mayores errores, con 29.30% y 26.98% respectivamente, evidenciando que los estudiantes presentan serias dificultades en el reconocimiento y aplicación de fórmulas algebraicas como el cuadrado de un binomio o la diferencia de cuadrados. Este bajo desempeño indica una comprensión

mecánica y superficial del contenido, sin asimilar los principios que sustentan las identidades algebraicas, lo que limita su capacidad para resolver ejercicios más complejos que dependen de estas nociones.

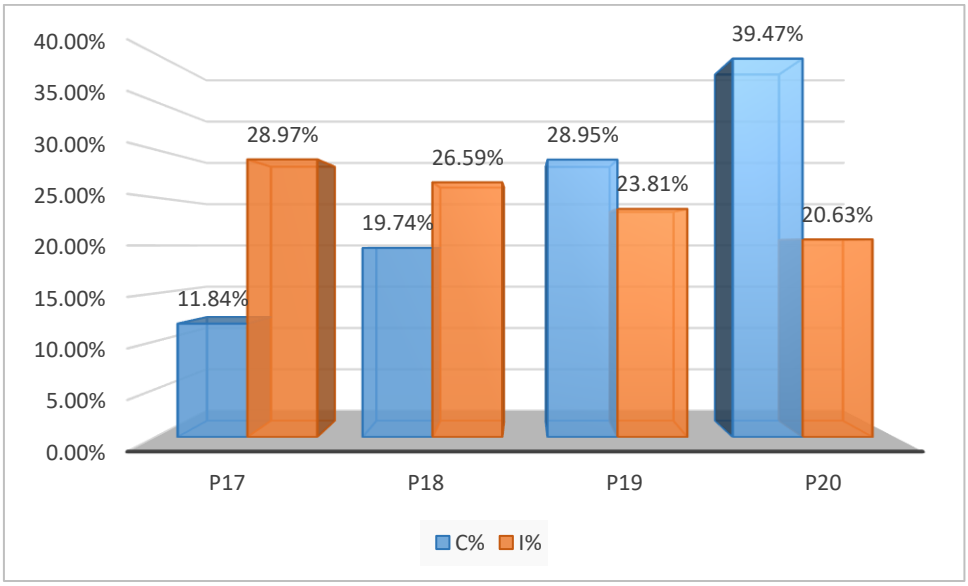
Tabla 10

Desempeño del Tema 5: Factorización

Tema V	Correctas	C%	Incorrectas	I%
P17	9	11.84%	73	28.97%
P18	15	19.74%	67	26.59%
P19	22	28.95%	60	23.81%
P20	30	39.47%	52	20.63%
Total	76	100%	252	100%

Figura 12

Comparación de respuestas correctas e incorrectas – Tema 5 (Factorización)



El tema de factorización representa el punto más crítico del diagnóstico, con solo 76 respuestas correctas (23.17%) frente a 252 incorrectas (76.83%). Las preguntas P17 y P18 agrupan los errores más altos, con tasas del 28.97% y 26.59%, respectivamente, lo que revela que los estudiantes no dominan técnicas elementales como el uso del factor común, trinomios cuadráticos o diferencia de cuadrados. Este patrón de bajo rendimiento sugiere una deficiencia estructural en la formación algebraica, ya que la factorización es

una competencia clave y transversal en las matemáticas. La elevada tasa de error evidencia una barrera cognitiva que requiere atención urgente mediante estrategias adaptativas y focalizadas que permitan superar estas brechas conceptuales.

4.1.3. Análisis de diagnóstico general

El diagnóstico inicial evidenció que el 63.41% de los estudiantes se ubicó en un nivel bajo de aprendizaje, mientras solo el 4.88% alcanzó un nivel alto. Al analizar el rendimiento por temas, se observó un desempeño relativamente equilibrado en *Exponentes* (48.90% de errores) y *Matrices* (51.39% de errores), lo que sugiere una comprensión parcial. Sin embargo, en *Polinomios* (55.17% de errores), *Productos Notables* (65.54%) y especialmente *Factorización* (76.83%), se detectaron fallas severas. Las preguntas con mayores errores se concentraron en los ítems P11 (35.91%), P13 (29.30%), P15 (26.98%), P17 (28.97%) y P18 (26.59%), lo que indica dificultades persistentes en la identificación de estructuras algebraicas y en la aplicación de procedimientos como factorización o desarrollo de productos notables. Estos resultados reflejan una base conceptual frágil que afecta la progresión académica, por lo que se diseñó un Sistema Tutor Inteligente adaptativo, centrado en reforzar los contenidos con mayor índice de error, brindar retroalimentación inmediata, y personalizar el aprendizaje según el nivel de desempeño individual.

4.2. Diseño del Sistema Tutor Inteligente

El diseño del Sistema Tutor Inteligente (STI) se estructuró en módulos funcionales con arquitecturas específicas para el frontend, backend Java, backend Python y la base de datos relacional. Cada componente fue diseñado para garantizar la escalabilidad, modularidad y mantenibilidad del sistema, con separación clara de responsabilidades, integración eficiente entre servicios y soporte para la adaptabilidad del contenido educativo.

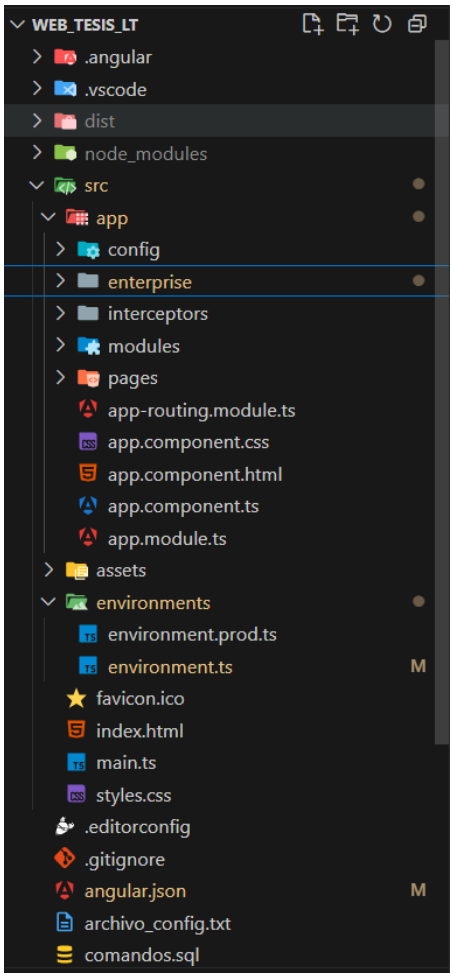
4.2.1. Arquitectura general del sistema

Se diseñó una arquitectura multicapa con separación de responsabilidades. La comunicación entre módulos se estableció mediante servicios RESTful. La base de datos central (MySQL) sirvió como punto de convergencia para el almacenamiento persistente. Se optó por contenedores Docker para garantizar la portabilidad, el aislamiento y la fácil implementación de cada servicio.

- **Frontend Angular (web_tesis_lt):** permitía la interacción directa con el estudiante o docente.

Figura 13

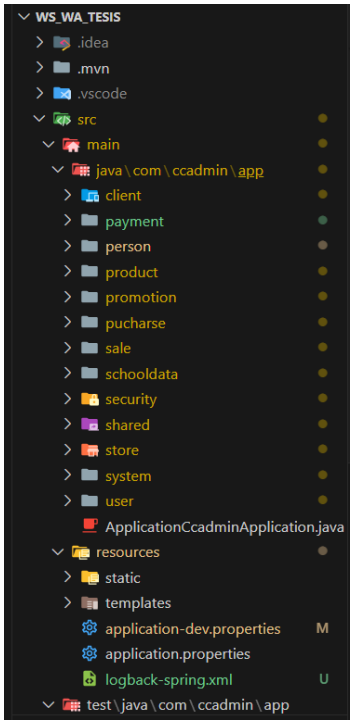
Módulo Fronted Angular (web_tesis_lt)



- **Backend Java (ws_wa_tesis):** gestionaba reglas de negocio, autenticación y lógica del sistema.

Figura 14

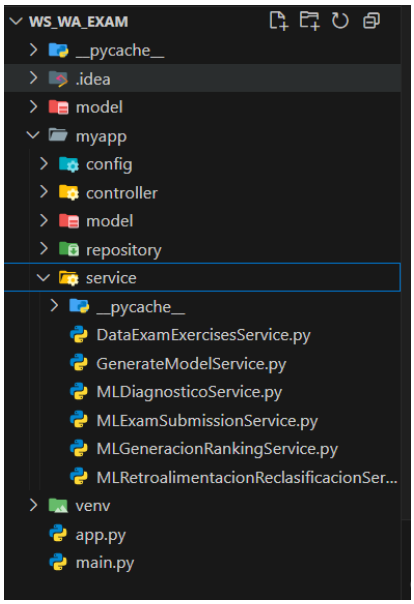
Módulo Fronted Angular (ws_wa_tesis)



- **Backend Python (ws_wa_exam):** analizaba el rendimiento y entrenaba modelos de predicción.

Figura 15

Módulo Fronted Angular (ws_wa_tesis)



- **Base de datos (MySQL):** centralizaba la persistencia de datos.

4.2.2. Interfaz web del estudiante

Se implementó una SPA (Single Page Application) en **Angular v15.2.10** para proporcionar una experiencia fluida e interactiva. Esta interfaz permitía a los usuarios crear, configurar, rendir exámenes y visualizar sus resultados.

Componentes clave:

- **student/views/** contenía vistas como "Lista de Estudiantes" o "Resultados".
- **student/components/** incluía elementos como formularios o tablas.
- **student/services/** gestionaba el consumo de la API backend.

A continuación, la integración de servicio:

```
// student/services/student.service.ts

getStudentResults(studentId: number): Observable<ExamResult[]> {

  return this.http.get<ExamResult[]>(`${this.apiUrl}/results/${studentId}`);

}
```

Se utilizaron bibliotecas como ngx-toastr para notificaciones amigables y sweetalert2 para confirmaciones interactivas. Se integraron también @ng-bootstrap para componentes visuales responsivos.

4.2.3. API para Gestión de Exámenes

Este módulo, desarrollado con **Spring Boot 3.2.0**, se encargó de la lógica central del sistema: autenticación, gestión de usuarios, exámenes, preguntas y registros históricos.

Tecnologías clave:

- **JPA/Hibernate:** persistencia de datos.

- **JWT:** autenticación basada en tokens.
- **Endpoints REST:** para la interacción con el frontend.

Se utilizó JPA para mapear entidades como Exam, Student y Question a sus respectivas tablas.

```
@GetMapping("/exams/{studentId}")
public ResponseEntity<List<ExamDTO>> getExamsByStudent(@PathVariable Long studentId) {
    return ResponseEntity.ok(examService.getExamsForStudent(studentId));
}
```

4.2.4. Análisis de rendimiento y ML

Este microservicio se desarrolló en **Flask** y actuaba como motor de inteligencia artificial. Se encargaba de procesar el rendimiento histórico, generar rankings y entrenar modelos para mejorar la personalización de los exámenes.

Servicios principales:

- MLDiagnosticoService.py: generaba modelos de diagnóstico del estudiante.
- GeneracionRankingController.py: producía exámenes adaptativos según rendimiento.

Servicio de predicción:

```
# service/MLDiagnosticoService.py
def predecir_rendimiento(student_data):
    modelo = joblib.load("modelo_random_forest.pkl")
    prediccion = modelo.predict([student_data])
```

return predicción

Frameworks utilizados:

- pandas, numpy: procesamiento de datos.
- scikit-learn: entrenamiento y validación de modelos.
- joblib: serialización de modelos.
- Flask-CORS: habilitación de peticiones desde el frontend.

4.2.5. Base de Datos Relacional

Se diseñó un esquema relacional en MySQL 8.0.32 que almacenó datos de usuarios, preguntas, exámenes, respuestas, puntuaciones y modelos. Esta base era compartida por ambos backends, asegurando integridad y sincronización:

- Frameworks utilizados:
- pandas, numpy: procesamiento de datos.
- scikit-learn: entrenamiento y validación de modelos.
- joblib: serialización de modelos.
- Flask-CORS: habilitación de peticiones desde el frontend.

4.2.6. Preparación del Entorno de Trabajo

Para facilitar la ejecución en distintos entornos, se creó un entorno Dockerizado con **Docker Compose**, lo que permitió levantar todos los servicios con un único comando.

Pasos seguidos:

1. Se creó un archivo .env con las variables de entorno.
2. Se definieron contenedores para:
    - Frontend (Nginx + Angular build)
    - Backend Java (Spring Boot)

- Backend Python (Flask)
- Base de datos (MySQL)

3. Se incluyó un script para cargar automáticamente el backup db_sys_expert_backup.sql.

Fragmento de Docker-compose.yml

```
services:
  frontend:
    build: ./web_tesis_lt
    ports:
      - "80:80"
  backend_java:
    build: ./ws_wa_tesis
    ports:
      - "8080:8080"


  backend_python:
    build: ./ws_wa_exam
    ports:
      - "4000:4000"
  mysql:
    image: mysql:8.0
    ports:
      - "3306:3306"
    environment:
      MYSQL_ROOT_PASSWORD: root
```

MYSQL_DATABASE: db_sys_expert

4.2.7. Comunicación entre Módulos

La comunicación entre módulos se realizó a través de peticiones HTTP usando APIs REST. La interfaz Angular consumió servicios expuestos por el backend Java, y este, a su vez, consumió endpoints del backend Python para realizar análisis más complejos, como, por ejemplo:

- El frontend solicitaba la generación de un nuevo examen al backend Java.
- Este consultaba preguntas en la base de datos, evaluaba el desempeño previo a través del servicio Python, y retornaba un conjunto personalizado de preguntas.

4.3. Diseño del algoritmo adaptativo

4.3.1. Requerimientos del algoritmo adaptativo

El objetivo principal fue diseñar un algoritmo adaptativo que ajustara el nivel de enseñanza según el desempeño del estudiante, utilizando datos históricos de exámenes. El sistema debía ser capaz de:

- Analizar el desempeño del estudiante en función de los ejercicios resueltos.
- Identificar fortalezas y debilidades en diferentes tópicos.
- Ajustar el contenido de enseñanza de acuerdo con los resultados de cada estudiante.

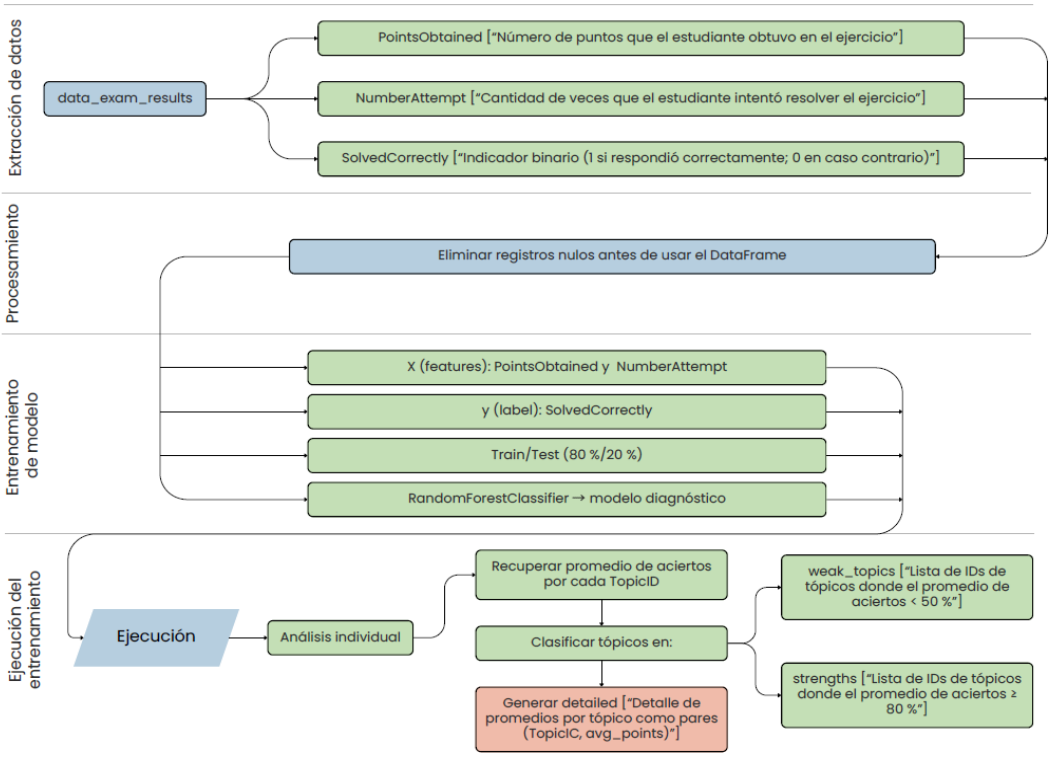
4.3.2. Desarrollo del algoritmo

El algoritmo fue desarrollado los siguientes pasos, que se detallan a continuación:

4.3.2.1.Análisis del algoritmo

Figura 16

Diagrama de flujo del diagnóstico de desempeño del sistema tutor inteligente



Se describe el proceso automatizado que permite al sistema tutor inteligente identificar las fortalezas y debilidades de cada estudiante, iniciando así con la extracción de datos de los resultados de los ejercicios, considerando puntos obtenidos, número de intentos y si la respuesta fue correcta; tras depurar la información, se entrena un modelo de clasificación (*Random Forest*) que analiza el desempeño y calcula el promedio de aciertos por tema. Con estos datos, los tópicos se clasifican en débiles (menos del 50 % de aciertos) y fuertes (80 % o más), generando un reporte detallado que servirá como base para personalizar los contenidos y reforzar las áreas de mayor dificultad.

4.3.2.2.Extracción y Preprocesamiento de Datos

Primero, se obtuvo la información sobre el desempeño de los estudiantes desde una base de datos llamada **"data_exam_results"**. Esta base guarda detalles sobre cómo les fue en cada ejercicio del examen, incluyendo:

- **PointsObtained:** los puntos que el estudiante obtuvo en cada ejercicio.
- **NumberAttempt:** cuántas veces intentó responder ese ejercicio.
- **SolvedCorrectly:** un valor que indica si respondió correctamente o no.

Por ejemplo, si respondió bien, el valor es 1, y si respondió mal, es 0.

Se seleccionaron únicamente los registros con datos completos, sin valores faltantes, para garantizar la calidad del análisis. Esto evito que el algoritmo genere resultados incorrectos debido a información incompleta. Por ejemplo, solo se consideraron los registros con el número de intentos registrado para asegurar la precisión del procesamiento. A continuación, se muestra el fragmento de código utilizado para esta selección:

```
query = """
   SELECT PointsObtained, NumberAttempt, SolvedCorrectly
   FROM data_exam_results
   WHERE NumberAttempt IS NOT NULL
"""
df = pd.read_sql(query, con=db.engine)
```

4.3.2.3.Entrenamiento del Modelo de Diagnostico

Después de preparar los datos, se entrenó un modelo usando el algoritmo “RandomForestClassifier” para predecir si un estudiante responderá correctamente un ejercicio. Este algoritmo crea múltiples árboles de decisión basados en ejemplos previos y combina sus resultados para generar una predicción precisa. A continuación, el procedimiento:

1) **Seleccionar las características y etiquetas**:

   Características (X): Son los datos que el modelo usará para aprender, en este caso:

   - Cuántos puntos obtuvo el estudiante en cada ejercicio (PointsObtained)
   - Cuántos intentos hizo para ese ejercicio (NumberAttempt)

   Etiqueta (y): Es la respuesta que queremos predecir, si el estudiante respondió correctamente o no (SolvedCorrectly).

2) **Separar los datos para entrenamiento y prueba**:

   Para verificar que el modelo funciona bien, los datos se dividieron en dos grupos:

   - Uno para que el modelo aprenda (entrenamiento)
   - Otro para probar si el modelo realmente predice bien (prueba)

   Se usó un 80% de los datos para entrenar y un 20% para probar.

3) **Entrenar el modelo**:

   El modelo se "entrena" para aprender la relación entre los puntos, intentos y si la respuesta fue correcta o no.

4) **Guardar el modelo entrenado**:

   Para usar el modelo después, se guardó en un archivo llamado diagnostic_model.pkl.

   A continuación, el código desarrollado:

   from sklearn.ensemble import RandomForestClassifier # Importa el algoritmo Random Forest

   import joblib # Librería para guardar y cargar modelos

```
# Selecciona las columnas que usaremos para predecir (características)

X = df[['PointsObtained', 'NumberAttempt']]  # Datos de puntos obtenidos y número de intentos

# Selecciona la columna que queremos predecir (etiqueta)

y = df['SolvedCorrectly']  # Si respondió correctamente (1) o no (0)

# Divide los datos en dos partes: entrenamiento y prueba

from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(

    X, y, test_size=0.2, random_state=42

)

# test_size=0.2 significa que el 20% de los datos se usa para prueba

# random_state=42 asegura que la división sea reproducible

# Crea el modelo Random Forest

model = RandomForestClassifier()

# Entrena el modelo con los datos de entrenamiento

model.fit(X_train, y_train)

# Guarda el modelo entrenado en un archivo para usarlo después

joblib.dump(model, 'diagnostic_model.pkl')
```

4.3.2.4.Identificación de Fortalezas y Debilidades

Una vez entrenado el modelo de diagnóstico, se utilizó para analizar el rendimiento de cada estudiante en los diferentes temas (o "tópicos") del curso. El objetivo fue identificar en qué temas el estudiante es fuerte y en cuáles necesita más apoyo. A continuación, la clasificación de los temas:

- **Débil**: Si el promedio de aciertos del estudiante en un tema fue menor al 50%.
- **Fuerte**: Si el promedio fue igual o superior al 80%.
- Los temas con puntajes entre 50% y 79% no se consideraron en el ajuste, ya que se consideran en un nivel medio.

Este análisis permitió que el sistema adaptara el contenido y las preguntas según las necesidades reales del estudiante. A continuación, lo indicado:

```
def analyze_student_performance(student_id):
    # Obtiene el promedio de aciertos del estudiante por cada tópico (tema)
    topic_rows = PerformanceRepository.get_topic_avg_points(student_id)
        weak_topics = []      # Lista para guardar los temas donde el estudiante es débil
    strengths = []      # Lista para guardar los temas donde el estudiante es fuerte
    # Recorre cada tema con su promedio de puntos obtenidos
    for row in topic_rows:
        avg_points = row['avg_points']  # Promedio de puntos en el tema actual
        # Si el promedio de aciertos es menor al 50%, se considera un tema débil
```

```
            if avg_points < 50:
                weak_topics.append(row['TopicID'])  # Se añade a la lista de temas débiles
            # Si el promedio de aciertos es mayor o igual a 80%, se considera un tema fuerte
            elif avg_points >= 80:
                strengths.append(row['TopicID'])  # Se añade a la lista de temas fuertes
            # Si está entre 50% y 79%, no se clasifica como fuerte ni débil
        # Devuelve dos listas: una con los temas débiles y otra con los fuertes
        return weak_topics, strengths
```

4.4. Desarrollo del algoritmo de análisis de rendimiento

4.4.1. Análisis del algoritmo de rendimiento

El diagrama de flujo detalla el proceso para construir un modelo predictivo basado en datos históricos que evalúa la utilidad de cada ejercicio y optimice la selección de contenidos para los estudiantes.

1. **Definir alcance y parámetros**: Se establecen los objetivos del análisis y los criterios que guiarán el modelado, como umbrales mínimos de datos o métricas de precisión esperadas.
2. **Recopilar datos históricos**: Se extrae información acumulada de intentos de ejercicios, respuestas correctas e incorrectas y niveles de dificultad.
3. **Comprobar volumen de datos por ejercicio**: Se verifica que cada ejercicio cuente con un número suficiente de registros para generar métricas confiables.

4. **Decisión – umbral mínimo**: Si un ejercicio no alcanza el mínimo de datos, se omite del análisis; si lo supera, los datos se limpian y homogenizan para evitar inconsistencias.

5. **Calcular métricas básicas por ejercicio** *(comentario: total de intentos, total de aciertos, tasa de acierto)*:

    - **Total de intentos**: suma de todos los intentos realizados.
    - **Total de aciertos**: conteo de intentos con respuesta correcta.
    - **Tasa de acierto**: (aciertos / intentos) × 100 %.

6. **Derivar métrica de refuerzo (utilidad histórica)** *(comentario: promedio de intentos + 1 % de error)*:

    - El **% de error** se calcula como 100 % – tasa de acierto.
    - La utilidad histórica combina la dificultad que tuvo el alumno con la frecuencia de intentos, valorando más los ejercicios donde hay margen de aprendizaje.

7. **Escalar utilidad a un rango estándar** *(comentario: valores entre 0 y 100)*: Se multiplica la métrica de refuerzo por 100 y se redondea.

8. **Preparar el conjunto de entrenamiento** *(comentario: características y opcionales)*:

    - Características: utilidad histórica escalada, nivel de dificultad, y opcionalmente categoría o tema.

9. **Verificar idoneidad del conjunto**: Se revisa que las variables sean relevantes y estén correctamente formateadas.

10. **Dividir datos para entrenamiento y validación** *(comentario: 80 % / 20 %)*.

11. **Configurar y entrenar el modelo de regresión** *(comentario: Random Forest con parámetros iniciales)*: Se define número de árboles, profundidad máxima, semilla aleatoria y se entrena el modelo.

12. **Evaluar rendimiento en el conjunto de validación**: Se calculan métricas como R^2 y RMSE.

13. **Decisión – criterios de precisión** *(ejemplo: $R^2 \geq 0.6$ y $RMSE \leq 10$)*:

    - Si no se cumplen, se ajustan parámetros y se repite la evaluación.

14. **Retrain final**: Se entrena nuevamente usando todos los datos disponibles.

15. **Persistir el modelo entrenado**: Guardar el modelo para su uso futuro.

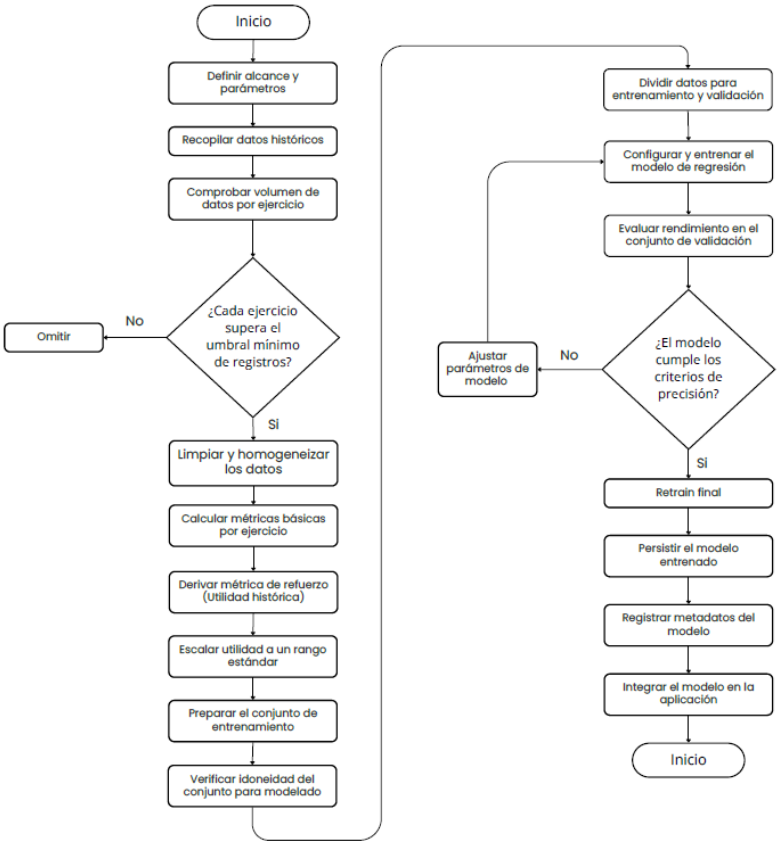
16. **Registrar metadatos del modelo**: Documentar fecha, parámetros y métricas obtenidas.

17. **Integrar el modelo en la aplicación**: Hacer que esté disponible para el tutor inteligente.

18. **Fin**.

Figura 17

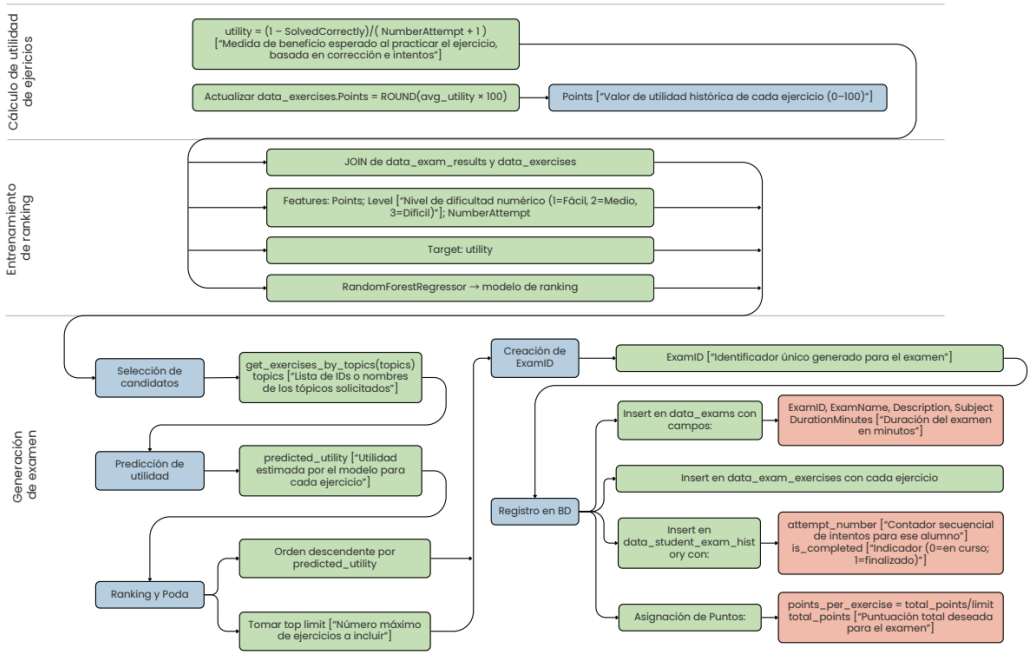
Diagrama de flujo del algoritmo de análisis de rendimiento del sistema tutor inteligente



4.4.2. Análisis del algoritmo de ranking y generación de examen

Figura 18

Diagrama de flujo del algoritmo de ranking y generación de examen del SIT

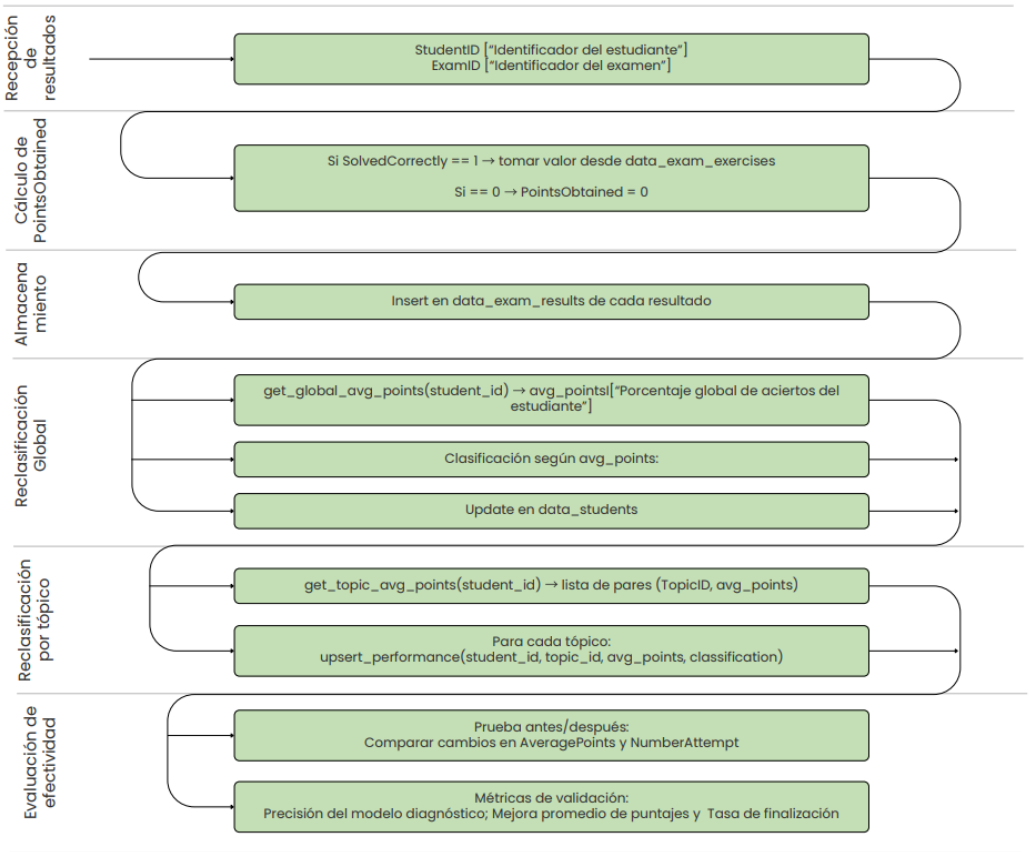


Se describe el proceso para seleccionar y organizar los ejercicios más útiles para cada estudiante, generando automáticamente un examen personalizado. El procedimiento comienza calculando la **utilidad histórica** de cada ejercicio, en función de si fue resuelto correctamente y el número de intentos. Luego, se entrena un modelo predictivo (*Random Forest Regressor*) que estima la utilidad futura de los ejercicios considerando su dificultad y el historial del estudiante. Con esta información, el sistema filtra los ejercicios por temas solicitados, predice su utilidad, los ordena de mayor a menor y selecciona los de mayor valor. Finalmente, se genera un identificador único para el examen, se registran sus detalles en la base de datos, se asocian los ejercicios seleccionados y se actualiza el historial del alumno, garantizando así una evaluación adaptada a sus necesidades y nivel de dominio.

4.4.3. Análisis del algoritmo de la retroalimentación continua

Figura 19

Diagrama de flujo del algoritmo de retroalimentación continua del SIT



Representa el proceso mediante el cual el sistema tutor inteligente actualiza y evalúa el progreso del estudiante de forma constante. Comienza con la recepción de los resultados de cada examen, identificando al estudiante y la prueba correspondiente. A partir de la corrección de las respuestas, se asignan los puntos obtenidos y se almacenan en la base de datos. Luego, el sistema realiza una reclasificación global, calculando el promedio general de aciertos del alumno, y una reclasificación por tópico, actualizando el rendimiento en cada área temática. Finalmente, se evalúa la efectividad del sistema comparando el desempeño antes y después de la intervención, midiendo indicadores como la precisión del modelo diagnóstico, la mejora promedio de puntajes y la tasa de finalización de exámenes. Este proceso asegura que el tutor inteligente no solo se adapte, sino que también verifique su impacto en el aprendizaje de forma objetiva y continua.

4.5. Implementación de los algoritmos en el STI

4.5.1. Configuración del STI

El servicio inicia cargando sus variables críticas desde el entorno (por ejemplo: DB_URI, FLASK_ENV, LOG_LEVEL), evitando que datos sensibles como las cadenas de conexión queden expuestos en el código fuente. La inicialización se gestiona en **app.py**, donde:

- Se crea la aplicación Flask y se aplica la configuración definida en **Config.py**.
- Se establece la conexión con la base de datos mediante **SQLAlchemy**, utilizando la variable de entorno DB_URI.
- Se registran los **Blueprints** correspondientes a cada controlador.

4.5.2. Estructura y Diseño modular

Para garantizar orden y escalabilidad, la carpeta **myapp/** se organiza de la siguiente manera:

- **controller/**: contiene los *Blueprints* de cada API (diagnóstico, ranking, envío de exámenes, entre otros).
- **service/**: implementa la lógica de negocio, coordinando repositorios y modelos de *Machine Learning*.
- **repository/**: agrupa las clases que gestionan el acceso a la base de datos, ya sea mediante **SQLAlchemy** o *queries* personalizadas.
- **model/dto/**: define los *Data Transfer Objects* (DTO) que estandarizan la comunicación en formato JSON.
- **model/entity/**: incluye las entidades mapeadas a tablas de la base de datos (por ejemplo: *data_exam_exercises*, *data_student_exam_history*).
- **config/**: almacena los archivos de configuración y definiciones de logs.

4.5.3. Endpoints principales

4.5.3.1. Recuperación de ejercicios

Permite obtener los ejercicios vinculados a un examen específico. Antes de devolver los datos, valida la existencia del examen y los permisos de acceso del usuario.

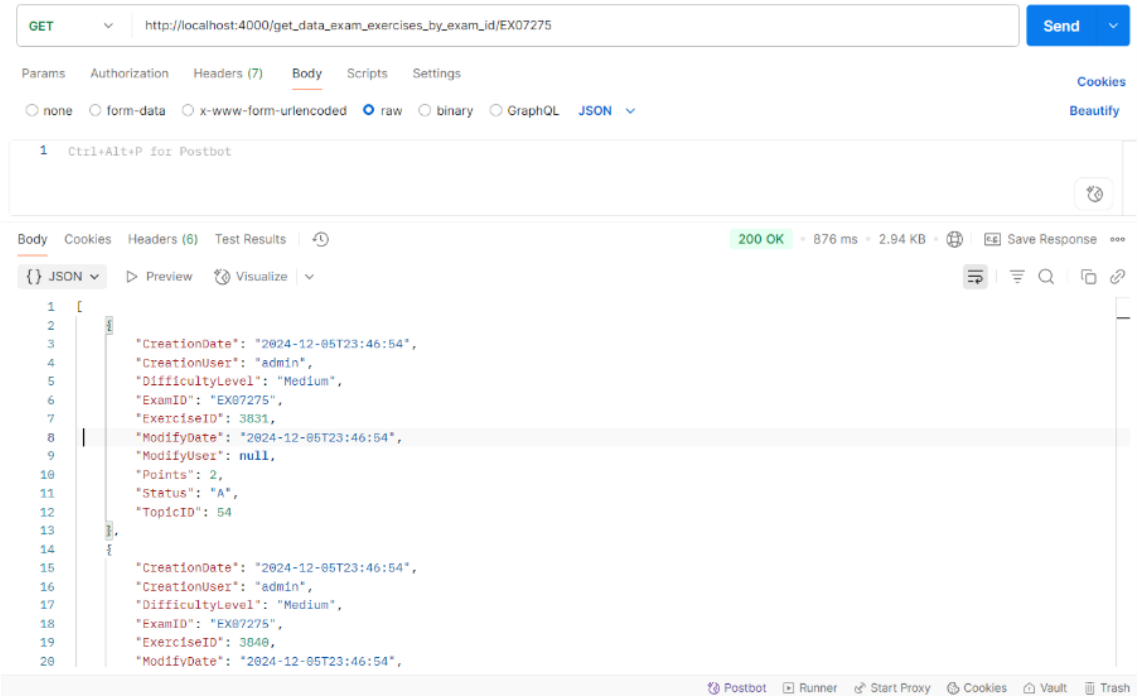
Tabla 11

Endpoint de Recuperación de Ejercicios

Propiedad	Detalle
Descripción	Recupera la lista de ejercicios asociados a un examen validando permisos
URL	GET /exam/get_data_exam_exercises_by_exam_id/<exam_id>
Controlador	DataExamExercisesController
Función	get_data_exam_exercises_by_exam_id(exam_id)

Figura 20

Ejemplo de recuperación de ejercicios de un examen



4.5.3.2.Diagnóstico del estudiante

Este endpoint entrena o actualiza el modelo de diagnóstico utilizando registros históricos del estudiante (respuestas correctas/incorrectas, tiempo de resolución). Devuelve métricas y la versión actualizada del modelo.

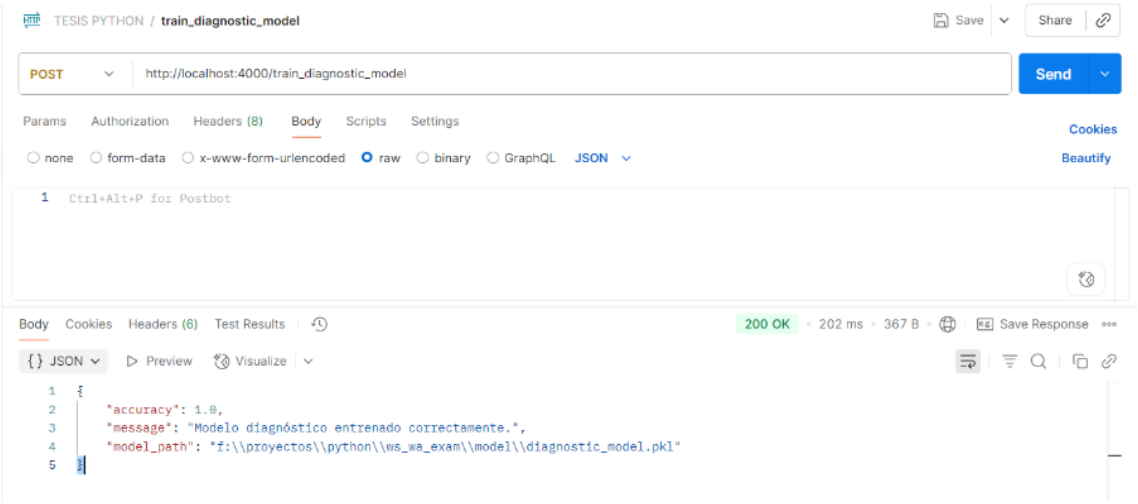
Tabla 12

Endpoint de Diagnóstico

Propiedad	Detalle
Descripción	Entrena o actualiza el modelo de diagnóstico con datos históricos
URL	POST /train_diagnostic_model
Controlador	DiagnosticModelController
Función	train_diagnostic_model()

Figura 21

Entrenamiento de modelo diagnóstico con datos históricos



4.5.3.3.Generación de modelos genéricos

Permite la creación o actualización de modelos genéricos de predicción, identificados por un nombre único.

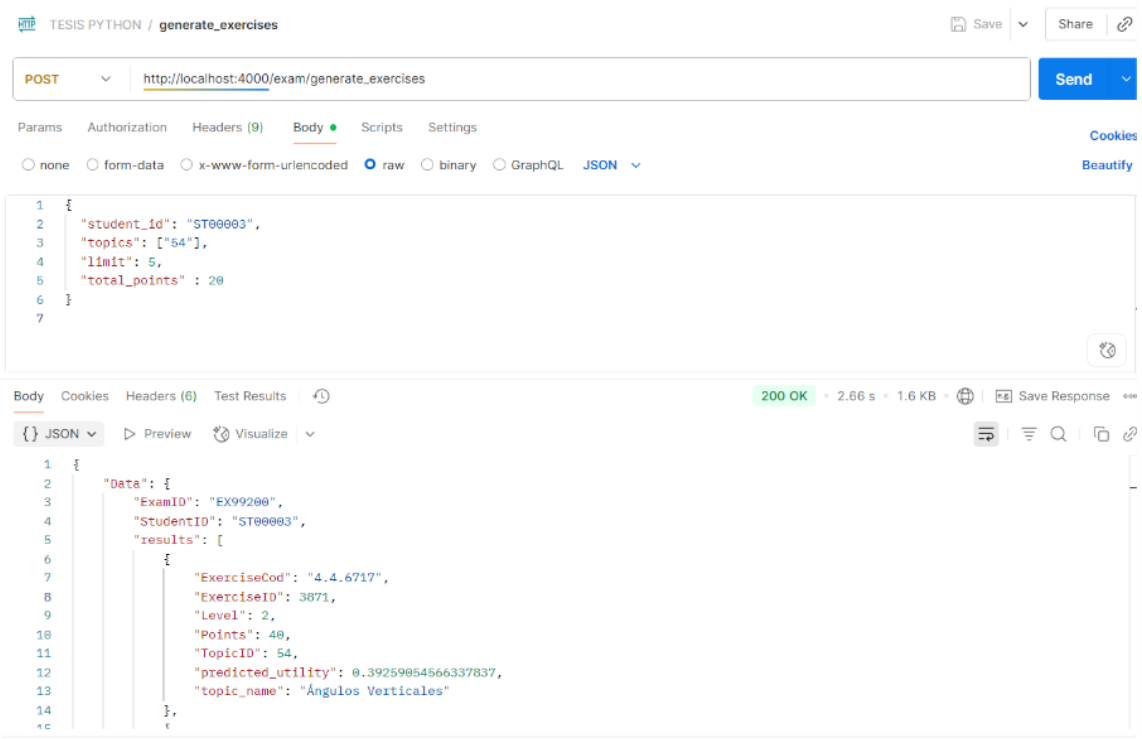
Tabla 13

Endpoint de Modelos Genéricos

Propiedad	Detalle
Descripción	Crea o actualiza un modelo genérico identificado por nombre
URL	GET /generatemodel/accept/<ModelName>
Controlador	GenerateModelController
Función	accept(ModelName)

Figura 22

Generador de modelos genérico



4.5.3.4.Generación de exámenes

Los exámenes pueden ser personalizados según tópicos y dificultad, o bien diagnósticos de entrada para nuevos estudiantes.

a) Examen Personalizado

Genera un examen con preguntas adaptadas al estudiante y lo registra con estado `IN_PROGRESS`.

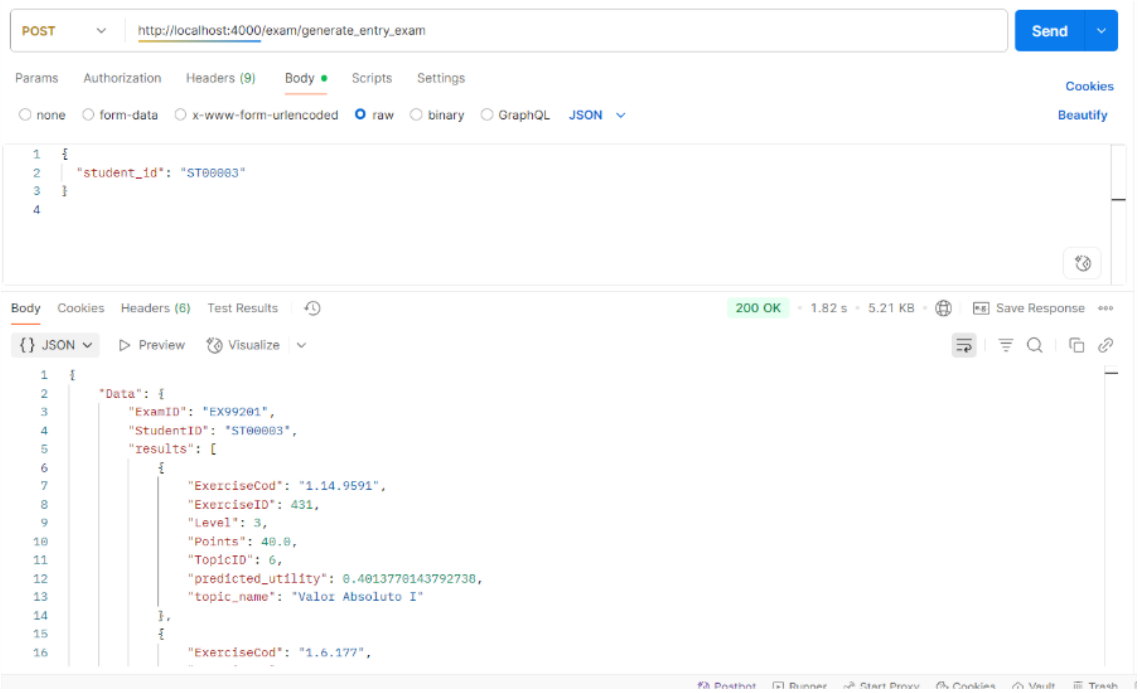
Tabla 14

Endpoint de Generación de Examen Personalizado

Propiedad	Detalle
Descripción	Genera un examen adaptado según tópicos, dificultad y cantidad de ítems
URL	POST /exam/generate_exercises
Controlador	ExamSubmissionController
Función	generate_exercises(request_dto)

Figura 23

Generación de examen personalizado



b) Examen Diagnóstico Inicial

Asigna automáticamente tópicos y dificultad para un alumno nuevo.

Tabla 15

Endpoint de Generación de Examen Diagnóstico

Propiedad	Detalle
Descripción	Genera examen diagnóstico inicial para estudiantes
URL	POST /exam/generate_entry_exam
Controlador	ExamSubmissionController
Función	generate_entry_exam(request_dto)

4.5.3.5.Envió y calificación

Recibe las respuestas del alumno, valida el estado del examen, calcula puntajes y actualiza el historial. También dispara el proceso de reclasificación.

Tabla 16

Endpoint de Envío y Calificación

Propiedad	Detalle
Descripción	Envía respuestas, calcula puntaje total y actualiza historial
URL	POST /exam/submit
Controlador	ExamSubmissionController
Función	submit_exam(submission_dto)

4.5.3.6.Ranking dinámico

Este módulo organiza los ejercicios en función de su utilidad predicha y permite reentrenar el modelo de ranking.

- **Generación de examen rankeado**: GET /generate_rank_exam/generate_exam/<student_id>
- **Entrenamiento de modelo de ranking**: POST /generate_rank_exam/train

Tabla 17

Endpoints de Ranking Dinámico

Propiedad	Detalle
Descripción	Genera examen ordenado por utilidad o reentrena el modelo
Controlador	GeneracionRankingController

4.5.3.7.Reclasificación y retroalimentación

Estos servicios recalculan el nivel de desempeño del estudiante, tanto a nivel global como por tópico.

- **Por tópicos**: POST /reclassify_student/topics/<student_id>
- **Global**: POST /reclassify_student/global/<student_id>

Tabla 18

Endpoints de Reclasificación

Propiedad	Detalle
Descripción	Recalcula nivel global y por tópicos del estudiante
Controlador	RetroalimentacionReclasificacionController

4.5.4. Modelos de Datos y Lógica

4.5.4.1.DTOs y Entidades

Los modelos de datos son la base del intercambio de información. Se utilizan DTOs para el transporte de datos en formato JSON y Entities para la persistencia en base de datos.

Tabla 19

Comparación entre DTOs y Entidades

Elemento	Características	Aplicado
DTOs	Clases simples con atributos públicos, método to_dict(). Mantienen contrato JSON inmutable.	DataExamExercisesDTO que representa la entidad y expone solo campos relevantes.
Entities	Heredan de db.Model. Incluyen campos de auditoría (CreationUser, CreationDate, ModifyUser, ModifyDate, Status).	DataExamExercisesEntity, DataStudentExamHistoryEntity.

4.5.4.2.Repositorios de Acceso y persistencia

La carpeta repository/ concentra las operaciones contra la base de datos, aislando las consultas SQL de la lógica de negocio.

Tabla 20

Repositorios y método

Repositorio	Métodos	Funcionalidad
DataStudentExamHistoryRepository	insert_exam_attempt(), mark_attempt_completed()	Registra intentos y los marca como finalizados
DataExamExercisesRepository	find_by_exam_id()	Obtiene ejercicios por examen
ExamResultsRepository	insert_result()	Guarda resultados de cada pregunta
PerformanceRepository	get_topic_avg_points()	Calcula desempeño promedio por tópico

La transaccionalidad se asegura con db.engine.begin(), garantizando atomicidad en cada operación.

4.5.4.3.Servicios de negocio y flujo de ejecución

os servicios orquestan repositorios y modelos de ML para dar soporte al ciclo completo del examen.

4.5.4.3.1 Diagnóstico de Áreas Críticas

- **Entrada**: MLDiagnosticoService.train(historical_results)
- **Preprocesamiento**: imputación de valores faltantes y cálculo de métricas derivadas (accuracy, velocidad promedio).

- **Entrenamiento**: RandomForestClassifier con n_estimators=100, max_depth=10.
- **Persistencia**: modelo serializado con joblib.
- **Salida**: {model_version, metrics}.

4.5.4.3.2 Generación de Ranking de Ejercicios

- Entrada: MLGeneracionRankingService.generate(weak_topics, student_id).
- Predice utilidad (predicted_utility) por ejercicio.
- Aplica reglas de negocio: máximo 5 ejercicios por tópico, diversidad de dificultad.
- Devuelve lista priorizada de ejercicios.

4.5.4.3.3 Creación de Examen Personalizado

- Orquestación desde ExamSubmissionController.generate_exercises(request_dto).
- Consulta al ranking → repositorios → construcción de ExamEntity.
- Registro inicial en data_student_exam_history con estado IN_PROGRESS.
- Respuesta con exam_id y lista de preguntas.

4.5.4.3.4 Procesamiento de Envío y Calificación

- Validaciones: existencia del examen, estado IN_PROGRESS, autor válido.
- Cálculo de puntos y persistencia de resultados.
- Actualización de historial y disparo asincrónico de reclasificación.

- Respuesta con puntaje total y desglose.

4.5.4.3.5 Retroalimentación y Reclasificación

- **Global**: reasigna nivel (Beginner, Intermediate, Advanced) en data_students.
- **Por tópico**: recalcula desempeño en student_topic_performance.

4.5.4.4.Gestión de excepciones y Retries

Todas las operaciones críticas están envueltas en try/except para mantener la robustez.

- Rollback automático en errores de base de datos.
- Reintentos configurados con backoff exponencial (librería tenacity).
- Respuestas normalizadas:
    - 400 Bad Request para errores de validación.
    - 500 Internal Server Error para fallos internos.

4.5.4.5.Manejo de errores y formato de respuesta

Se centraliza la salida con la clase ResponseWsDto, que incluye:

- Status
- Message
- Data
- ErrorStatus
- ErrorID
- DataAdditional

4.5.4.6.Pruebas Unitarias e Integración Continua

Para garantizar calidad:

- **Unit tests** con pytest (mocks de repositorios y servicios ML).
- **Integration tests** con contenedor MySQL vía Docker Compose.
- **Pipeline CI** en GitHub Actions: instala dependencias, ejecuta linters, corre pruebas y genera reporte de cobertura.

4.5.4.7.Seguridad, Autenticación y Control de acceso

La seguridad se gestiona mediante middleware que valida tokens **JWT** en el Authorization Header.

- Control de acceso por roles: student, instructor, admin.
- Manejo de claves privadas en variables de entorno o servicios de Vault.
- Respuestas estandarizadas:
    - 401 Unauthorized cuando falta el token.
    - 403 Forbidden cuando el rol no tiene permisos.

4.5.5. Desarrollo del Backend

Este módulo implementa el núcleo de servicios del SIT, basado en Spring Boot y diseñado bajo criterios de separación de responsabilidades, mantenibilidad y escalabilidad.

4.5.5.1.Configuración y arranque

- **Framework:** Spring Boot.
- **Configuración sensible:** variables de entorno o application-{profile}.properties.

- **Escaneo automático:** paquetes com.ccadmin.app.schooldata para controladores, servicios, repositorios e interceptores.
- **Formato estándar de respuesta:** ResponseWsDto (Data, Status/Message, DataAdditional).

4.5.5.2.Diseño modular y estructura de carpetas

Tabla 21

Estructura de carpetas

Carpeta	Responsabilidad	Ejemplos
controller/	APIs REST	ExamController, StudentController, ImageController
service/	Lógica de negocio	ExamService, ExerciseService, StudentService
repository/	Acceso a datos con JPA/SQL/SP	ExamRepository, StudentRepository
model/dto/	Transferencia de datos (entrada/salida)	ExamRegisterDto, StudentRegisterWithUserDto
model/entity/	Mapeo a tablas (JPA)	data_exams, data_student_exam_history
model/idto/	Proyecciones de queries nativas	ICourseWeaknessDTO, IStudentExamAttemptInfoDto
model/entity/id/	Claves compuestas	ExamExerciseID, StudentTopicPerformanceId

4.5.5.3.Funcionalidades (APIs principales)

Exámenes

- Crear/actualizar examen (/api/v1/exam/save).
- Obtener examen por ID o búsqueda paginada.
- Consultar tópicos débiles del alumno (/api/v1/exam/getStudentWeakTopics).

Ejercicios

- Registro individual o masivo.
- Asociación de ejercicios a exámenes.
- Almacenamiento de imágenes en base64.

Estudiantes

- Registro y actualización de alumnos.
- Creación de usuarios con rol *student*.

Historial y desempeño

- Registro de intentos de examen.
- Seguimiento de desempeño por tópico (student-topic-performance).

Imágenes

- Subida (/api/image/upload) y recuperación (/api/image/view/{filename}).

4.5.6. Interfaces del Sistema Experto

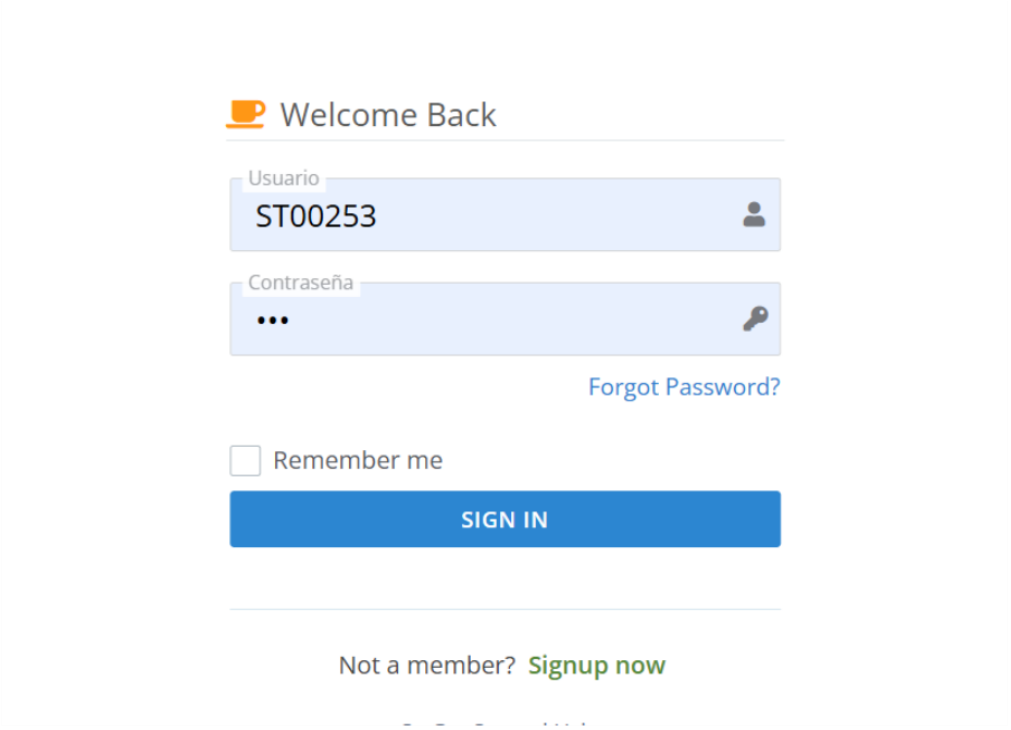
En esta sección se describen las principales pantallas del Sistema Experto de Evaluación y Diagnóstico Académico (SIT), implementadas como parte del frontend web para estudiantes. Cada vista corresponde a un flujo de interacción específico, integrando componentes visuales con las funcionalidades de backend descritas previamente.

4.5.6.1 Formulario de Inicio de Sesión

- Entrada de credenciales: código de usuario y contraseña.
- Función “Recordar sesión” para persistir la autenticación.
- Enlace de recuperación de contraseña.
- Botón **SIGN IN** que valida contra el servicio AuthController.

Figura 24

Pantalla de Login del Estudiante

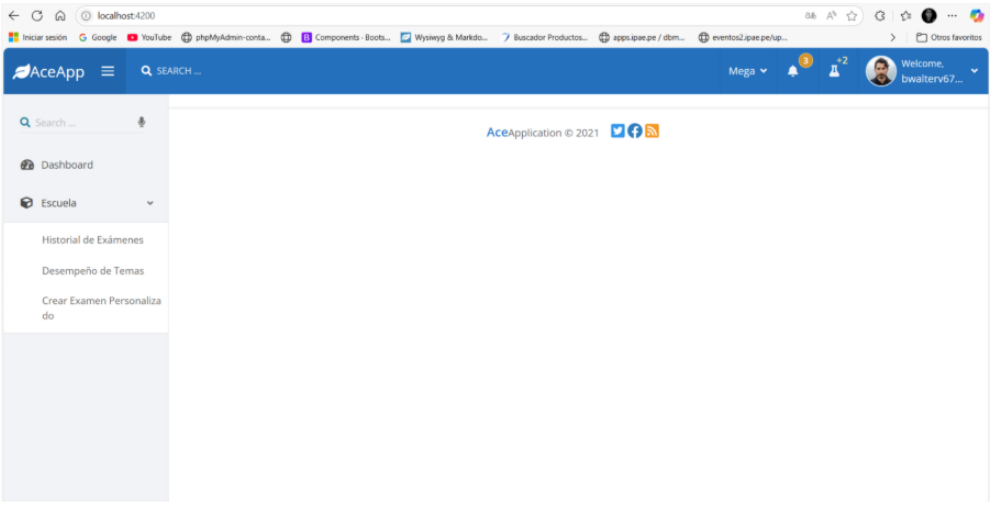


4.5.6.2 Vista Principal del Estudiante

- **Barra superior:** logo institucional, buscador, menú contextual "Mega", notificaciones y acceso al perfil.
- **Menú lateral:** accesos rápidos a:
    - Dashboard.
    - Historial de Exámenes.
    - Desempeño de Temas.
    - Crear Examen Personalizado.
- Esta vista centraliza la navegación y define la experiencia principal del usuario.

Figura 25

Dashboard inicial del Alumno

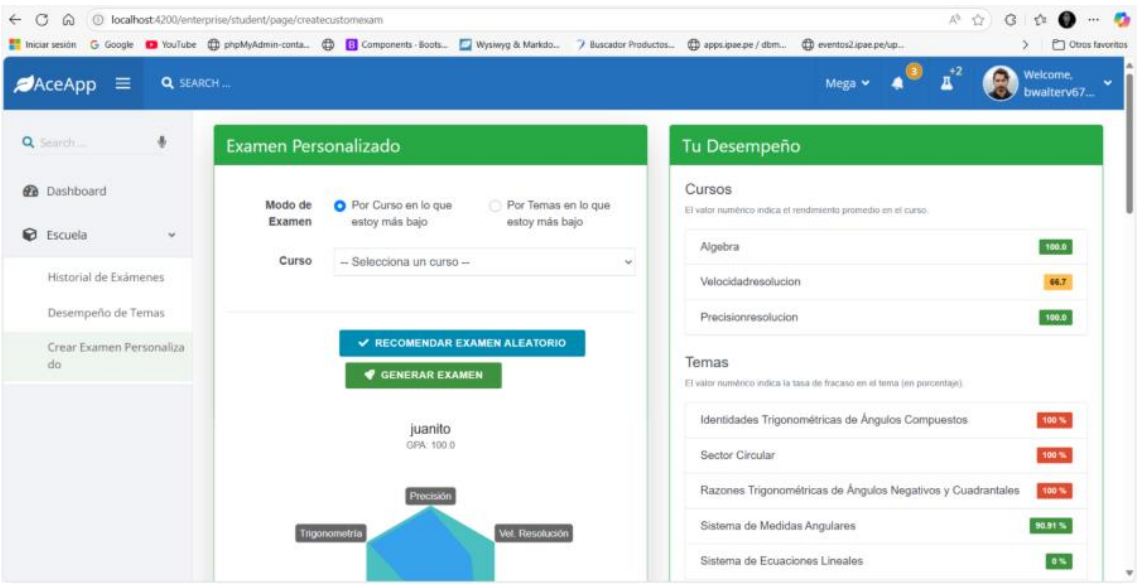


4.5.6.3 Página "Crear Examen Personalizado"

- Selección de **modo de examen**: por curso o por tema.
- Selector desplegable de curso.
- Botón **"Recomendar examen aleatorio"** que consulta los modelos de ranking de ejercicios.
- Botón **"Generar examen"** para iniciar la creación del examen en ExamService.

Figura 26

Generación de Exámenes Personalizados



4.5.6.4 Panel “Examen Personalizado”

- Visualiza el nombre del alumno y su GPA.
- **Gráfico de radar** que muestra el nivel en Álgebra, Aritmética, Trigonometría, Precisión, Velocidad de resolución y Geometría.
- **Barras de progreso** con porcentaje de aciertos por curso.
- Opciones de generación de examen (curso/tema) conectadas al repositorio ExamExercisesRepository.

Figura 27

Panel de Examen Personalizado con Estadísticas

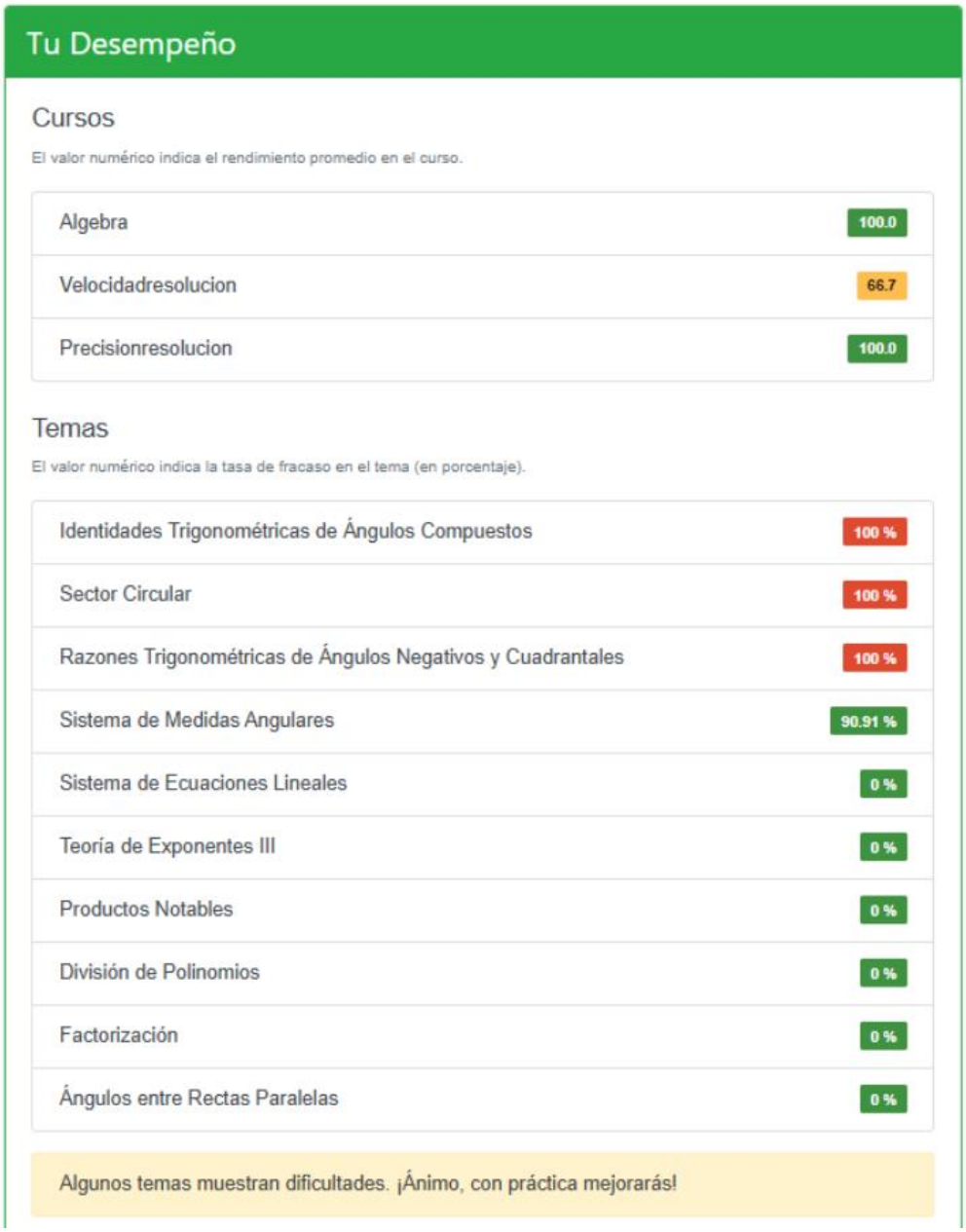


4.5.6.5 Panel “Tu Desempeño”

- **Lista de cursos:** promedio de aciertos por curso.
- **Lista de temas:** tasa de error (%) por tópico específico.
- **Etiquetas de colores:** verde (fortalezas), rojo/naranja (áreas críticas).
- **Mensaje motivacional dinámico**, generado según el desempeño del estudiante.

Figura 28

Retroalimentación de Desempeño Académico

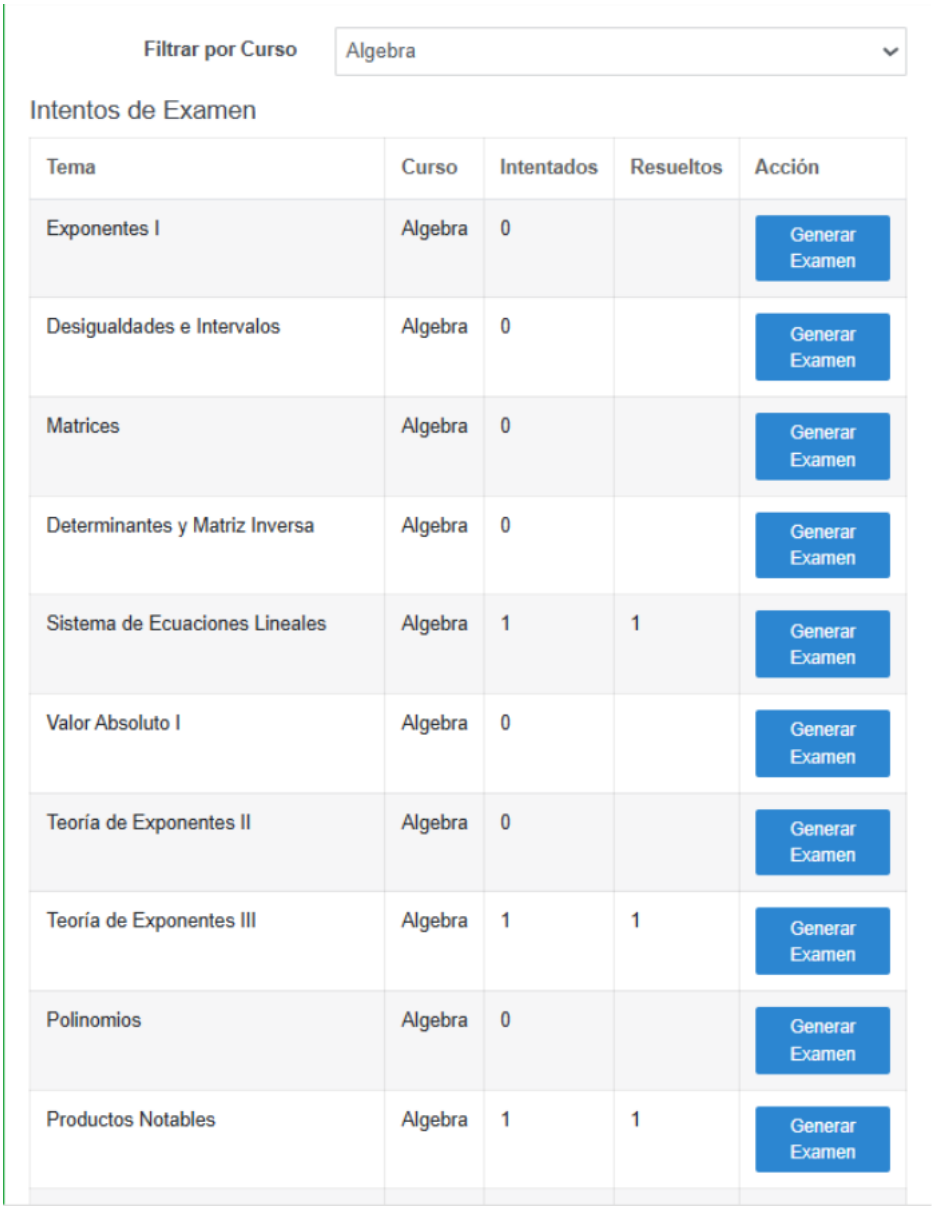


4.5.6.6 Lista de Intentos de Examen por Tema

- Filtro "**Filtrar por Curso**" para personalizar la vista.
- **Tabla de intentos** con columnas: Tema, Curso, Número de Intentos, Resultados y Acción.
- Botón **"Generar Examen"** en cada fila que crea un nuevo intento basado en ese tópico.

Figura 29

Historial de Intentos por Tema



4.5.6.7 Vista de Resolución de Examen

- Encabezado: nombre del examen, código del estudiante y duración asignada.
- Para cada ejercicio:
    - Enunciado, opciones, dificultad y puntaje.
    - Campo de texto para respuesta.
    - Botón **"Confirmar"** que valida y registra en el historial (data_student_exam_history).
- Preguntas respondidas se bloquean mostrando la respuesta enviada.

Figura 30

Ejecución de un Examen Personalizado

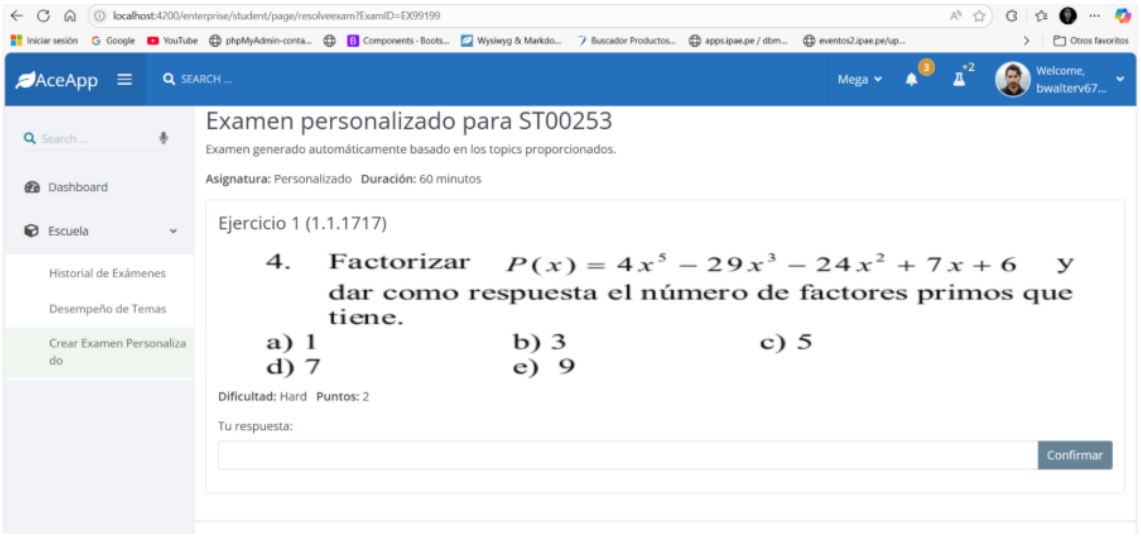
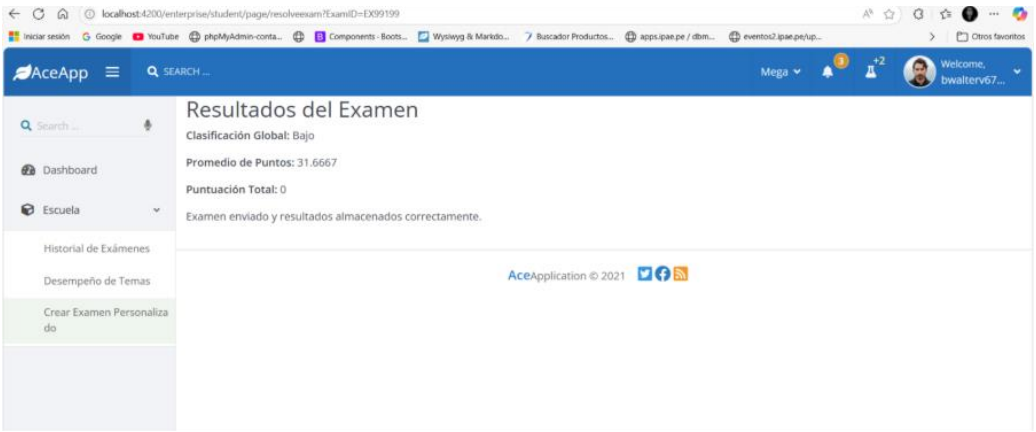


Figura 31

Resultados de examen personalizado

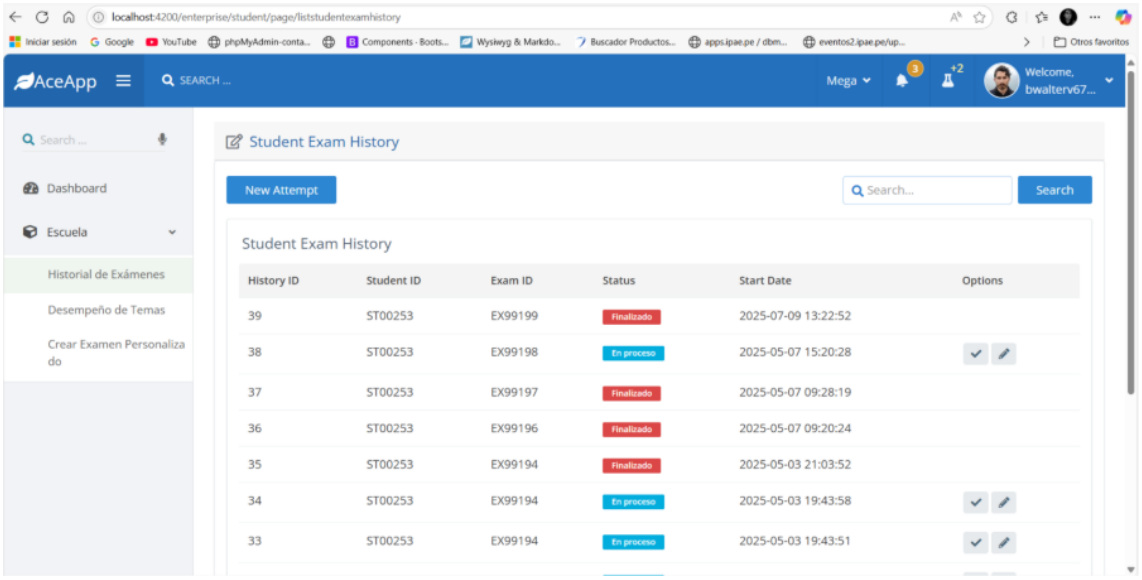


4.5.6.8 Historial de Exámenes del Estudiante

- Botón **"New Attempt"** para iniciar un nuevo examen.
- **Tabla de historial** con columnas:
    - History ID.
    - Student ID.
    - Exam ID.
    - Status (En proceso / Finalizado).
    - Start Date.
    - Options: continuar intento o ver detalles.

Figura 32

Registro de Intentos de Exámenes



4.6. Evaluación de la efectividad del STI

4.6.1. Nivel de conocimiento

Tras la implementación del Sistema Tutor Inteligente, se aplicó una prueba final con la misma estructura y cantidad de ítems que el diagnóstico inicial (20 preguntas sobre exponentes, matrices, polinomios, productos notables y factorización). El total de

estudiantes evaluados fue nuevamente de 82, con el fin de comparar la evolución de los resultados.

Tabla 22

Distribución del Nivel de Conocimiento según Calificaciones Obtenidas

Rango	Cantidad	%	Nivel
0 a 10	14	17.07%	Bajo
11 a 15	39	47.56%	Medio
16 a 20	29	35.37%	Alto
Total	82	100%	

La comparación entre ambos diagnósticos evidencia una reducción drástica de estudiantes en nivel bajo (de 63.41% a 17.07%), mientras que el nivel alto creció de 4.88% a 35.37%. Esto refleja un impacto positivo del STI, al fortalecer competencias clave en álgebra mediante ejercicios adaptativos, retroalimentación personalizada y seguimiento continuo.

4.6.2. Análisis de desempeño por tema

Tabla 23

Desempeño del Tema 1: Exponentes I, II y III

Tema I	Correctas	C%	Incorrectas	I%
P1	72	35.29%	10	12.35%
P2	64	31.37%	18	22.22%
P3	60	29.41%	22	27.16%
P4	68	33.33%	14	17.28%
Total	264	100%	64	100%

En exponentes se alcanzó una mejora significativa: el porcentaje de errores bajó del 48.90% al 19.51%. Aunque persisten dificultades en P3 (27.16% de errores), el progreso general evidencia que los estudiantes lograron dominar mejor la aplicación de leyes de potencias y exponentes fraccionarios.

Tabla 24

Desempeño del Tema 2: Matrices

Tema II	Correctas	C%	Incorrectas	I%
P5	59	31.05%	23	25.00%
P6	68	35.79%	14	15.22%
P7	57	30.00%	25	27.17%
P8	61	32.11%	21	22.61%
Total	245	100%	83	100%

En matrices, los errores descendieron del 51.39% al **25.29%**, aunque se mantuvo cierta dificultad en P7 (27.17%). Esto indica que, si bien los estudiantes mejoraron en operaciones básicas, aún requieren reforzar la multiplicación de matrices y la interpretación de problemas aplicados.

Tabla 25

Desempeño del Tema 3: Polinomios

Tema III	Correctas	C%	Incorrectas	I%
P9	63	32.98%	19	20.43%
P10	57	29.84%	25	26.88%
P11	41	21.50%	41	44.09%
P12	60	31.68%	22	23.66%
Total	221	100%	107	100%

En polinomios se redujo el error de 55.17% a **32.62%**, aunque P11 sigue siendo la pregunta con mayor dificultad (44.09% de errores). Esto muestra un avance importante en identificación de términos semejantes y factorización básica, pero aún con retos en expresiones de mayor complejidad.

Tabla 26

Desempeño del Tema 4: Productos Notables

Tema IV	Correctas	C%	Incorrectas	I%
P13	48	25.13%	34	28.81%
P14	62	32.46%	20	16.95%
P15	52	27.23%	30	25.42%
P16	59	30.18%	23	19.83%
Total	221	100%	107	100%

El porcentaje de errores en productos notables bajó de 65.54% a 32.62%. Pese a la mejora, las preguntas P13 y P15 siguen generando confusión, lo que indica que algunos estudiantes aún aplican fórmulas algebraicas de manera mecánica.

Tabla 27

Desempeño del Tema 5: Factorización

Tema V	Correctas	C%	Incorrectas	I%
P17	41	26.28%	41	27.15%
P18	47	30.13%	35	23.18%
P19	55	35.26%	27	17.88%
P20	68	43.31%	14	9.79%
Total	211	100%	117	100%

Factorización, que fue el punto más crítico en el diagnóstico inicial (76.83% de errores), mejoró de manera considerable, reduciendo los errores a 35.67%. Sin embargo, P17 sigue mostrando dificultades (27.15%), lo que sugiere que algunos alumnos no logran consolidar completamente las técnicas de trinomios y factorización por agrupación.

4.6.3. Análisis de diagnóstico final

El diagnóstico final evidencia una mejoría significativa en el nivel global de aprendizaje**.** Mientras que en la evaluación inicial más del 63% se encontraba en nivel bajo, en la final esta cifra descendió a 17.07%. Los niveles medio y alto aumentaron, destacando que más de un tercio de los estudiantes alcanzó un desempeño sobresaliente.

Por temas, los errores disminuyeron en todos los casos: Exponentes (de 48.90% a 19.51%), Matrices (de 51.39% a 25.29%), Polinomios (de 55.17% a 32.62%), Productos Notables (de 65.54% a 32.62%) y Factorización (de 76.83% a 35.67%). Aunque persisten dificultades en ítems específicos (P3, P7, P11, P13 y P17), los resultados reflejan que el STI tuvo un impacto positivo y medible en el proceso de aprendizaje, confirmando su efectividad para personalizar la enseñanza y superar debilidades iniciales.

V. DISCUSIÓN DE RESULTADOS

La mejora observada en el nivel de conocimiento de los estudiantes mediante la implementación del Sistema Tutor Inteligente (STI) coincide con los hallazgos de Mora (2022) y Mamani (2020), quienes reportaron que los sistemas tutores adaptativos permitieron un aprendizaje más personalizado y progresivo, ajustando la dificultad de los ejercicios según el rendimiento individual de los estudiantes y promoviendo el desarrollo de competencias específicas. De manera similar, Benavides (2019) evidenció que la integración de inteligencia artificial en un STI optimizó el aprendizaje en matemáticas, fortaleciendo habilidades cognitivas y técnicas, lo que respalda los resultados obtenidos en el presente estudio, donde se redujo la proporción de estudiantes en nivel bajo y se incrementó el número de alumnos con desempeño alto. Estos hallazgos sugieren que la personalización de la enseñanza mediante algoritmos adaptativos y de análisis de desempeño contribuye de forma efectiva a superar las dificultades iniciales y a consolidar la progresión académica, validando la efectividad del STI como herramienta educativa para fortalecer el aprendizaje matemático de manera individualizada y progresiva.

El diseño y funcionamiento del algoritmo adaptativo desarrollado en este estudio coincide con lo reportado por Mora (2022) y Mamani (2020), quienes enfatizaron que los sistemas tutores inteligentes capaces de ajustar la dificultad de los ejercicios según el rendimiento individual favorecen un aprendizaje personalizado y progresivo, atendiendo las necesidades específicas de cada estudiante. Los resultados obtenidos evidenciaron que el algoritmo permitió focalizar los contenidos en las áreas con mayor dificultad, reforzando competencias algebraicas esenciales y contribuyendo a la disminución significativa del porcentaje de estudiantes en nivel bajo. Este hallazgo respalda la relevancia de emplear algoritmos adaptativos como mecanismo para optimizar la

enseñanza individualizada, facilitando la retroalimentación inmediata y promoviendo un progreso académico más consistente entre los estudiantes.

El desarrollo del algoritmo de análisis de desempeño en este estudio se alinea con los hallazgos de Romero (2024), Mora (2022), Mamani (2020), Benavides (2019), Rosas y Sagastegui (2022), Montenegro y Ríos (2020), Saavedra (2024), Céspedes (2021) y Paz (2021), quienes destacaron que los sistemas tutores inteligentes que generan evaluaciones temáticas y rankings personalizados facilitan la identificación de áreas de dificultad y permiten planificar intervenciones educativas focalizadas. Los resultados obtenidos mostraron que este algoritmo posibilitó priorizar los contenidos de refuerzo según las debilidades individuales de cada estudiante, optimizando la selección de ejercicios y favoreciendo un aprendizaje progresivo y adaptativo. La utilización de puntajes compuestos y clasificación de tópicos permitió no solo medir el rendimiento por tema, sino también brindar retroalimentación inmediata y precisa, fortaleciendo las competencias matemáticas y promoviendo un seguimiento detallado del progreso académico de los alumnos.

La integración de los algoritmos adaptativo y de análisis de desempeño dentro del entorno funcional del STI concuerda con los hallazgos de Romero (2024), Mora (2022), Mamani (2020), Benavides (2019), Rosas y Sagastegui (2022), Montenegro y Ríos (2020), Saavedra (2024), Céspedes (2021) y Paz (2021), quienes evidenciaron que la combinación de diferentes módulos dentro de un tutor inteligente permite ajustar dinámicamente el contenido educativo según el desempeño del estudiante y generar ejercicios adaptativos en tiempo real. En este estudio, dicha integración facilitó la personalización de la enseñanza, ya que el sistema utilizó la información del ranking temático y del rendimiento individual para ofrecer ejercicios que reforzaban las áreas más débiles, proporcionar retroalimentación inmediata y visualizar gráficas de habilidades y

progreso por curso y tema. Esta estrategia contribuyó a consolidar el aprendizaje, optimizar la planificación pedagógica y mantener un seguimiento detallado del avance académico de cada estudiante, promoviendo así un proceso de enseñanza más efectivo y centrado en las necesidades individuales.

La evaluación de la efectividad del STI confirma los resultados reportados por Romero (2024), Mora (2022), Mamani (2020), Benavides (2019), Rosas y Sagastegui (2022), Montenegro y Ríos (2020), Saavedra (2024), Céspedes (2021) y Paz (2021), quienes encontraron que los sistemas tutores inteligentes mejoran el aprendizaje mediante la personalización de contenidos y la retroalimentación adaptativa. En el presente estudio, la prueba posterior evidenció un notable incremento en el aprendizaje, reduciendo el porcentaje de estudiantes en nivel bajo a 17% y aumentando los niveles medio y alto, mientras que los errores por tema descendieron de manera considerable en Exponentes, Matrices y Factorización. Estos resultados indican que la combinación de los algoritmos adaptativo y de análisis de desempeño permitió reforzar eficazmente las áreas con mayor dificultad, consolidar competencias matemáticas y favorecer la progresión académica individual de los estudiantes, demostrando que la integración de estas estrategias dentro del STI contribuye a una enseñanza más personalizada y eficiente.

VI. CONCLUSIONES

La implementación del Sistema Tutor Inteligente (STI) permitió fortalecer el aprendizaje matemático de los estudiantes de tercer grado de secundaria, evidenciando una mejora notoria en el nivel de conocimiento, pasando de un 63.41% de estudiantes en nivel bajo en el diagnóstico inicial a solo 17.07% en la evaluación final, mientras que los estudiantes en nivel alto aumentaron de 4.88% a 35.37%, así reflejaron que el STI logró personalizar la enseñanza, focalizando los contenidos en las áreas con mayores dificultades y promoviendo la adquisición de competencias algebraicas de manera progresiva y adaptativa.

Se diseñó un algoritmo adaptativo que ajustó el nivel de enseñanza en función del rendimiento individual de cada estudiante, donde utilizó los resultados de los ejercicios y pruebas previas para modificar la dificultad de las preguntas y ejercicios mostrados, priorizando aquellas áreas en las que el estudiante presentó errores o bajo desempeño, asegurando un aprendizaje progresivo y personalizado.

Se desarrolló un algoritmo de análisis de desempeño que generó un ranking temático personalizado para cada estudiante y evaluó los aciertos y errores por tema, calculó puntajes compuestos (CompositeScore) y clasificó los tópicos de mayor dificultad, permitiendo al sistema seleccionar automáticamente los contenidos de refuerzo más pertinentes según las debilidades individuales.

Se integraron ambos algoritmos dentro del entorno funcional del tutor inteligente, permitiendo la generación dinámica de ejercicios y contenidos de aprendizaje, asegurando que el sistema utilizara la información del desempeño y el ranking temático para ajustar en tiempo real los ejercicios presentados, ofreciendo retroalimentación inmediata y visualizando gráficas de habilidades y progreso por curso y tema.

Se evaluó la efectividad del SIT mediante una prueba posterior, mostrando una mejora en el aprendizaje con la disminución del porcentaje de estudiantes en nivel bajo a 17%, mientras que los niveles medio y alto aumentaron notablemente, además los errores por tema se redujeron considerablemente en Exponentes con 19,5%, en Matrices con 25,3% y en Factorización con 35,7%, lo que evidencia que la integración de los algoritmos adaptativo y de análisis de desempeño permitió reforzar eficazmente las áreas con mayor dificultad y favorecer la progresión académica de los estudiantes.

VII. RECOMENDACIONES

Se recomienda al Ministerio de Educación continuar promoviendo el uso de Sistemas Tutores Inteligentes en los colegios, capacitando a los docentes y brindando soporte técnico a las instituciones, con el fin de garantizar que más estudiantes se beneficien de herramientas que personalicen la enseñanza y refuercen las competencias matemáticas de manera progresiva y adaptativa

Se sugiere a los docentes aprovechar el algoritmo adaptativo para ajustar constantemente la dificultad de los ejercicios según el rendimiento de cada estudiante, revisando los resultados y orientando el refuerzo en áreas donde se identifiquen debilidades persistentes

Se recomienda a los docentes y coordinadores académicos utilizar el ranking temático generado por el algoritmo de análisis de desempeño para planificar estrategias de refuerzo personalizadas, enfocándose en los tópicos donde los estudiantes muestran mayores dificultades y promoviendo ejercicios de práctica adicionales

Se aconseja al equipo de soporte técnico y desarrolladores del sistema mantener y actualizar la integración de los algoritmos, garantizando la generación dinámica de ejercicios y contenidos, así como la correcta visualización de gráficas de progreso, para que los docentes y estudiantes puedan aprovechar toda la información en tiempo real

Se sugiere a los padres y tutores motivar a los estudiantes a utilizar el STI de manera constante, observando el progreso reflejado en las evaluaciones y gráficos, mientras que los docentes deben continuar monitoreando los resultados y ajustando las intervenciones pedagógicas para consolidar el aprendizaje logrado, especialmente en los temas donde aún se registran errores relevantes

VIII. REFERENCIAS

Ampuero R., N. (2022). Enseñanza aprendizaje: Síntesis del análisis conceptual desde el enfoque centrado en procesos. Venezuela: https://www.redalyc.org/journal/280/28073815009/28073815009.pdf.

Andreas Hippderks, Martin Schrepp, & Jörg Thomaschewski. (2014). Applying the User Experience Questionnaire (UEQ) in Different Evaluation Scenarios. *DOI: 10.1007/978-3-319-07668-3_37.*

Arce, M., Conejo, L., & Muñoz, J. (2019). *Aprendizaje y enseñanza de las matemáticas.* Madrid: Síntesis. https://bit.ly/3Bby0CV

Benavides C., A. (2019). Diseño de un software inteligente para mejorar el aprendizaje en estudiantes de la institución educativa Flores de Vila, Lima-2019. Lima: https://repositorio.cientifica.edu.pe/bitstream/handle/20.500.12805/893/TB-Benavides%20A-Ext.pdf?sequence=1&isAllowed=y.

Bettina Laugwitz, Theo Held, & Martin Schrepp. (2008). Construction and evaluation of a user experience questionnaire. *USAB*, LNCS 5298, pag. 63-779.

Cespedes S., M. (2021). Sistema tutor inteligente para asistir el aprendizaje de la asignatura de inglés en el tercer grado de secundaria de la I.E. "Santa Lucia". https://tesis.usat.edu.pe/bitstream/20.500.12423/4063/1/TL_CespedesSegundoMartha.pdf.

Chacón, D., Rodríguez, A., & Burguet, I. (2020). Aprendizaje basado en problemas para la enseñanza de la matemática en un entorno virtual de aprendizaje. Serie Científica de La Universidad de Las Ciencias Informáticas, 13(12), 191–201.

Devia, R., & Pinilla, C. (2012). La enseñanza de la matemática de la formación al trabajo del aula. *Redalyc*, 361-371. https://bit.ly/32qsdFF

Espinosa C., P. (2024). Problemas comunes en el aprendizaje en el área de las matemáticas. DOI: https://doi.org/10.23857/dc.v10i1.3703.

González J., P. (2024). La inteligencia artificial "ChatBot" y el proceso enseñanza-aprendizaje de la Física. Ecuador: https://repositorio.uta.edu.ec/server/api/core/bitstreams/10db7e65-f688-4cbd-ad78-4288a8645880/content.

Guamán C., L., Chucho R., J., Chucho M., J., Pailiacho A., D., & Inga C., W. (2025). Implementación de Sistemas de Tutoría Inteligente Basados en IA para la Personalización del Aprendizaje en Matemáticas. DOI: https://doi.org/10.37811/cl_rcm.v8i6.15792.

Mamani, A. (2020). Tutor Inteligente para el aprendizaje de matemáticas en sexto de secundaria. El Alto, Bolivia: https://repositorio.upea.bo/jspui/bitstream/123456789/202/1/TESIS-ALFREDO%20MAMANI%20HUALLPA.pdf.

Martin Scherepp, & Jörg Thomaschewski. (2019). Eine modulare Erweiterung des User Experience Questionnaire. *Mensch und Computer 2019 – Usability Professionals*, (págs. 148-56). Hamburgo.

Ministerio de educación Ecuador. (2018). *Área Matemática.* Ecuador: Ministerio de Educación. https://bit.ly/3POpNbR

Montenegro C., N., & Ríos R., J. (2020). Tutor inteligente con realidad aumentada para mejorar la comprensión lectora de los estudiantes de cuarto grado en una institución educativa, 2020 . Trujillo:

https://repositorio.ucv.edu.pe/bitstream/handle/20.500.12692/50555/Montenegro_CNL-R%c3%adoz_RJG-SD.pdf?sequence=1&isAllowed=y.

Mora C., G. (2022). Diseño de un protipo de tutor inteligente Para aprendizaje en lenguajes de programación . Ecuador: http://repositorio.ucsg.edu.ec/bitstream/3317/18434/1/T-UCSG-PRE-ING-CIC-18.pdf.

Ocaña F., Y., Valenzuela F., L., & Garro A., L. (2019). Inteligencia artificial y sus implicaciones en la educación superior . http://www.scielo.org.pe/pdf/pyr/v7n2/a21v7n2.pdf.

OCDE. (2019). Informe PISA 2018. http://umc.minedu.gob.pe/resultadospisa2018/.

Paz V., N. (2021). Sistema tutor intiligente para reforzar el aprendizaje de geometría en estudiantes de una I.E.P. secundaria del distrito Tumán. https://tesis.usat.edu.pe/bitstream/20.500.12423/4065/1/TL_PazVilchezNoemi.pdf.

Rios, B., & Portugal, W. (2019). Análisis de Tutores Inteligentes como sustento en la Universidad Mayor de San Marcos. *Scielo*, 1-10. https://bit.ly/3OxMcZI

Rodríguez C., M. (2021). Sistemas de tutoría inteligente y su aplicación en la educación superior . https://www.scielo.org.mx/pdf/ride/v11n22/2007-7467-ride-11-22-e015.pdf.

Rosario Snachis-Font, María Castr-Bleda, Carlos Turró, & Ignacio Despujol. (2018). Integración del "User Experience Questionnaire Short en MOOCs UPV. *Congreso Nacional de innovación educativa y de docencia en red.* Valencia: Universidad Politécnica de Valencia.

Rosas R., F., & Sagastegui C., P. (2022). Tutor inteligente para mejorar el aprendizaje de contenido musical en el Conservatorio Carlos Valderrama de Trujillo, 2021 . Trujillo: https://repositorio.ucv.edu.pe/bitstream/handle/20.500.12692/85250/Rosas_RFE-Sagastegui_CPB-SD.pdf?sequence=1&isAllowed=y.

Saavedra A., C. (2024). Sistema tutor inteligente para fortalecer el aprendizaje de comprensión lectora, en el segundo grado de primaria de la I.E. 11050 Rómulo Cabrera Chilcon. https://tesis.usat.edu.pe/bitstream/20.500.12423/7423/1/TL_SaavedraApagueoCheril.pdf.

Sierra, E., García, R., Cataldi, Z., & Hossian, A. (2015). Fundamentos para una metodología de siseño de sistemas tutoriales inteligentes centrada en la reparación de mecanismos. *Core*, 1-15. https://bit.ly/3BfhjWY

Villamizar, G., Araujo, T., & Trujillo, W. (2020). Relación entre ansiedad matemática y rendimiento académico en matemáticas en estudiantes de secundaria. Ciencias Psicológicas, 14(1), 1–13.

Yusef Hassan, & Martín Fernández. (2005). La experiencia de usuario. *no solo usabilidad: revista sobre personas, diseño y tecnología*.

IX. ANEXOS

ANEXO 1: CUESTIONARIO DE LA USABILIDAD DEL SISTEMA TUTOR INTELIGENTE

Estimado estudiante:

Queremos conocer tu experiencia utilizando el Sistema Tutor Inteligente. Tu opinión nos ayudará a mejorar su usabilidad y efectividad. Por favor, responde con sinceridad.

Instrucciones:

A continuación, encontrarás una serie de preguntas sobre el Sistema Tutor Inteligente. Marca con una (X) la opción que mejor represente tu opinión:

- 1 = Totalmente en desacuerdo
- 2 = En desacuerdo
- 3 = Neutral
- 4 = De acuerdo
- 5 = Totalmente de acuerdo

N°	Preguntas	1	2	3	4	5
1	El sistema tutor inteligente me ayuda a comprender mejor los temas del curso de algebra					
2	Con el sistema tutor inteligente, puedo resolver mis dudas de manera más rápida y efectiva					
3	Usar el sistema tutor inteligente me permite organizar mejor mi tiempo para estudiar					
4	El sistema tutor inteligente me apoya a estudiar y reforzar mis conocimientos					
5	Siento que mi rendimiento académico mejora con el uso del sistema de tutor inteligente					
6	La información que proporciona el sistema tutor inteligente es clara y fácil de entender					
7	Me resulta útil usar el sistema de tutor inteligente como apoyo en mis tareas y exámenes					
8	Considero que el contenido del sistema tutor inteligente está alineado con lo que aprendo en clase					

ANEXO 2 – CUESTIONARIO DE CONOCIMIENTO DEL CURSO DE ÁLBEGRA

Estimado estudiante:

Este cuestionario ha sido diseñado con el propósito de evaluar tu nivel de conocimiento en los temas fundamentales del curso de Álgebra, incluyendo Exponentes, Polinomios, Productos Notables, División de Polinomios y Factorización. Las preguntas están organizadas por tema y cada una cuenta con cuatro opciones de respuesta, de las cuales solo una es la correcta.

Instrucciones:

- Lee atentamente cada pregunta antes de seleccionar tu respuesta.
- Marca con una [X] la alternativa que consideres correcta.
- No dejes ninguna pregunta sin responder.
- Responde de manera individual y con honestidad para conocer tu verdadero nivel de comprensión.
- Una vez finalizado, revisa tus respuestas antes de entregar el cuestionario.

Preguntas

TEMA 1 – EXPONENTES I, II y III

1. Simplificar

$$R = 64^{-8^{-27^{-\frac{1}{3}}}}$$

a) 8 b) 4 c) 1/8 d) 1/4 e) 6

2. Simplificar

$$F = \frac{75^{10}\ x\ 24^{16}\ x\ 18^{22}}{36^{24}\ x\ 10^{20}\ x\ 9^{11}}$$

a) 3 b) 1 c) 2 d) 4 e) 5

3. Calcular

$$\frac{2^{x+4} + 2^{x+3} + 2^{x+2} + 2^{x+1}}{2^{x-4} + 2^{x-3} + 2^{x-2} + 2^{x-1}}$$

a) 21 b) 32 c) 13 d) 44
e) 55

4. Simplificar

$$\sqrt[ma]{\frac{20^{ma+1}}{2^{2ma+4} + 2^{2ma+2}}}$$

a) 2 b) 4 c) 5 d) 10
e) NA

TEMA 2 – MATRICES

5. Dadas las siguientes determinantes de orden 2, indica cuál es el resultado correcto

$$C = \begin{bmatrix} 7 & 6 \\ 2 & 3 \end{bmatrix}$$

a) -9 b) -33 c) 33 d) 9
e) 11

6. Dadas las siguientes determinantes de orden 3, indica cuál es el resultado correcto

$$F = \begin{bmatrix} 2 & 1 & 4 \\ 3 & 5 & 8 \\ 4 & 7 & 6 \end{bmatrix}$$

a) 62 b) -34 c) 15 d) 7
e) 2

7. Dada las siguientes matrices $A = \begin{bmatrix} 2 & 3 & 1 \\ 4 & 6 & -1 \\ 5 & 0 & 9 \end{bmatrix}$ y $B = \begin{bmatrix} 6 & 2 & 1 \\ -3 & 8 & -5 \\ 2 & 12 & 7 \end{bmatrix}$

Resolver: 3 . A + 2 .B

a) $\begin{bmatrix} 12 & 13 & 5 \\ -18 & 34 & -13 \\ 19 & 24 & 41 \end{bmatrix}$ b) $\begin{bmatrix} 18 & 13 & 5 \\ 6 & 34 & -13 \\ 19 & 24 & 41 \end{bmatrix}$ c)

$\begin{bmatrix} 12 & 13 & 5 \\ -6 & 34 & -7 \\ 19 & 0 & 41 \end{bmatrix}$ d)

$\begin{bmatrix} 20 & 23 & -5 \\ 5 & 14 & -13 \\ 12 & 4 & 21 \end{bmatrix}$ e) $\begin{bmatrix} -15 & 15 & 8 \\ 7 & 14 & -8 \\ -14 & 34 & 51 \end{bmatrix}$

8. Dada las siguientes matrices $D = \begin{bmatrix} 2 & 3 & 5 \\ 8 & 6 & -1 \\ 4 & 0 & 9 \end{bmatrix}$ y $E = \begin{bmatrix} 2 & 3 \\ 4 & 6 \\ 5 & 1 \end{bmatrix}$

Resolver: D . E

a) $\begin{bmatrix} 41 & 29 \\ 35 & 59 \\ 53 & 21 \end{bmatrix}$ b) $\begin{bmatrix} 41 & 29 \\ 40 & 60 \\ 57 & 27 \end{bmatrix}$ c) $\begin{bmatrix} 41 & 29 \\ -40 & -60 \\ 53 & 21 \end{bmatrix}$ d) $\begin{bmatrix} 41 & 17 \\ 25 & 19 \\ 13 & 11 \end{bmatrix}$ e)

$\begin{bmatrix} 24 & 29 \\ 15 & 29 \\ 23 & 21 \end{bmatrix}$

TEMA 3 – POLINOMIOS

9. Calcula la suma de coeficientes del polinomio:

$$G(x) = 8x^3 - 4x^2 + 8x - 6$$

a) 17 b) 11 c) 12 d) 10

e) 14

10. Dado: $E(x) = 7x^4 - 3x + 6 - x^2$; Halla E(1)

a) 8 b) 12 c) 9 d) 5 e) N.A.

11. Si el polinomio es de octavo grado, calcula el valor de "m"

$$F(x) = \sqrt{7}x^{m+5} + 2x^{m} - 0.2x^{m+6}$$

a) 1 b) -3 c) -1 d) 4
e) 2

12. Calcula el grado del siguiente término:

$$M(x, y) = -\frac{5}{3}\sqrt{7}x^{2}ywt$$

a) 1 b) 2 c) 3 d) 4 e)
No tiene

TEMA 4 - PRODUCTOS NOTABLES

13. Reduce

$$T = (x+2)^{3} - (x+3)(x+2)(x+1) - x$$

a) 1 b) 2 c) 3 d) 4
e) 5

14. Reduce:

$$V = (a+b+c)(a+b+d) - (a+c+d)(b+c+d) - (a+b+c+d)(a+b-c-d) - cd$$

a) 2a b) 4ab c) 0 d)$a^2 + c^2$ e)
-ab

15. Reduce

$$L = \frac{(x+2)(x+3)(x+4)(x+5) - (x^2+7x+11)^2}{(x^2+9x+19)^2 - (x+3)(x+5)(x+6)(x+4)}$$

a) 1 b) 1/2 c) 0 d) -1
e) -1/2

16. Simplifica:

$$C = \frac{(2m^5 + 3)^2 + (2m^5 - 3)^2}{4m^{10} + 9}$$

a) 6 b) 5 c) 4 d) 2
e) 16

TEMA 5 - FACTORIZACIÓN

17. Factorizar:

$$2x^3y + 3xy^2$$

a) $xy(2x^2 + 3y)$ b) $xy(2x + 3y)$ c) $xy(2x^2 + 3y^2)$ d) $xy(2x + 3)$
e) x^2

18. Factorizar:

$$4x^2 - 12x + 9$$

a) $(2x + 3)^2$ b) $(2x - 3)$ c) $(2x - 3)^2$ d) $(4x - 9)^2$
e) 2x

19. Factorizar:

$$\text{abc} + \text{ab} + \text{c} + 1$$

a) $(c + 1)(ab + 1)$ b) $(c + b)(ab + 1)$ c) $(c + 1)(a + b)$
d) $(c + a)(b + a + 1)$ e) $(a + b + c)$

20. Factorizar:

$$a^2 - 10a + 25$$

a) $(a+5)^2$ b) $(a-5)^2$ c) $(a+25)^2$ d) $(a-25)^2$ e) 5a