

UNIVERSIDAD PÚBLICA DE EL ALTO

CARRERA INGENIERÍA DE SISTEMAS



PROYECTO DE GRADO

**“TUTOR INTELIGENTE BASADO EN REDES NEURONALES
ARTIFICIALES PARA MEJORAR EL LOGRO DE
COMPETENCIAS EN EL INTERNADO ROTATORIO”
CASO: CARRERA DE MEDICINA DE LA UNIVERSIDAD PÚBLICA DE
EL ALTO**

**Para optar al título de Licenciatura en Ingeniería de sistemas
MENCIÓN: INFORMÁTICA Y COMUNICACIONES**

Postulante: Cecilio Poma Muñoz

Tutor Metodológico: M.Sc. Lic. Ing. Marisol Arguedas Balladares

Tutor Especialista: Lic. Margarita Bernarda Lopez Mariaca

Tutor Revisor: Lic. Ing. Miguel Ángel Pérez Bustillos

EL ALTO – BOLIVIA

2024

EDECLARACIÓN JURADA DE AUTENTICIDAD Y RESPONSABILIDAD

Yo **Cecilio Poma Muñoz con C.I. 7339585 OR** mediante la presente declaro de manera pública que la propuesta del **TRABAJO DE GRADO** titulada **“TUTOR INTELIGENTE BASADO EN REDES NEURONALES ARTIFICIALES PARA MEJORAR EL LOGRO DE COMPETENCIAS EN EL INTERNADO ROTATORIO” CASO: CARRERA DE MEDICINA DE LA UNIVERSIDAD PÚBLICA DE EL ALTO** es original, siendo resultado de mi trabajo personal y no constituye una copia o replica de trabajos similares elaborados, Autorizo la publicación del resumen de mi propuesta en internet y me comprometo a responder a todos los cuestionamientos que se desprenden de su lectura.

Asimismo, me hago responsable ante la universidad o terceros, de cualquiera irregularidad o daño que pudiera ocasionar, por el incumplimiento de lo declarado.

De identificarse falsificación, plagio, fraude, o que el **TRABAJO DE GRADO** haya sido publicado anteriormente; asumo las consecuencias y sanciones que de mi acción se deriven, responsabilizándome por todas las cargas legales que se deriven de ello sometiéndome a las normas establecidas y vigentes de la Carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

El Alto, noviembre de 2024



Cecilio Poma Muñoz
C.I. 7339585 OR
email: cpoma.m.is@gmail.com

DEDICATORIA

*A Dios, que siempre ha estado a mi lado para fortalecerme, cuidarme, enseñarme a no darme
por vencido nunca y que todo es posible si vamos de su mano.*

*A mis padres, por estar conmigo, por enseñarme a crecer y a que si caigo debo levantarme, por
apoyarme y guiarme, por ser las bases que me ayudaron a llegar hasta aquí.*

Cecilio Poma Muñoz

AGRADECIMIENTOS

Agradecer primeramente a Dios por la oportunidad estar hoy en este punto de mi vida.

*A mi familia por tener la paciencia como y la fuerza de seguir a mi lado, unida
apoyándome indudablemente.*

*A todos los docentes pertenecientes a mi estimada carrera Ingeniería de Sistemas, los cuales
dieron su dedicación plena en cada asignatura.*

Cecilio Poma Muñoz

ÍNDICE GENERAL

CAPÍTULO I	1
1. MARCO PRELIMINAR	1
1.1. INTRODUCCIÓN	1
1.2. ANTECEDENTES	2
1.2.1. Antecedentes Institucionales	2
1.2.2. Antecedentes Afines al Proyecto de Grado	4
1.2.2.1. Antecedentes Internacionales	4
1.2.2.2. Antecedentes Nacionales	5
1.3. PLANTEAMIENTO DEL PROBLEMA	5
1.3.1. Problema Principal	5
1.3.2. Problemas Secundarios	6
1.3.3. Formulación del Problema	7
1.4. OBJETIVOS	7
1.4.1. Objetivo General	7
1.4.2. Objetivos Específicos	7
1.5. JUSTIFICACIÓN	8
1.5.1. Justificación Técnica	8
1.5.2. Justificación Económica	8
1.5.3. Justificación Social	8
1.6. METODOLOGÍAS	9
1.6.1. Metodología CRISP-DM	9
1.6.2. Metodología MeISE	9
1.7. PRUEBAS DE SOFTWARE	9
1.7.1. Pruebas de Funcionalidad	9
1.7.2. Técnica de Caja Negra	10

1.7.3. Técnica de Caja Blanca.....	10
1.8. HERRAMIENTAS.....	10
1.8.1. Para Desarrollar el Sistema Web	10
1.8.2. Para Desarrollar el Tutor Inteligente	10
1.9. LIMITES Y ALCANCES.....	10
1.9.1. Limites.....	10
1.9.2. Alcances.....	11
1.10. APORTES	11
CAPÍTULO II	13
2. MARCO TEÓRICO.....	13
2.1. INTRODUCCIÓN	13
2.2. INGENIERÍA DE SOFTWARE	13
2.3. INGENIERÍA DE REQUERIMIENTOS	13
2.4. SISTEMA TUTOR INTELIGENTE (STI)	14
2.5. ARQUITECTURA DE UN STI	14
2.5.1. Módulo Tutor	15
2.5.2. Módulo Dominio	15
2.5.3. Módulo Estudiante.....	16
2.5.4. Interfaz	16
2.6. SOFTWARE DE INTELIGENCIA ARTIFICIAL	16
2.7. INTELIGENCIA ARTIFICIAL	17
2.7.1. Redes Neuronales.....	17
2.7.2. Deep Learning.....	19
2.7.3. Natural Language Processing	20
2.7.4. Tokenización	21

2.7.5. Embeddings	21
2.7.6. Transformer.....	23
2.7.7. Transfer Learning	25
2.7.8. Transfer Learning Basado en Modelos.....	27
2.7.9. Inteligencia Artificial Generativa	27
2.7.10. Language Large Model	27
2.7.11. Aplicaciones de los LLM	28
2.8. METODOLOGÍA DE INGENIERÍA DE SOFTWARE EDUCATIVO.....	30
2.8.1. Fase Conceptual	31
2.8.2. Análisis y Diseño Inicial	32
2.8.3. Plan de Iteraciones.....	33
2.8.4. Diseño Computacional	34
2.8.5. Desarrollo.....	35
2.8.6. Despliegue	36
2.9. METODOLOGÍA CRISP-DM.....	37
2.9.1. Fase de Comprensión del Negocio o Problema.....	38
2.9.2. Fase de Comprensión de Los Datos	39
2.9.3. Fase de Preparación de los Datos.....	40
2.9.4. Fase de Modelado.....	41
2.9.5. Fase de Evaluación.....	42
2.9.6. Fase de Implementación	43
2.10. TECNOLOGÍAS PARA EL DESARROLLO.....	44
2.10.1. Django	44
2.10.2. Django Rest Framework	44
2.10.3. Vue	45
2.10.4. Pytorch	46

2.11.	PRUEBAS DE SOFTWARE	47
2.11.1.	Pruebas Funcionales	47
2.11.2.	Pruebas de Integración.....	48
2.11.3.	Pruebas Unitarias	49
2.12.	MÉTRICAS DE CALIDAD.....	49
2.12.1.	ISO/IEC 25010	49
2.13.	MÉTRICAS DE SEGURIDAD.....	51
2.13.1.	ISO/IEC 27002	51
CAPÍTULO III		53
3.	MARCO APLICATIVO	53
3.1.	INTRODUCCIÓN	53
3.2.	FASE DE DEFINICIÓN	54
3.2.1.	Análisis de Necesidades Educativas	54
3.2.2.	Revisión de Alternativas de Solución.....	55
3.2.3.	Estudio de Riesgos	55
3.2.4.	Identificación de Funcionalidades del Software	55
3.2.5.	Criterios de Medición.....	66
3.2.6.	Determinación de los Objetivos del Negocio	66
3.2.7.	Determinación de los Objetivos de Data Mining	66
3.3.	ANÁLISIS, DISEÑO Y COMPRENSIÓN DE DATOS	66
3.3.1.	Requerimientos Funcionales y No Funcionales	67
3.3.2.	Establecimiento de la Arquitectura del Software.....	70
3.3.3.	Diseño Educativo y Diseño de Comunicación	70
3.3.4.	Recolección de Datos Iniciales.....	77
3.3.5.	Descripción de los Datos.....	77

3.3.6. Exploración de los Datos.....	77
3.4. PLANEACIÓN DE ITERACIONES Y PREPARACIÓN DE DATOS	77
3.4.1. Plan de Iteraciones.....	78
3.4.2. Selección de los Datos	80
3.4.3. Limpieza de los Datos	80
3.4.4. Integración de los Datos.....	81
3.4.5. Formateo de los Datos	81
3.5. DISEÑO COMPUTACIONAL Y MODELADO	83
3.5.1. Primera Iteración	84
3.5.2. Segunda Iteración	86
3.5.3. Tercera Iteración	89
3.5.4. Cuarta Iteración.....	91
3.5.5. Quinta Iteración	93
3.5.6. Sexta Iteración	95
3.5.7. Séptima Iteración	96
3.5.8. Selección de la Técnica de Modelado	96
3.5.9. Construcción del Modelo	97
3.6. DESARROLLO Y EVALUACIÓN	101
3.6.1. Primera Iteración	101
3.6.2. Segunda Iteración	102
3.6.3. Tercera Iteración	104
3.6.4. Cuarta Iteración.....	105
3.6.5. Quinta Iteración.....	106
3.6.6. Sexta Iteración	107
3.6.7. Séptima Iteración	108
3.6.8. Pruebas Primera Iteración.....	109

3.6.9. Pruebas Segunda Iteración	114
3.6.10. Pruebas Tercera Iteración	118
3.6.11. Pruebas Cuarta Iteración	121
3.6.12. Pruebas Quinta Iteración	122
3.6.13. Pruebas Sexta Iteración.....	123
3.6.14. Pruebas Séptima Iteración.....	125
3.6.15. Evaluación de Los Resultados Del Modelo	127
3.6.16. Proceso de Revisión	127
3.7. DESPLIEGUE	127
3.7.1. Evaluación de Satisfacción.....	128
3.7.2. Planificación de Nuevas Iteraciones	128
3.7.3. Implementación de la red neuronal.....	128
CAPÍTULO IV.....	130
4. CALIDAD, COSTO Y SEGURIDAD	130
4.1. INTRODUCCIÓN	130
4.2. MÉTRICA DE CALIDAD 25010.....	130
4.2.1. Rendimiento	130
4.2.2. Usabilidad	131
4.2.3. Confiabilidad	133
4.2.4. Funcionalidad.....	135
4.3. DETERMINACIÓN DE COSTOS	137
4.3.1. Licencias de Software	137
4.3.2. Herramientas de Desarrollo.....	137
4.3.3. Desarrollo de software.....	138
4.3.3.1. Estrategia de medición.....	138

4.3.3.2. Mapeo	138
4.3.3.3. Medición.....	140
4.4. SEGURIDAD DEL SOFTWARE.....	141
4.4.1. Control de Acceso.....	141
4.4.2. Seguridad en Procesos de Desarrollo	142
CAPÍTULO V.....	144
5. CONCLUSIONES Y RECOMENDACIONES	144
5.1. CONCLUSIONES	144
5.2. RECOMENDACIONES	145
REFERENCIAS.....	147
ANEXOS	148

ÍNDICE DE TABLAS

Tabla 1 <i>Áreas del Internado Rotatorio</i>	4
Tabla 2 <i>Actividades y Artefactos de la Fase Conceptual</i>	31
Tabla 3 <i>Actividades y Artefactos de la Fase de Análisis y Diseño Inicial</i>	32
Tabla 4 <i>Actividades y Artefactos de la Fase del Plan de Iteraciones</i>	34
Tabla 5 <i>Actividades y Artefactos de la Fase de Diseño Computacional</i>	34
Tabla 6 <i>Actividades y Artefactos de la Fase de Desarrollo</i>	35
Tabla 7 <i>Actividades y Artefactos de la Fase de Despliegue</i>	36
Tabla 8 <i>Combinación de la metodología MeISE y CRISP-DM</i>	53
Tabla 9 <i>Especificaciones de Caso de Uso Expandido Inicio de sesión</i>	57
Tabla 10 <i>Especificaciones de Caso de Uso Módulo Internado Rotatorio</i>	59
Tabla 11 <i>Especificaciones de Caso de Uso Expandido Progreso de estudio</i>	60
Tabla 12 <i>Especificaciones de Caso de Uso Expandido Módulo TutorAI</i>	62
Tabla 13 <i>Especificaciones de Caso de Uso Expandido Módulo estudiante</i>	63
Tabla 14 <i>Especificaciones de Caso de Uso expandido Módulo personal</i>	65
Tabla 15 <i>Requerimientos funcionales</i>	67
Tabla 16 <i>Cantidad de datos, textos PDFs</i>	77
Tabla 17 <i>Prueba funcional iniciar sesión con credenciales incorrectas</i>	109
Tabla 18 <i>Prueba funcional iniciar sesión con credenciales correctas</i>	110
Tabla 19 <i>Prueba de Integración, app authentication, endpoint n-login</i>	111
Tabla 20 <i>Prueba Unitaria, App authentication, verificar contraseña</i>	112
Tabla 21 <i>Prueba Funcional, Registrar estudiante</i>	114
Tabla 22 <i>Prueba de Integración, app estudiante, endpoint n-estudiante-list-create</i>	115
Tabla 23 <i>Prueba Unitaria, App estudiante, serializador</i>	116
Tabla 24 <i>Prueba funcional registrar personal institucional</i>	118
Tabla 25 <i>Prueba de Integración, app personal, endpoint n-personal-list-create</i>	119

Tabla 26 <i>Prueba funcional Internado Rotatorio área de medicina interna</i>	121
Tabla 27 <i>Prueba Funcional cargar material de estudio</i>	122
Tabla 28 <i>Prueba Funcional, cargar progreso de estudio</i>	123
Tabla 29 <i>Prueba Unitaria, App estudiante, listar progreso de estudio</i>	124
Tabla 30 <i>Prueba funcional interactuar con TutorAI (Red neuronal)</i>	125
Tabla 31 <i>Prueba Unitaria, Red Neuronal</i>	126
Tabla 32 <i>Métricas de Calidad, Rendimiento</i>	130
Tabla 33 <i>Encuesta, Métrica de Calidad de Rendimiento</i>	131
Tabla 34 <i>Métrica de Calidad, Usabilidad</i>	132
Tabla 35 <i>Encuesta, Métrica de Calidad de Usabilidad</i>	132
Tabla 36 <i>Métrica de Calidad Confiabilidad</i>	133
Tabla 37 <i>Encuesta, Métrica de Calidad Confiabilidad</i>	134
Tabla 38 <i>Métrica de Calidad Funcionalidad</i>	135
Tabla 39 <i>Encuesta, Métrica de Calidad Funcionalidad</i>	136
Tabla 40 <i>Herramientas de desarrollo</i>	137
Tabla 41 <i>Mapeo de Requerimientos Funcionales</i>	139

ÍNDICE DE FIGURAS

Figura 1 <i>Organigrama Carrera de Medicina</i>	3
Figura 2 <i>Arquitectura de un Sistema Tutor Inteligente</i>	15
Figura 3 <i>Neurona Biológica</i>	18
Figura 4 <i>Neurona Artificial</i>	19
Figura 5 <i>Diagrama de una red neuronal profunda</i>	20
Figura 6 <i>Word embeddings</i>	22
Figura 7 <i>Incrustar palabras (word embeddings) en vectores</i>	23
Figura 8 <i>Arquitectura de un transformer</i>	24
Figura 9 <i>Proceso de Transfer Learning</i>	25
Figura 10 <i>Inteligencia Artificial</i>	28
Figura 11 <i>Fine-tuning</i>	29
Figura 12 <i>Ciclo de vida de la Metodología MeISE</i>	30
Figura 13 <i>Modelo de procesos CRISP-DM</i>	37
Figura 14 <i>ISO/IEC 25010</i>	50
Figura 15 <i>Arquitectura STI del sistema</i>	55
Figura 16 <i>Caso de Uso de Alto Nivel</i>	56
Figura 17 <i>Caso de Uso Expandido Inicio de sesión</i>	56
Figura 18 <i>Caso de Uso Expandido Módulo Internado Rotatorio</i>	58
Figura 19 <i>Caso de Uso Expandido Módulo progreso de estudio</i>	60
Figura 20 <i>Caso de Uso Expandido Módulo TutorAI</i>	61
Figura 21 <i>Caso de Uso Expandido Modulo estudiante</i>	63
Figura 22 <i>Caso de Uso Expandido Modulo Personal institucional</i>	64
Figura 23 <i>Requerimientos no funcionales</i>	69
Figura 24 <i>Modelo de Navegación</i>	71
Figura 25 <i>Prototipo, Inicio de sesión</i>	72

Figura 26 <i>Prototipo, Vista inicio</i>	72
Figura 27 <i>Prototipo, Vista Internado Rotatorio</i>	73
Figura 28 <i>Prototipo, Vista área de estudio</i>	73
Figura 29 <i>Prototipo, Vista progreso de estudio</i>	74
Figura 30 <i>Prototipo, Vista TutorAI</i>	74
Figura 31 <i>Prototipo, Vista estudiantes</i>	75
Figura 32 <i>Prototipo, Vista formulario de estudiantes</i>	75
Figura 33 <i>Prototipo, vista personal institucional</i>	76
Figura 34 <i>Prototipo, formulario personal institucional</i>	76
Figura 35 <i>Plan de iteraciones</i>	78
Figura 36 <i>Prioridades de las Iteraciones</i>	79
Figura 37 <i>Función para limpieza de datos</i>	81
Figura 38 <i>Formateo de los datos</i>	82
Figura 39 <i>DataSet</i>	82
Figura 40 <i>Verificar y cargar el DataSet</i>	83
Figura 41 <i>Diagrama de clases primera iteración</i>	84
Figura 42 <i>Diagrama de secuencia primera iteración</i>	85
Figura 43 <i>Diagrama conceptual de la base de datos</i>	86
Figura 44 <i>Diagrama de clases segunda iteración</i>	87
Figura 45 <i>Diagrama de secuencia, segunda iteración</i>	88
Figura 46 <i>Diagrama de clases, tercera iteración</i>	89
Figura 47 <i>Diagrama de secuencia, tercera iteración</i>	90
Figura 48 <i>Diagrama de clases, cuarta iteración</i>	91
Figura 49 <i>Diagrama de clases cuarta iteración</i>	92
Figura 50 <i>Diagrama de clases quinta iteración</i>	93
Figura 51 <i>Diagrama de secuencia quinta iteración</i>	94

Figura 52 <i>Diagrama de secuencia sexta iteración</i>	95
Figura 53 <i>Diagrama de secuencia séptima iteración</i>	96
Figura 54 <i>Simulación de la red neuronal de forma tradicional</i>	97
Figura 55 <i>Arquitectura del modelo representado de forma tradicional</i>	98
Figura 56 <i>Preparar el modelo y configurar el tokenizador</i>	98
Figura 57 <i>Tokenizar el DataSet</i>	99
Figura 58 <i>Entrenamiento de la red neuronal</i>	99
Figura 59 <i>Embedding, espacio vectorial de los tokens</i>	100
Figura 60 <i>Embedding</i>	100
Figura 61 <i>Módulo Inicio de sesión</i>	101
Figura 62 <i>Interfaz perfil de usuario</i>	102
Figura 63 <i>Módulo Estudiante</i>	102
Figura 64 <i>Módulo Estudiante, ver información del registro</i>	103
Figura 65 <i>Módulo Estudiante, interfaz para crear nuevo registro</i>	103
Figura 66 <i>Módulo Institucional</i>	104
Figura 67 <i>Módulo Institucional, interfaz para crear nuevo registro</i>	104
Figura 68 <i>Modulo Institucional, interfaz para eliminar registro</i>	105
Figura 69 <i>Modulo Áreas del Internado Rotatorio</i>	105
Figura 70 <i>Sub Módulo Estudio</i>	106
Figura 71 <i>Sub Módulo Estudio, generación de cuestionario</i>	106
Figura 72 <i>Sub Módulo Estudio, evaluación de cuestionario</i>	107
Figura 73 <i>Módulo Progreso de estudio</i>	108
Figura 74 <i>Módulo TutorAI</i>	108

ÍNDICE DE ECUACIONES

Ecuación 1 <i>Ecuación de usabilidad</i>	133
Ecuación 2 <i>Ecuación de confiabilidad</i>	135
Ecuación 3 <i>Ecuación de esfuerzo</i>	140
Ecuación 4 <i>Ecuación de costos</i>	140

RESUMEN

El "Sistema Tutor Inteligente basado en redes neuronales para mejorar el logro de competencias en el Internado Rotatorio" es un proyecto diseñado para apoyar a los estudiantes de medicina durante su internado rotatorio. El objetivo principal del sistema es mejorar la comprensión y el rendimiento de los estudiantes a través del uso de inteligencia artificial, específicamente un modelo de lenguaje grande (LLM) ajustado con datos médicos. El sistema está compuesto por un frontend desarrollado en Vue.js y Vuetify, y un backend basado en Django Rest Framework. Su estructura se organiza en varias secciones, incluyendo un área de estudio que abarca diversas disciplinas médicas como Medicina Interna, Cirugía, Ginecología y Obstetricia, Pediatría, y Salud Pública, donde los estudiantes encuentran materiales y cuestionarios generados por la IA. Además, se incluye un asistente virtual basado en una red neuronal de procesamiento de lenguaje natural que interactúa con los estudiantes, respondiendo sus preguntas y proporcionando retroalimentación, y una sección que permite a los estudiantes ver su progreso en los cuestionarios y en su aprendizaje. En conjunto, el "Sistema Tutor Inteligente" busca optimizar el proceso de aprendizaje de los estudiantes de medicina, brindándoles un recurso tecnológico avanzado que integra inteligencia artificial con contenido médico.

Palabras clave: redes neuronales, llm, aprendizaje por transferencia, embeddings, tokenización, internado rotatorio.

ABSTRACT

The “Neural Network-Based Smart Tutoring System for Improving Competency Achievement in Rotating Internship” is a project designed to support medical students during their rotating internship. The main goal of the system is to improve students’ understanding and performance through the use of artificial intelligence, specifically a large language model (LLM) tuned with medical data. The system is composed of a frontend developed in Vue.js and Vuetify, and a backend based on the Django Rest Framework. Its structure is organized into several sections, including a study area covering various medical disciplines such as Internal Medicine, Surgery, Gynecology and Obstetrics, Pediatrics, and Public Health, where students find AI-generated materials and quizzes. Additionally, a virtual assistant based on a natural language processing neural network is included that interacts with students, answering their questions and providing feedback, and a section that allows students to view their progress on quizzes and in their learning. As a whole, the "Smart Tutor System" seeks to optimize the learning process of medical students, providing them with an advanced technological resource that integrates artificial intelligence with medical content.

Keywords: neural networks, llm, transfer learning, embeddings, tokenization, rotating internship.

LISTADO DE ABREVIATURAS

LLM - Modelo de Lenguaje Grande (Large Language Model)

IA - Inteligencia Artificial

QLoRA - Quantized Low-Rank Adaptation (técnica utilizada para optimizar modelos de lenguaje)

JWT - JSON Web Token

Vue - Framework de JavaScript para la construcción de interfaces de usuario

Vuetify - Framework de componentes de interfaz de usuario basado en Vue.js

Django - Framework de desarrollo web para Python

DRF - Django Rest Framework

API - Interfaz de Programación de Aplicaciones (Application Programming Interface)

CORS - Recursos de Origen Cruzado (Cross-Origin Resource Sharing)

JSON - Notación de Objetos JavaScript (JavaScript Object Notation)

SQuaRE - Software Quality Requirements and Evaluation (Norma de Calidad de Software)

APIView - Vista basada en API de Django Rest Framework

CSP - Política de Seguridad de Contenidos (Content Security Policy)

PEFT - Pipelined Efficient Fine-Tuning (técnica de ajuste fino eficiente de modelos)

BitsAndBytes - Biblioteca para la manipulación eficiente de datos binarios

datasets - Biblioteca para manejar y procesar datasets en machine learning

Transformers - Biblioteca de Hugging Face para modelos de lenguaje

SQuaRE - Software Quality Requirements and Evaluation (Norma de Calidad de Software)

Access Token - Token de acceso utilizado para autenticar a los usuarios en un sistema

CAPÍTULO I

MARCO PRELIMINAR

**INGENIERÍA
DE SISTEMAS**
UNIVERSIDAD PÚBLICA DE EL ALTO



CAPÍTULO I

1. MARCO PRELIMINAR

1.1. INTRODUCCIÓN

Los estudiantes de la Carrera de Medicina de la Universidad Pública de El Alto, tienen ciertas dificultades al realizar el Internado Rotatorio, durante este periodo los estudiantes aplican los conocimientos adquiridos durante sus estudios, pero la cantidad y complejidad de la información requerida para los diferentes campos médicos es considerable, por lo cual ellos no pueden lograr las competencias de manera adecuada.

En ese sentido el presente proyecto desea mejorar el logro de competencias en el Internado Rotatorio con el desarrollo del tutor inteligente capaz de colaborar en el aprendizaje y preparación de los estudiantes de medicina para enfrentarse a las competencias de dicho internado. Entonces, para este propósito se utilizarán redes neuronales artificiales con procesamiento de lenguaje natural (NLP), aprovechando su capacidad para entender, interpretar y generar texto de manera fluida y coherente. Estas redes permiten que el sistema tutor inteligente interactúe de manera efectiva con los estudiantes, brindando respuestas personalizadas. La integración de técnicas avanzadas como QLoRA resulta crucial para el proceso del entrenamiento, ya que no solo optimiza el rendimiento de la red neuronal en términos de precisión, sino que también minimiza el uso de recursos computacionales, lo que permite entrenar modelos grandes sin sacrificar la exactitud.

Se aplicará la metodología MelSE en combinación con otra metodología para desarrollar la red neuronal, para la creación del software educativo. Además, se utilizarán herramientas de inteligencia artificial, como redes neuronales artificiales para la implementación del modelo neuronal. La aplicación web se desarrollará con lenguajes de programación JavaScript y Python se utilizarán librerías y frameworks como Vue.

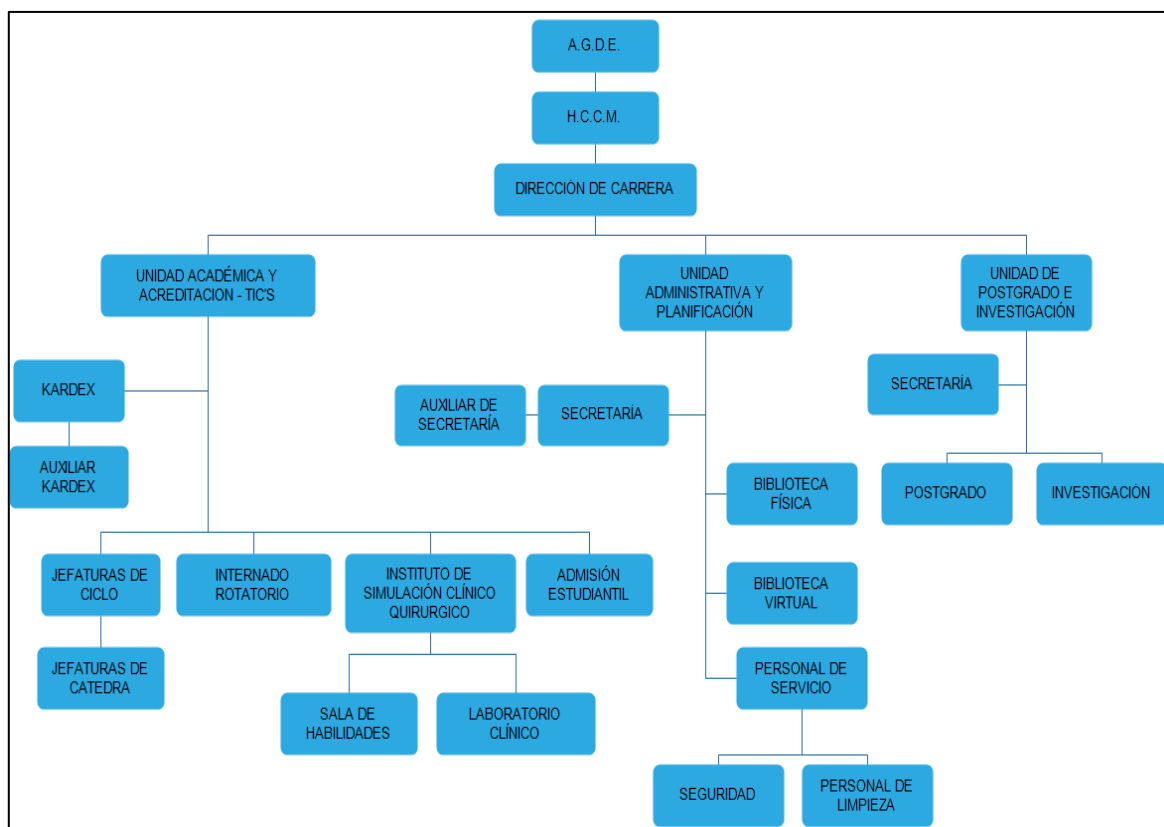
1.2. ANTECEDENTES

1.2.1. *Antecedentes Institucionales*

Indudablemente, la salud en nuestra nación está experimentando cambios significativos que buscan mejorar la calidad de vida de nuestra población en general.

Es así donde la Carrera de Medicina de la Universidad Pública de El Alto tiene como objetivo formar profesionales capaces de promover la investigación y ofrecer soluciones a problemas de salud prioritarios a nivel nacional. Esto implica la participación de investigadores en diversas disciplinas de las Ciencias de la Salud. Además, se busca influir en la formación de nuevos médicos al incorporar y capacitar a un gran número de estudiantes.

- **Visión:** Referente en la formación médica con valores espirituales éticos con solvencia científica, tecnológica de calidad y calidez, reconocida socialmente y acreditada, preservando la excelencia y consolidando su reconocimiento nacional e internacional.
- **Misión:** Formar médicos cirujanos generales e integrales, socialmente comprometidos con valores éticos, para promover la salud, prevenir la enfermedad, desarrollar investigación e interactuar con la familia y comunidad, en el marco de las políticas de salud del país.
- **Objetivos:** Fomentar la realización de investigación y proponer soluciones sobre las problemáticas en salud, que son prioritarias para el país, incluyendo su dimensión Internacional, que requieran la participación de investigadores en diferentes disciplinas de las Ciencias de la salud.

Figura 1*Organigrama Carrera de Medicina*

Nota. Adaptado de *Diseño Curricular Carrera de Medicina*, por Carrera de Medicina - UPEA, 2019.

La modalidad de graduación esta normado por el Plan de Estudios de la Carrera de Medicina aprobado la gestión 2010, refrendado por el Reglamento de modalidades de graduación del Estatuto Orgánico del II congreso Interno de la Universidad Pública De El Alto. La Carrera de Medicina cuenta con un sistema de graduación peculiar en relación a las carreras de la universidad, que consiste en realizar prácticas hospitalarias durante una gestión además de realizar el Servicio Social de Salud Rural Obligatorio.

El Internado Rotatorio es la modalidad que lleva al estudiante de Medicina a concluir la carrera para posteriormente llegar a su titulación y consiste en los siguientes roles:

Tabla 1*Áreas del Internado Rotatorio*

Sigla	Internado Rotatorio	Meses por rote	Total, horas
INT-1	Medicina Interna	2 meses	500
INT-2	Cirugía	2 meses	500
INT-3	Pediatría	2 meses	500
INT-4	Ginecología Obstetricia	3 meses	750
INT-5	Salud Publica -S.S.R.O.	3 meses	750
Total		12 meses	3000

Nota. Adaptado de *Diseño Curricular Carrera de Medicina*, por Carrera de Medicina UPEA, 2019.

1.2.2. Antecedentes Afines al Proyecto de Grado

A continuación, se describen los antecedentes afines al proyecto.

1.2.2.1. Antecedentes Internacionales

(López Guillén, 2020), “Diseño e Implementación de una Funcionalidad de Transfer Learning en un Modelo de Aprendizaje por Refuerzo Aplicado a la Navegación con Drones”, Este trabajo busca diseñar, desarrollar e implementar una funcionalidad de Transfer Learning en un modelo de aprendizaje por refuerzo. Parte de un entorno simulado donde se usan técnicas de aprendizaje por refuerzo para la navegación de drones. El objetivo es mejorar el sistema, permitiendo transferir el aprendizaje adquirido a otros entornos. El trabajo mencionado fue desarrollado en la Universidad Politécnica de Madrid.

(Sanz Torres, 2023), “Detección de Contenido Sexual Mediante Aprendizaje Profundo y Aprendizaje por Transferencia”, Este Trabajo Fin de Grado fue desarrollado en la Universidad Complutense de Madrid se enfoca en desarrollar una herramienta para identificar de forma automática el contenido sexual en imágenes, dado el gran volumen de este tipo de contenido en internet. Se investigaron herramientas previas para detección de contenido

sexual y se utilizarán técnicas de aprendizaje profundo y aprendizaje por transferencia. Además, se creará y limpiará un conjunto de datos con imágenes suficientes para entrenar los modelos que cumplirán con este propósito.

1.2.2.2. Antecedentes Nacionales

(Mendoza Apaza, 2020), “Tutor Inteligente Móvil Para la Enseñanza de la Estática y Dinámica en Estudiantes de Quinto de Secundaria”, este proyecto se llevó a cabo en la Universidad Pública de El Alto y tuvo como objetivo la creación de un Tutor Inteligente Móvil diseñado para enseñar Estática y Dinámica a estudiantes de quinto año de secundaria en la Unidad Educativa "Dionicio Morales". El propósito es mejorar la comprensión de los conocimientos de manera eficiente y dinámica en un tiempo mínimo.

(Flores Mamani, 2022), “Tutor Inteligente Para el Apoyo del Aprendizaje en la Lectura De 1° de Primaria Comunitaria Vocacional de la Unidad Educativa Técnico Humanístico Don Bosco”, el proyecto citado fue desarrollado en la Universidad Mayor de San Andrés, el cual trata sobre el desarrollo de un Sistema Tutor Inteligente para apoyar la lectura de niños de 1° de primaria en la unidad educativa Don Bosco. El sistema complementa el proceso enseñanza-aprendizaje entre estudiantes y profesores, ayudando a los niños a seguir el ritmo de las clases. Para su creación, se utilizó la Metodología de Ingeniería de Software Educativo (MEISE) de Antonieta Abud, combinada con las fases de Mobile-D.

1.3. PLANTEAMIENTO DEL PROBLEMA

1.3.1. Problema Principal

Actualmente, como modalidad de graduación de la carrera de Medicina en la Universidad Pública de El Alto se da a través del Internado Rotatorio. Es importante recalcar que esta etapa para los estudiantes es de suma importancia puesto que ellos aplican todos los conocimientos adquiridos a lo largo de los 5 años. Sin embargo, la cantidad de conocimientos requeridos para esta área suele ser muy ampulosa, en ese sentido muchos de

los estudiantes tienden a fallar en los diversos roteos como ser: Pediatría, Cirugía, Medicina Interna Ginecología Obstétrica y Salud Pública.

Este hecho genera la necesidad imperativa de crear una metodología que permita a los estudiantes repasar las diversas temáticas y que lo mismos puedan tener un entrenamiento continuo en estas áreas médicas.

1.3.2. Problemas Secundarios

- Los estudiantes no pueden prepararse de forma adecuada antes de realizar el Internado Rotatorio, porque ellos no tienen material a su disposición o no tienen material de calidad, como consecuencia al momento de estar realizando el internado en un determinado roteo el interno tiene ciertos inconvenientes en cuanto a conocimientos de algún tema bastante complejo.
- Los estudiantes de medicina a menudo se enfrentan a grandes cantidades de información médica desorganizada, este desorden dificulta la consulta eficiente de recursos y la comprensión de los temas, especialmente cuando se trata de contenido clínico especializado en áreas como Cirugía, Pediatría o Medicina Interna, además que ellos no cuentan con soporte educativo mediante inteligencia artificial.
- Los estudiantes a menudo tienen dificultades para encontrar rápidamente información relevante sobre un determinado tema del Internado Rotatorio, lo que genera un estudio menos eficiente.
- Para los estudiantes es difícil realizar la autoevaluación de los temas estudiados, por lo que no les permite identificar las áreas que deben mejorar.
- Los estudiantes a menudo enfrentan la dificultad de encontrar respuestas rápidas y precisas a preguntas específicas durante su proceso de estudio, porque en muchas ocasiones los docentes solo pueden guiar al estudiante en horario de clases, por lo

cual el estudiante debe acudir a otras fuentes donde en su mayoría no se encuentran algunos conceptos clave.

1.3.3. *Formulación del Problema*

El problema principal en el Internado Rotatorio de la carrera de Medicina en la Universidad Pública de El Alto radica en la necesidad repasar y entrenar los diversos conceptos adquiridos a lo largo de los 5 años de conocimientos adquiridos. Por lo tanto, surge la siguiente pregunta: **¿Cómo mejorar el logro de competencias en el Internado Rotatorio?**

1.4. OBJETIVOS

1.4.1. *Objetivo General*

Desarrollar un Tutor Inteligente Basado en Redes Neuronales Artificiales para mejorar el logro Competencias en el Internado Rotatorio, en la carrera de Medicina de la Universidad Pública de El Alto.

1.4.2. *Objetivos Específicos*

- Seleccionar textos y libros de los temas relevantes del Internado Rotatorio según sus áreas: Cirugía, Pediatría, Medicina Interna, Ginecología Obstetricia y Salud Publica en colaboración con profesionales de medicina para crear una base de datos con material a disposición para el estudiante.
- Eliminar el ruido de los textos seleccionados para realizar el entrenamiento de la red neuronal.
- Visualizar distintos temas de manera organizada para facilitar al estudiante el acceso rápido, facilitando así la retención y comprensión de conceptos de medicina.
- Evaluar al estudiante mediante un cuestionario al finalizar la lectura o estudio de un determinado tema para fortalecer los conocimientos del mismo.

- Disponer de un tutor mediante un modelo de procesamiento de lenguaje natural para interactuar con el estudiante.

1.5. JUSTIFICACIÓN

1.5.1. *Justificación Técnica*

El tutor inteligente basado en redes neuronales tiene capacidades de procesamiento de información y aprendizaje automático.

El tutor inteligente será una aplicación web donde este trabajará mediante consumo de microservicios, así como también el mismo, estará desarrollado con frameworks para aplicaciones web y librerías para entrenamiento de una red neuronal en Python, el consumo de microservicios se estará ejecutando en un servidor de la institución, así como también el consumo de la misma aplicación.

1.5.2. *Justificación Económica*

El presente proyecto es de carácter social por lo cual no tiene una justificación económica.

1.5.3. *Justificación Social*

La decisión de desarrollar un tutor virtual inteligente responde a la necesidad de mejorar la formación de los estudiantes de medicina, así también como consecuencia mejoraríamos los estándares de atención medica en la sociedad. Esta contribución es muy relevante ya que los estudiantes podrán lograr el Internado Rotatorio de manera satisfactoria, como consecuencia tendremos profesionales de la salud más competentes.

Así como también estaríamos ampliando la cantidad de estudiantes que logren la competencia del Internado Rotatorio con la ayuda de la aplicación, asimismo a nivel carrera y a nivel universidad tendríamos un cierto nivel ideal de formación académica.

Ayudar a todos los estudiantes de último semestre de la carrera de medicina de la Universidad Pública de El Alto, donde ellos podrán mejorar sus conocimientos mediante la interactividad con la aplicación web.

1.6. METODOLOGÍAS

El presente proyecto combinará dos metodologías de desarrollo, la metodología CRISP-DM y la metodología MeISE, para aprovechar sus fortalezas complementarias. La metodología CRISP-DM es ideal para entrenar modelos de aprendizaje profundo. Por otro lado, la metodología MeISE proporciona un enfoque educativo y estructurado. Al integrar ambas metodologías, se busca crear un tutor inteligente efectivo.

1.6.1. Metodología CRISP-DM

La metodología CRISP-DM se aplicará en el entrenamiento del modelo LLM para el "Sistema Tutor Inteligente". Se comenzará por comprender las necesidades educativas de los estudiantes en su internado, lo que guiará la recolección y análisis de datos relevantes, como PDFs y textos sobre los temas del internado. El proceso de modelado incluirá el entrenamiento del LLM utilizando técnicas como QLoRA, optimizando sus capacidades de generación de texto. Finalmente, el modelo se integrará en el sistema, asegurando que cumpla con los objetivos educativos establecidos.

1.6.2. Metodología MeISE

Para el desarrollo del software también se aplicará la metodología MeISE, el cual permite un enfoque educativo y estructurado en el desarrollo del sistema desde la conceptualización hasta el despliegue del sistema, asegurando que se cubran todos los aspectos del desarrollo del software.

1.7. PRUEBAS DE SOFTWARE

El presente proyecto opta por realizar las pruebas de funcionalidad, unitarias e integración.

1.7.1. Pruebas de Funcionalidad

Se realizarán pruebas de funcionalidad para asegurar que el sistema cumpla con los requerimientos, especialmente aquellos relacionados con el procesamiento de información médica, funcionen de manera óptima.

1.7.2. Técnica de Caja Negra

Se aplicará principalmente a las pruebas de funcionalidad, evaluando el sistema desde la perspectiva del usuario final, es decir, los estudiantes de medicina.

1.7.3. Técnica de Caja Blanca

Esta técnica se utilizará para las pruebas unitarias e integración, asegurando el correcto funcionamiento para visualizar contenido, proceso de registro de estudiantes y otros.

1.8. HERRAMIENTAS

Se utilizarán las siguientes herramientas para desarrollar el tutor inteligente basado en redes neuronales artificiales:

1.8.1. Para Desarrollar el Sistema Web

- Django: Se utilizará Django para el backend, manejando la lógica de negocio, el control de acceso a los datos médicos y la comunicación con la interfaz del sistema tutor.
- Vue: Vue se utilizará para desarrollar la interfaz de usuario, proporcionando una experiencia interactiva y ágil para los estudiantes al presentar los temas a estudiar propuesto por el tutor inteligente y recibir retroalimentación del mismo.

1.8.2. Para Desarrollar el Tutor Inteligente

Pytorch e empleará para entrenar el modelo que impulsarán el tutor inteligente. Este modelo será entrenado con textos médicos recopilados y se usarán para generar las respuestas del tutor, simulando el razonamiento de un docente del área de medicina.

1.9. LIMITES Y ALCANCES

1.9.1. Limites

- En áreas rurales donde no se cuente con conexiones a internet, existirá dificultades podrá acceder a la aplicación web.

- En el Internado Rotatorio en sus diferentes áreas existen muchos temas y son bastantes extensos, por lo cual la aplicación solo contendrá los temas más impactantes y/o importantes.

1.9.2. Alcances

- Los estudiantes podrán acceder a la aplicación web repasando los temas más importantes y/o relevantes de cada área del Internado Rotatorio.
- La aplicación tendrá información de los temas más relevantes de todas las áreas del Internado Rotatorio.
- El sistema web contendrá contenidos de estudio.
- El tutor inteligente podrá interactuar con el estudiante.
- Se visualizará en la aplicación contenido adaptativo para el estudiante.
- Los estudiantes serán evaluados por el tutor inteligente.

1.10. APORTES

El desarrollo del tutor inteligente será un aporte muy importante para los estudiantes de la carrera de medicina de la Universidad Pública de El Alto, este contribuirá de manera significativa el logro de competencias del Internado Rotatorio. El estudiante tendrá acceso a la aplicación web, donde podrá interactuar con el tutor inteligente, así mediante contenidos de estudio podrá fortalecer sus destrezas de esta manera los estudiantes de medicina tendrán una comprensión y repaso de los distritos temas del Internado Rotatorio, así los estudiantes podrán lograr el Internado Rotatorio de manera satisfactoria, entonces como consecuencia tendremos profesionales de la salud más competentes, teniendo un impacto positivo para la universidad y sociedad.

CAPÍTULO II

MARCO TEÓRICO

**INGENIERÍA
DE SISTEMAS**
UNIVERSIDAD PÚBLICA DE EL ALTO



CAPÍTULO II

2. MARCO TEÓRICO

2.1. INTRODUCCIÓN

En el presente capítulo de este proyecto, se abarcará la base teórica, para la creación del tutor inteligente. Este apartado explorara los conceptos relevantes en el campo de la inteligencia artificial, sistemas web, frameworks y entre otras bases teóricas, el cual nos ayudara a crear el sistema.

2.2. INGENIERÍA DE SOFTWARE

La ingeniería de software es una rama de la ingeniería que está enfocado a la creación de software, desde las primeras fases de definición del sistema hasta su mantenimiento una vez en funcionamiento. En esta definición destacan dos conceptos clave:

- Rama de la ingeniería: Los ingenieros se encargan de hacer que las cosas funcionen. Aplican teorías, métodos y herramientas cuando es apropiado.
- Todos los aspectos de la producción de software: La ingeniería de software no solo se enfoca en los aspectos técnicos del desarrollo, sino que también incluye actividades como la gestión de proyectos y el desarrollo de herramientas (Sommerville, 2011).
- Redes neuronales artificiales: modelos que simulan el comportamiento del cerebro humano (Pressman, 2010).

2.3. INGENIERÍA DE REQUERIMIENTOS

El diseño de software es un proceso creativo y atractivo, pero presenta retos significativos. La ingeniería de requerimientos abarca todas las actividades necesarias para definir las necesidades del proyecto, por lo cual es una disciplina fundamental dentro de la ingeniería de software y se extiende desde la fase de comunicación hasta el modelado, el cual actúa como un puente entre la definición de lo que el cliente necesita y el producto. Muchos

de estos puentes se originan en los participantes del proyecto (gerentes, clientes, usuarios) al definir las necesidades del negocio, las funciones, las características y las restricciones.

La ingeniería de requerimientos permite entender lo que desea el cliente, analizar las necesidades, evaluar la factibilidad, evaluar soluciones, especificar los requerimientos de forma clara y sin ambigüedades, validar la especificación y finalmente, estos mismos requerimientos nos ayudara a tener un sistema funcional (Pressman, 2010).

2.4. SISTEMA TUTOR INTELIGENTE (STI)

Los Sistemas Tutores Inteligentes (STI) surgieron en la década de los ochenta, con el objetivo de impartir conocimiento empleando la inteligencia artificial, ofreciendo asistencia y orientación al estudiante, similar a como lo haría un tutor humano. Por lo tanto, un Sistema Tutor Inteligente es un software que emplea técnicas de inteligencia artificial, con el fin de representar el conocimiento humano.

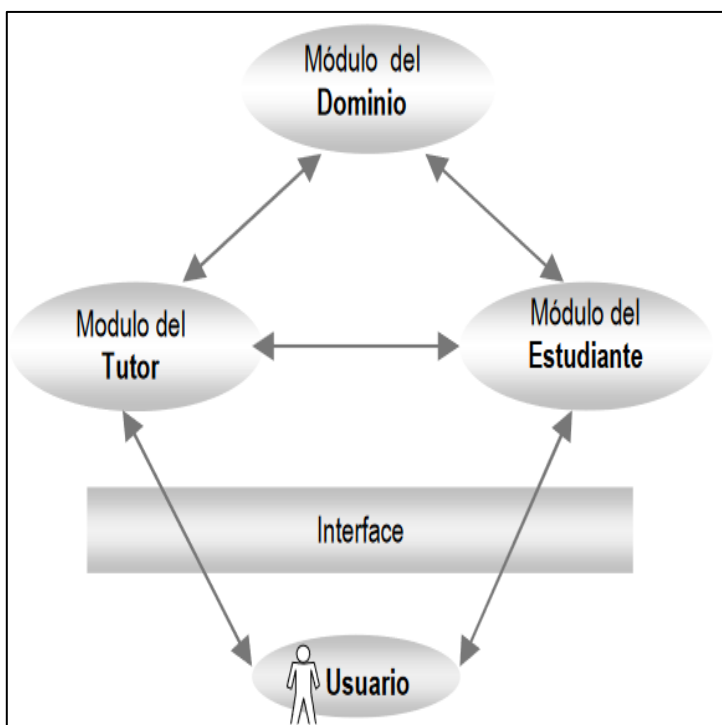
Se ha notado que muchos STI no alcanzan el nivel de “Inteligencia” esperado, debido a las dificultades que existe para simular con precisión el funcionamiento de la mente humana, entonces según investigadores, un STI es una herramienta para apoyar al tutor humano (Cataldi y Lage, 2009).

2.5. ARQUITECTURA DE UN STI

Un Sistema Tutor Inteligente, emula las funciones de un tutor humano, mediante cuatro módulos principales (Cataldi y Lage, 2009).

Figura 2

Arquitectura de un Sistema Tutor Inteligente.



Nota. Adaptado de “Sistemas Tutores Inteligentes Orientados a la Enseñanza Para La Comprensión” (N° 28), por Cataldi y Lage, 2009, Revista Electrónica de Tecnología Educativa (Eduotec).

2.5.1. Módulo Tutor

El módulo tutor de un STI es responsable de aplicar y definir estrategias pedagógicas de enseñanza. El módulo contiene los objetivos de aprendizaje que se deben llegar y los planes necesarios para lograrlos. Monitorea el desempeño del estudiante, provee asistencia cuando sea necesario o cuando el estudiante lo necesite. El módulo tutor integra el conocimiento acerca del método de enseñanza, las técnicas didácticas y dominio que se está enseñanza (Cataldi y Lage, 2009).

2.5.2. Módulo Dominio

Este módulo del STI tiene como objetivo principal almacenar todos los conocimientos, como ser independientes y dependientes del área de la aplicación del STI. Entre los submódulos se encuentran los siguientes:

- **Parámetros Básicos del Sistema:** Almacenada en una base de datos.
- **Conocimientos:** Aquí se almacenan los contenidos que el sistema debe gestionar, a través de los conceptos, preguntas, ejercicios y otros.
- **Elementos Didácticos:** Este submódulo se enfoca en el material multimedia necesario como ser imágenes, pdf, sonidos y otros (Cataldi y Lage, 2009).

2.5.3. Módulo Estudiante

El módulo estudiante del STI tiene como objetivo de realizar un diagnóstico cognitivo del alumno para proporcionar una retroalimentación adecuada. Este módulo incluye varios submódulos (Los datos del se almacenan en una base de datos):

- **Estilos de Aprendizaje:** Este submódulo consiste en una base de datos que contiene los estilos de aprendizaje disponibles en el sistema, así como los métodos de selección y las características de cada uno.
- **Estado de Conocimientos:** Este componente contiene un mapa de conocimientos que se obtiene inicialmente del módulo del dominio (Cataldi y Lage, 2009).

2.5.4. Interfaz

Facilita la interacción del estudiante con el STI, optimizando la presentación de los contenidos (Cataldi y Lage, 2009).

2.6. SOFTWARE DE INTELIGENCIA ARTIFICIAL

El software de inteligencia artificial (IA) emplea algoritmos no numéricos para abordar problemas complejos que no se pueden resolver fácilmente mediante métodos convencionales. Las aplicaciones más comunes de este tipo de software son:

- **Robótica:** control y automatización de máquinas inteligentes que interactúan con su entorno.
- **Sistemas expertos:** software que emula la toma de decisiones de un experto humano.

- Reconocimiento de patrones: análisis de imágenes y voz para identificar características específicas.

2.7. INTELIGENCIA ARTIFICIAL

La Inteligencia Artificial (IA) surgió por una pregunta propuesta por Alan Turing “¿Pueden las maquinas pensar?”. Esta pregunta a motivado a investigadores a buscar formas de hacer que las maquinas actúen de forma inteligente (Yang *et al.*, 2020).

La Inteligencia Artificial, es una rama de la ciencia que estudia la inteligencia en elementos artificiales. Desde la vista de ingeniería, busca crear elementos que obtengan un comportamiento inteligente, es decir, la IA tiene como objetivo de construir máquinas y sistemas inteligentes que actúen de forma similar a un humano. La IA es interdisciplinaria, de las cuales podemos mencionar: Neurociencia, Medicina, Psicología, Matemáticas, Física, y entre otras disciplinas (Romero *et al.*, 2007).

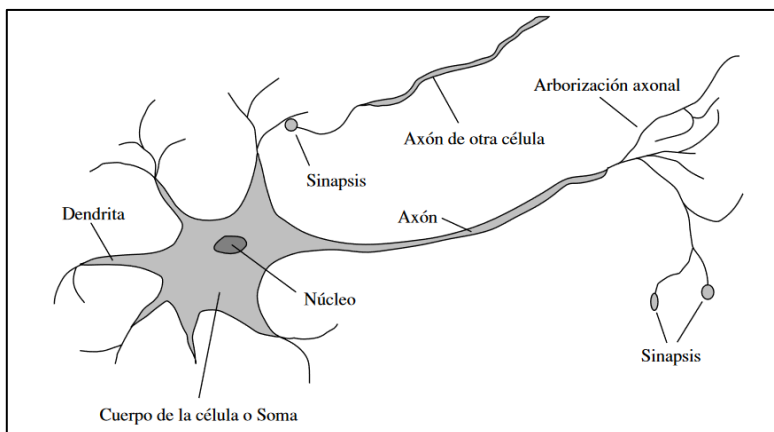
Los esfuerzos de hacer que las maquinas piensen como humanos han avanzado considerablemente, desde alimentar base de conocimientos basadas en reglas, a permitir el aprendizaje automático a partir de datos. Con el tiempo el aprendizaje automático se ha convertido en una fuerza industrial clave (Yang *et al.*, 2020).

2.7.1. Redes Neuronales

Una neurona es una célula encargada de transmitir información a través de señales eléctricas. Se dice que la capacidad del cerebro para procesar información es gracias a las redes, las cuales están compuestas por neuronas. Gracias a esto, los primeros estudios en Inteligencia Artificial se enfocaron en la creación de redes neuronales artificiales, el cual, es también conocido como: conexionismo, procesamiento distribuido paralelo o computación neuronal (Russell y Norvig, 2004).

Figura 3

Neurona Biológica



Nota. Adaptado de *Inteligencia Artificial Un Enfoque Moderno* (p. 13), por Russell y Norvig, 2004.

Una red neuronal artificial (RNA) es un modelo matemático que busca imitar la estructura y funciones de las redes neuronales biológicas. El componente esencial de toda red neuronal artificial es la neurona artificial, que se representa mediante un modelo matemático simple o función. Este modelo sigue tres reglas básicas: multiplicación, suma y activación.

En la entrada de la neurona artificial, los valores de entrada son ponderados, es decir, cada valor se multiplica por un peso específico. En la parte intermedia, se encuentra la función de suma, que acumula todas las entradas ponderadas y añade un sesgo. Finalmente, en la salida, esta suma pasa a través de una función de activación, también conocida como función de transferencia (Suzuki, 2011).

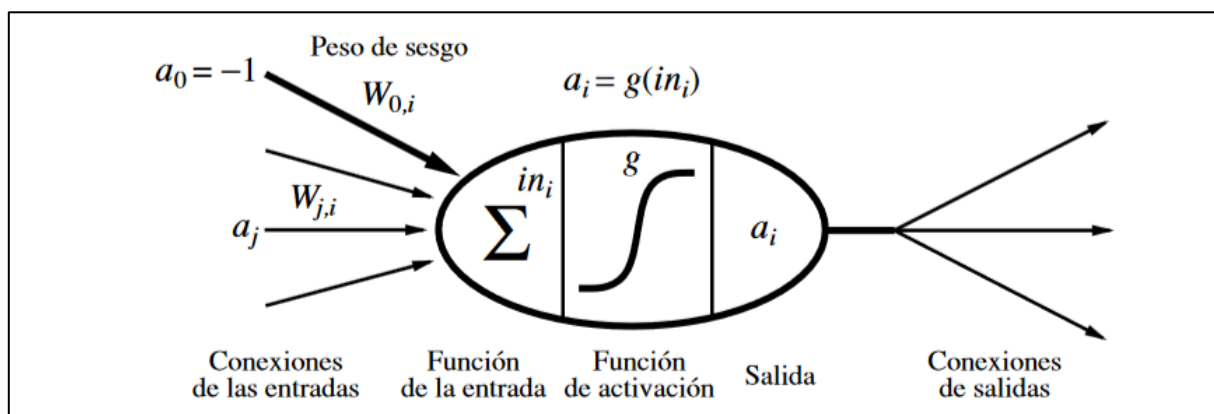
El modelo matemático de una neurona propuesto por McCulloch y Pitts en 1943 indica que la neurona se “activa” cuando una combinación lineal supera un umbral. Desde ese entonces se han desarrollado modelos más realistas, llevándonos a la neurociencia computacional.

Los investigadores de IA, se han interesado en las características abstractas de las redes neuronales, como su capacidad para realizar computación distribuida, tolerar el ruido

en las entradas y aprender de los datos, entonces por ese lado, las redes neuronales artificiales son muy populares y efectivas para construir sistemas modernos (Russell y Norvig, 2004).

Figura 4

Neurona Artificial



Nota. Adaptado de *Inteligencia Artificial Un Enfoque Moderno* (p. 839), por Russell y Norvig, 2004.

2.7.2. Deep Learning

El aprendizaje profundo (Deep Learning, DL) es una rama del aprendizaje automático centrada en el uso de algoritmos inspirados en el funcionamiento del cerebro humano. Las redes neuronales artificiales (RNA) fueron el primer intento de emular los procesos cerebrales en la computación.

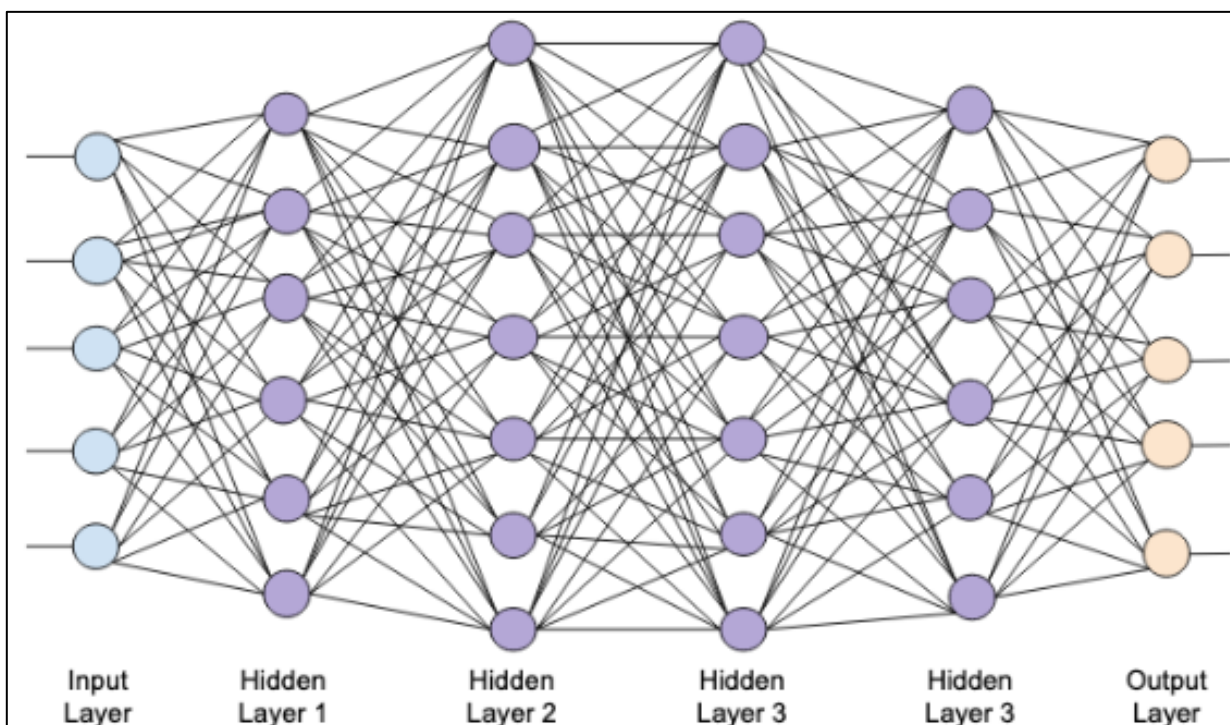
Las redes neuronales son modelos no lineales altamente complejos que pueden aprender de manera automática representaciones jerárquicas de los datos. El aprendizaje profundo utiliza múltiples capas de unidades de procesamiento no lineales (comúnmente conocidas como neuronas), donde cada capa transforma y extrae características de los datos basándose en las salidas de las capas anteriores.

Este enfoque permite que los modelos aprendan de manera jerárquica diferentes niveles de abstracción de los datos. Además, las redes neuronales profundas pueden resolver

problemas tanto supervisados como no supervisados, y se han demostrado ser especialmente prometedoras en el campo del NLP (Sarkar, 2019).

Figura 5

Diagrama de una red neuronal profunda



Nota. A medida que se agrega el número de capas ocultas a la red, su entrenamiento se vuelve más complejo en términos de recursos necesarios y el tiempo que lleva entrenar completamente la red.

Adaptado de *Deep Learning with Keras*, por Point, 2019.

2.7.3. Natural Language Processing

Para entender el Procesamiento del Lenguaje Natural (Natural Language Processing – NLP), lo primero es comprender que define a un lenguaje como “natural”. Entonces desde ese punto, un lenguaje natural es aquel que es desarrollado por el ser humano o generado por el mismo.

NLP es el campo especializado de la informática, la ingeniería y la inteligencia artificial con sus bases en la lingüística computacional. Su objetivo principal es diseñar y desarrollar aplicaciones y sistemas que permitan la interacción entre máquinas y los lenguajes naturales

creados por los seres humanos. Lo cual permite la relación entre persona-computadora (HCI). Las técnicas de NLP permiten a las computadoras procesar el lenguaje humano, lo cual es utilizado para generar resultados útiles (Sarkar, 2019).

Gran parte de las conferencias y publicaciones sobre el procesamiento del lenguaje natural llevan el nombre de “Lingüística Computacional”, y estos términos se usan como sinónimo. Pero existe una diferencia clave en su enfoque. En la lingüística, el lenguaje es el principal objeto de estudio. El cual se puede aplicar métodos computacionales tradicionales, al igual que en áreas científicas, pero solo como herramientas de apoyo. Y por otro lado el NLP se enfoca en analizar y diseñar algoritmos computacionales para procesar el lenguaje humano (Eisenstein, 2018).

2.7.4. Tokenización

La tokenización es el proceso de dividir un texto en unidades más pequeñas, normalmente palabras, subpalabras o caracteres. Estas unidades más pequeñas se denominan tokens y la tokenización es un paso de preprocesamiento fundamental en la mayoría de las tareas de NLP.

Tomemos como ejemplo una oración: “¡Me gusta el procesamiento del lenguaje natural!”. El resultado de la tokenización a nivel de palabra sería el siguiente:

[“Me”, “gusta”, “el”, “procesamiento”, “del”, “lenguaje”, “natural”, “!”]

En este ejemplo, el proceso de tokenización separa la oración en palabras individuales, eliminando la puntuación. Cada palabra se convierte en un token independiente, formando así una lista (vector) de tokens. La tokenización puede realizarse mediante diferentes métodos, y la elección del tokenizador depende de la tarea específica del procesamiento de lenguaje natural (NLP) y de las características del texto (Amaratunga, 2023).

2.7.5. Embeddings

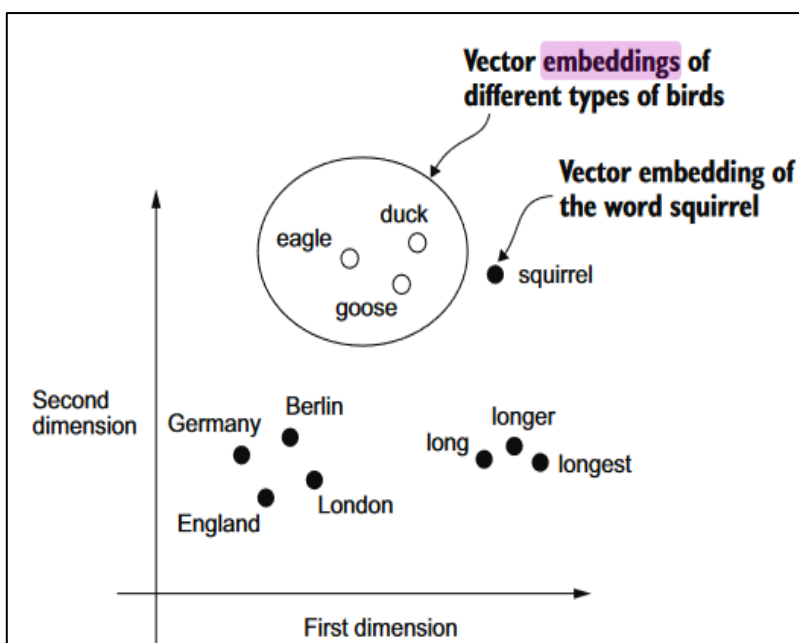
Los embeddings son representaciones vectoriales densas de palabras en un espacio vectorial continuo. En este espacio, las palabras que tienen significados similares se

encuentran más cercanas entre sí. Estas representaciones capturan relaciones semánticas entre palabras, lo que permite a los modelos de NLP comprender el significado de las palabras en función de su contexto (Amaratunga, 2023).

Los word embeddings pueden tener diversas dimensiones, que van desde uno hasta miles. Una mayor dimensionalidad permite capturar relaciones más complejas y matizadas entre las palabras, lo que puede mejorar la capacidad del modelo para comprender contextos más sutiles. Sin embargo, aumentar la dimensionalidad también implica un costo en términos de eficiencia computacional, ya que se requiere más memoria y tiempo de procesamiento (Raschka, 2024).

Figura 6

Word embeddings



Nota. Si los word embeddings son bidimensionales, podemos trazarlas en dos dimensiones. Las palabras que representan conceptos similares tienden a aparecer más cerca unas de otras en este espacio de incrustación. Adaptado de *Build a Large Language Model* (p. 20), por Raschka, 2024.

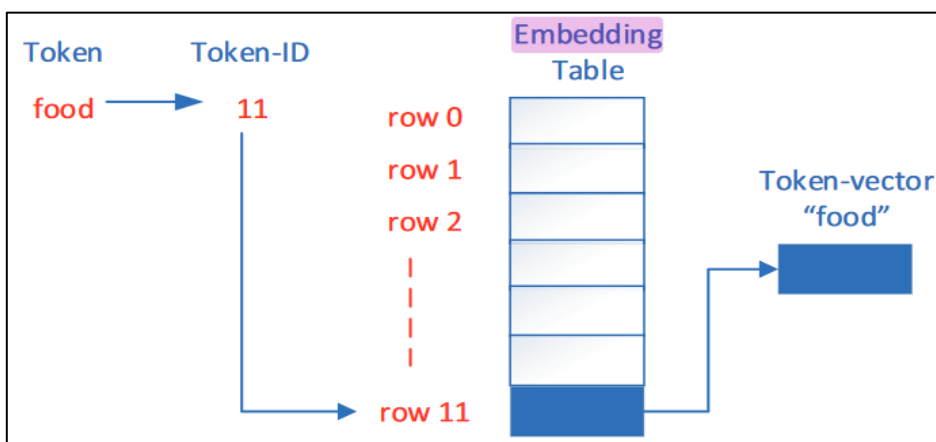
La técnica de word embedding transforma tokens en vectores numéricos. Este proceso se lleva a cabo mediante una tabla de incrustación que mapea cada token a un vector.

- Token: Cada token en el texto se asigna a un identificador numérico único (token-ID) utilizando un tokenizador.
- Token-ID: El ID del token se utiliza como índice para acceder a una fila en la tabla de incrustación.

Este proceso permite representar palabras en un espacio vectorial, lo que facilita la interpretación y manipulación del texto por parte de los algoritmos (Raymond S. T., 2024).

Figura 7

Incrustar palabras (word embeddings) en vectores



Nota. Adaptado de *Natural Language Processing a Textbook with Python Implementation* (p. 364), por Raymond S. T., 2024.

2.7.6. Transformer

El Transformer es un tipo de red neuronal diseñada para procesar secuencias de datos, y su principal característica es el uso del mecanismo de atención en lugar de la recurrencia o convolución que utilizan otras arquitecturas, como las LSTM (Long Short-Term Memory). Una de las diferencias clave entre Transformers y LSTM radica en sus procesos de entrenamiento:

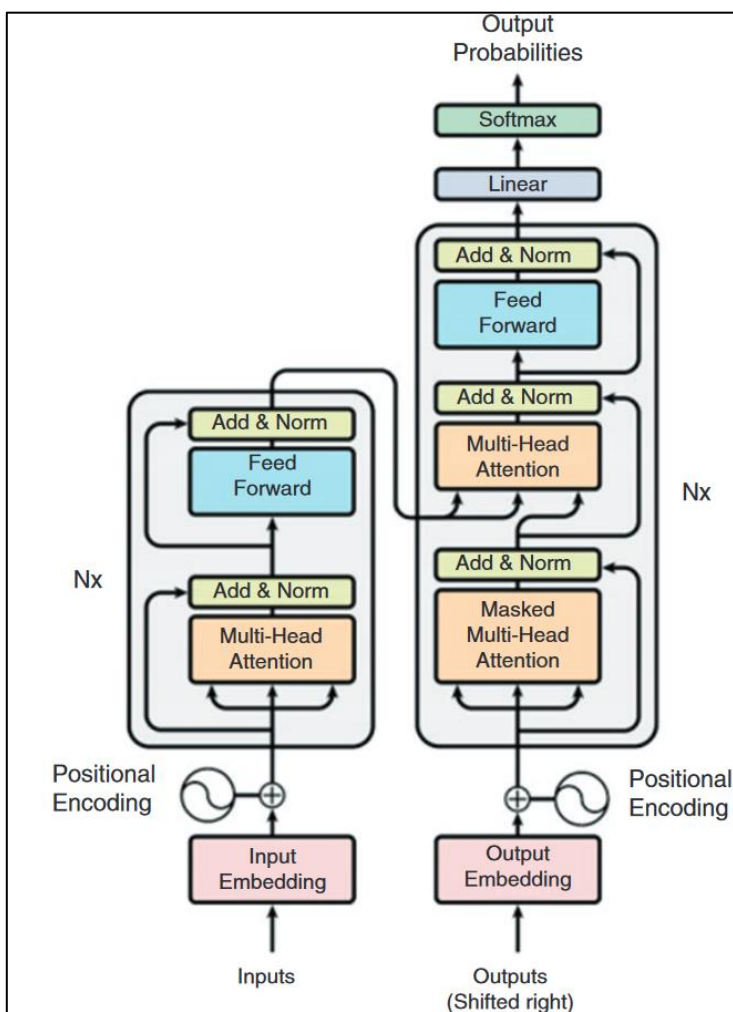
- En las LSTM, el entrenamiento es serial e iterativo, lo que significa que cada palabra de una secuencia se procesa de manera secuencial y la red no puede continuar hasta haber procesado la palabra anterior.

- En cambio, los Transformers permiten un entrenamiento paralelo, donde todas las palabras de la secuencia se procesan simultáneamente, lo que mejora significativamente la eficiencia computacional.

Este enfoque paralelo es posible gracias al mecanismo de autoatención del Transformer, que permite a la red enfocarse en diferentes partes de la secuencia sin necesidad de procesarla en orden (Raymond S. T., 2024).

Figura 8

Arquitectura de un transformer



Nota. Adaptado de *Natural Language Processing a Textbook with Python Implementation* (p. 189), por Raymond S. T., 2024.

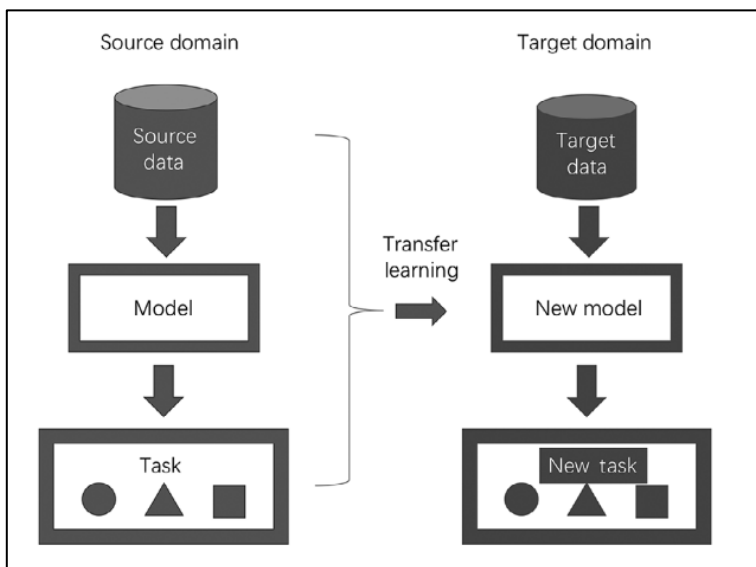
2.7.7. Transfer Learning

El aprendizaje por transferencia (Transfer Learning, por sus siglas en inglés) es un paradigma del aprendizaje automático, donde un algoritmo extrae conocimiento de un dominio para mejorar el rendimiento en una nueva tarea o escenario más específico. Mientras que el aprendizaje automático tradicional, requiere grandes volúmenes de información bien definidos para el entrenamiento, entonces desde ese punto el aprendizaje por transferencia podemos decir que es un nuevo paradigma de aprendizaje.

En la figura a continuación, el proceso en la izquierda representa un enfoque tradicional de aprendizaje automático, mientras que en la derecha muestra un enfoque de aprendizaje por transferencia (Transfer Learning). A diferencia del aprendizaje tradicional, el aprendizaje por transferencia no solo hace el uso los datos del dominio, sino que también aprovecha cualquier información proveniente de los procesos de aprendizaje en el dominio de origen, en otras palabras, toma un dominio con un conjunto de datos y posteriormente incorporamos conocimiento adicional.

Figura 9

Proceso de Transfer Learning



Nota. Adaptado de *Transfer Learning* (p. 8), por Yang et al., 2020.

En Transfer Learning se ha investigado bajo diferentes terminologías en el área de la Inteligencia Artificial, como la reutilización de conocimiento, el aprendizaje basado en casos (RBC), aprendizaje por analogía, la adaptación del dominio, el preentrenamiento y el ajuste fino (fine-tuning).

Para llevar a la practica el aprendizaje automático sería difícil de recopilar grandes cantidades de información para un determinado dominio. A continuación, veremos algunas de las razones para trabajar con aprendizaje por transferencia:

- Muchas aplicaciones solo disponen de pocos datos: Los datos etiquetados de alta calidad suelen ser muy limitados. Los métodos tradicionales de aprendizaje automático a menudo no logran generalizarse adecuadamente en nuevos escenarios, lo que como consecuencia tenemos el fenómeno conocido como sobreajuste.
- Los modelos de aprendizaje automático deben ser robustos: El aprendizaje automático tradicional suele asumir que los datos de entrenamiento y de prueba se obtienen de la misma fuente.
- La privacidad del usuario y la seguridad de los datos son cuestiones importantes: A menudo se debe trabajar con organizaciones aprovechando múltiples datos, así como también estos mismos datos son de carácter privativo.

Entonces de por esas mismas razones y entre otras, se motivaron a los sistemas desarrollar Transfer Learning. Una vez que obtengamos un conjunto de conocimientos bien desarrollado, en este caso un modelo de red neuronal, podemos llevar este mismo modelo para beneficiar a otros dominios más específicos, es decir, el modelo podría ser destinado para otras tareas similares (Yang *et al.*, 2020).

2.7.8. Transfer Learning Basado en Modelos

El aprendizaje por transferencia basado en modelos, lo cual es también conocido como aprendizaje por transferencia basado en parámetros, asume que la tarea de origen y la tarea de destino comparten algún conocimiento en común a nivel del modelo (Yang *et al.*, 2020).

Por ejemplo, un modelo generativo de texto se puede aplicar para una nueva tarea, la cual podría ser un modelo para resumir texto (seguirá siendo un modelo generativo a nivel del modelo).

2.7.9. Inteligencia Artificial Generativa

La inteligencia artificial generativa (Gen AI, por sus siglas en inglés) es un tipo de inteligencia artificial que emplea redes neuronales y algoritmos de aprendizaje profundo para detectar patrones en los datos existentes, a partir de ellos se generan contenidos. Aprendiendo patrones de grandes volúmenes de datos, estos algoritmos sintetizan el conocimiento necesario para generar texto, imágenes, audio, video y otras formas (Baum, 2023).

2.7.10. Language Large Model

Un modelo de lenguaje grande (Language Large Model – LLM) es una red neuronal diseñada para comprender, generar y responder a texto de manera similar a como lo haría una persona. Estos modelos son redes neuronales profundas que se entrenan con enormes cantidades de datos textuales, a menudo cubriendo una gran parte del contenido disponible públicamente en Internet.

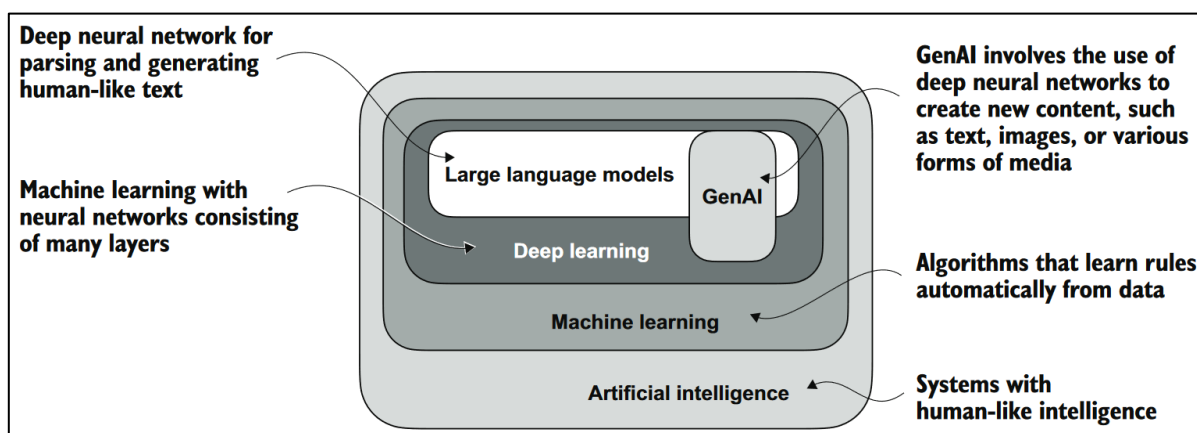
El término "large" en "large language model" hace referencia tanto al tamaño del modelo en términos de la cantidad de parámetros como a la gran cantidad de datos utilizados para entrenarlo. Estos modelos suelen contar con decenas o incluso cientos de miles de millones de parámetros, que son los pesos ajustables dentro de la red que se optimizan durante el proceso de entrenamiento para predecir la siguiente palabra en una secuencia. La tarea de predecir la siguiente palabra es relevante porque se aprovecha de la naturaleza

secuencial del lenguaje, lo que permite que los modelos aprendan el contexto, la estructura y las relaciones dentro del texto.

Los LLM utilizan una arquitectura llamada transformers, que les permite prestar atención de manera selectiva a diferentes partes de la entrada al hacer predicciones. Esta capacidad les otorga una especial habilidad para manejar los matices y complejidades del lenguaje humano. Debido a que los LLM pueden generar texto, se los considera una forma de inteligencia artificial generativa, comúnmente abreviada como IA generativa o GenAI (Raschka, 2024).

Figura 10

Inteligencia Artificial



Nota. Descripción jerárquica de la relación entre los diferentes campos, los LLM representan una aplicación específica de aprendizaje profundo, capaz de generar y procesar textos. El aprendizaje profundo es una rama del aprendizaje automático, el cual usa redes neuronales multicapa. Adaptado de *Build a Large Language Model* (p.3), por Raschka, 2024.

2.7.11. Aplicaciones de los LLM

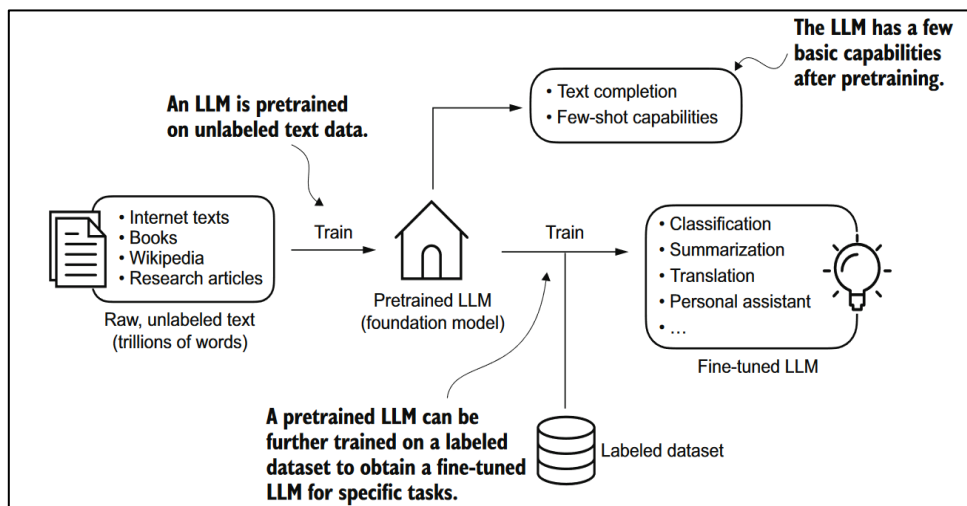
Debido a sus capacidades avanzadas para analizar y comprender datos de texto no estructurados, los modelos de lenguaje grande (LLM) tienen una amplia gama de aplicaciones en diversos dominios.

Recientemente, estos modelos se han utilizado para la creación de contenido, incluyendo la escritura de ficción, artículos e incluso código informático. Además, los LLM impulsan chatbots y asistentes virtuales sofisticados, como ChatGPT de OpenAI o Gemini de Google (anteriormente conocido como Bard), que responden a consultas de los usuarios y mejoran motores de búsqueda tradicionales como Google Search o Microsoft Bing.

El proceso general para desarrollar un modelo de lenguaje grande (LLM) incluye dos etapas principales: preentrenamiento y fine-tuning (ajuste fino). En el preentrenamiento, el modelo se entrena inicialmente utilizando un conjunto de datos extenso y variado, lo que le permite adquirir una comprensión general del lenguaje. Este modelo preentrenado actúa como una base clave que puede ser mejorada posteriormente a través del fine-tuning, un proceso en el que el modelo se entrena con un conjunto de datos más específico y limitado, adaptado a tareas o dominios particulares (Raschka, 2024).

Figura 11

Fine-tuning



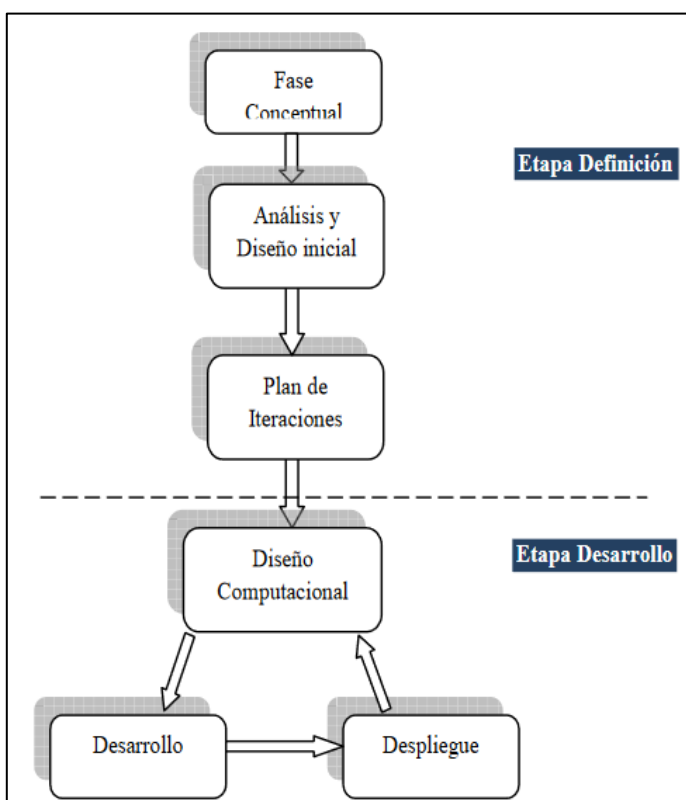
Nota. Pre-entrenamiento de un LLM implica la predicción de la siguiente palabra de acuerdo a grandes cantidades de información. Un LLM previamente capacitado se puede hacer un fine-tuning con datos más específicos. Adaptado de *Build a Large Language Model* (p. 6), por Raschka, 2024.

2.8. METODOLOGÍA DE INGENIERÍA DE SOFTWARE EDUCATIVO

La Metodología de Ingeniería de Software Educativo (MeISE) propone un ciclo de vida que se divide en dos etapas. La primera fase abarca la definición de requisito, el análisis y diseño preliminar, donde se identifican de manera general las características que se desean alcanzar con el producto y la arquitectura sobre la que se construirá el software. Esta etapa concluye con la elaboración de un plan de iteraciones, en el cual se asegura que el producto, al final de cada iteración esté didácticamente completo, es decir, que cubra algunos de los objetivos educativos del software. (Abud, 2009).

Figura 12

Ciclo de vida de la Metodología MeISE



Nota. Adaptado de “MeISE: Metodología de Ingeniería de Software Educativo” (Volumen 2), por Abud, 2009, Revista Internacional de Educación en Ingeniería.

2.8.1. Fase Conceptual

Es la primera fase de la metodología MeISE, comienza con una investigación enfocada en los requerimientos que se deberá cumplir en el producto a desarrollar. Se elabora el plan de proyecto, se evalúan riesgos y se definen los criterios de éxito (Abud, 2009).

Tabla 2

Actividades y Artefactos de la Fase Conceptual

Actividad	Artefacto
Analizar las necesidades educativas	Modelo instruccional (incluye temática a atender, objetivos, conocimientos previos, fuentes de información, modelo educativo a utilizar, elementos de motivación y formas de evaluación). Glosario (descripción de los términos que pueden causar confusión o duda)
Revisar alternativas de solución	Estudio de alternativas (establece las diferentes alternativas que se tienen para el desarrollo del software, se determina el tipo de modelo educativo y se justifica la elección).
Elaborar un estudio de riesgos	Lista de riesgos (establece los riesgos relativos al desarrollo y a los aspectos pedagógicos y la forma de atenderlos).
Conformar del equipo de trabajo y el plan inicial de desarrollo	Plan Inicial (se conforma el equipo de trabajo, se elabora la programación de actividades, se asignan responsables a cada una y se determinan los tiempos estimados para llevarlas a cabo).
Identificar la funcionalidad que se pretende alcanzar con el software	Modelo de actores (identifica los tipos de

	usuario que utilizarán el software y describe sus características).
	Modelo de casos de uso (establece un modelo general de las funciones que cubrirá el sistema a través de diagramas de casos de uso y su especificación).
Establecer los criterios de medición de calidad del proceso, considerando aspectos tanto técnicos como pedagógicos	Modelo de aceptación (incluye las características mínimas que deben cumplirse para que el producto se acepte).

Nota. Adaptado de “MeISE: Metodología de Ingeniería de Software Educativo” (Volumen 2), por Abud, 2009, Revista Internacional de Educación en Ingeniería.

2.8.2. Análisis y Diseño Inicial

En la fase de análisis y diseño inicial, se estudia el dominio del problema y se define la arquitectura del sistema. Durante esta etapa, se describe de manera detallada los requisitos del software, así como también las características educativas y de comunicación que el producto debe incorporar (Abud, 2009).

Tabla 3

Actividades y Artefactos de la Fase de Análisis y Diseño Inicial

Actividad	Artefacto
Identificar los requisitos funcionales y no funcionales que se cubrirán con el software	Modelo de requisitos (Se determinan los requisitos que debe cumplir el software en cuanto a funcionalidad, comunicación, interfaz y docencia).
Establecer la arquitectura del software	Descripción de la arquitectura (establecer la arquitectura base sobre la cual se desarrollará el software; se debe considerar que dicha arquitectura sea capaz de

Actividad	Artefacto
Elaborar el diseño educativo	<p>atender adecuadamente las tareas de aprendizaje que se van a manejar).</p> <p>Modelo educativo (Se definen el objetivo terminal y los subobjetivos, y en base a estos establecen las tareas de aprendizaje apegadas al tipo de modelo educativo).</p>
Elaborar el diseño de comunicación general del producto.	<p>Modelo de interfaz (diseño de las zonas de comunicación y pantallas que se seguirán a lo largo del desarrollo).</p> <p>Modelo de navegación (diseño de los caminos de navegación generales que presentará al usuario).</p> <p>Prototipo de la interfaz de usuario (establecer las plantillas de diseño que se seguirán a lo largo del desarrollo).</p>

Nota. Adaptado de “MeISE: Metodología de Ingeniería de Software Educativo” (Volumen 2), por Abud, 2009, Revista Internacional de Educación en Ingeniería.

2.8.3. Plan de Iteraciones

Después de identificar los requisitos que debe cumplir el software, se analiza cuantos subproductos funcionales pueden generarse para liberar partes operativas del sistema con el fin de llevar un mejor control durante el desarrollo del sistema. Tras identificar los incrementos, se debe priorizarlos, otorgando mayor importancia a aquellos que cubren los conocimientos fundamentales (Abud, 2009).

Tabla 4*Actividades y Artefactos de la Fase del Plan de Iteraciones*

Actividad	Artefacto
Diseñar las iteraciones de forma que las versiones ejecutables cubran objetivos didácticos bien planeados, de acuerdo a la secuencia de temas.	Plan de iteraciones (dividir el desarrollo en iteraciones, cuidando de que cada iteración cubre requisitos y objetivos educativos completos).
Priorizar las iteraciones, de modo que las que contienen conocimientos básicos que se requieren como base para aprendizajes posteriores se ejecuten primero.	Lista de iteraciones priorizadas (ordenar las iteraciones programadas de forma lógica de acuerdo a los contenidos).

Nota. Adaptado de “MeISE: Metodología de Ingeniería de Software Educativo” (Volumen 2), por Abud, 2009, Revista Internacional de Educación en Ingeniería.

2.8.4. Diseño Computacional

Para cada iteración, es importante elaborar un diseño computacional detallado que sirva como base para el desarrollo del sistema (Abud, 2009).

Tabla 5*Actividades y Artefactos de la Fase de Diseño Computacional*

Actividad	Artefacto
Realizar el plan de trabajo de la iteración	Plan de trabajo (se determina las tareas que se realizarán en el diseño de la aplicación se asignan a los miembros del equipo y se calendarizan).
Elaborar el diseño computacional	Modelo de diseño (detallar el diseño a través de diagramas de clases y secuencia, incluir la descripción de clases y métodos; para los desarrollos que requieren bases de datos, incluir

Actividad	Artefacto
	la especificación de diccionario de datos y diagramas entidad-relación).
Refinar el diseño de navegación	Modelo de navegación refinado (diseñar los caminos de navegación específicos para la iteración en desarrollo).
Refinar prototipo de interfaz	Modelo de interfaz usuario (desarrollar las pantallas específicas para los elementos de la iteración en desarrollo).

Nota. Adaptado de “MeISE: Metodología de Ingeniería de Software Educativo” (Volumen 2), por Abud, 2009, Revista Internacional de Educación en Ingeniería.

2.8.5. Desarrollo

En esta fase, se procede con el desarrollo del producto, implementando la arquitectura definida de manera que podamos obtener una versión funcional del software. Esta versión del software debe estar lista para ser utilizada por los usuarios finales (Abud, 2009).

Tabla 6

Actividades y Artefactos de la Fase de Desarrollo

Actividad	Artefacto
Desarrollar los componentes	Modelo de desarrollo (Determinar los componentes a desarrollar y documentarlos).
Probar los componentes	Modelo de pruebas unitarias (Realizar pruebas de los componentes contra los criterios previamente establecidos. Estas pruebas deben incluir las pruebas del diseño instruccional).
Integrar al desarrollo previo	Modelo de Integración (establecer un plan para incorporar el nuevo desarrollo a la liberación previa si es el caso).

Actividad	Artefacto
Realizar pruebas de integración	Pruebas de integración (realizar pruebas para verificar que la incorporación del nuevo incremento no ha inducido fallas al sistema).

Nota. Adaptado de “MeISE: Metodología de Ingeniería de Software Educativo” (Volumen 2), por Abud, 2009, Revista Internacional de Educación en Ingeniería.

2.8.6. Despliegue

En la fase de despliegue, se lleva a cabo la transición del producto desarrollado hacia los usuarios finales. Finalizando esta fase se debe analizar la conveniencia de continuar los desarrollos del producto, si es el caso, se debe regresar a la etapa de diseño computacional, y así continuar con el siguiente incremento (Abud, 2009).

Tabla 7

Actividades y Artefactos de la Fase de Despliegue

Actividad	Artefacto
Entregar producto al usuario	Producto (Se debe entregar el producto debidamente empacado, etiquetado y con información sobre su contenido, aplicación, población objetivo y requerimientos de instalación). Manual de Usuario (Debe contener información detallada de cómo utilizar el software). Manual de Instalación (información de los requerimientos para su funcionamiento y procedimiento de instalación).
Evaluar las características de calidad y satisfacción de los usuarios	Aceptación del Usuario (realizar pruebas con los usuarios finales y comprobar su grado de satisfacción y efectividad del software).

Actividad	Artefacto
Evaluar la conveniencia de continuar con otro incremento al producto	Evaluación de despliegue (analizar los resultados de la prueba de aceptación del usuario y determinar si es conveniente seguir con otra iteración).

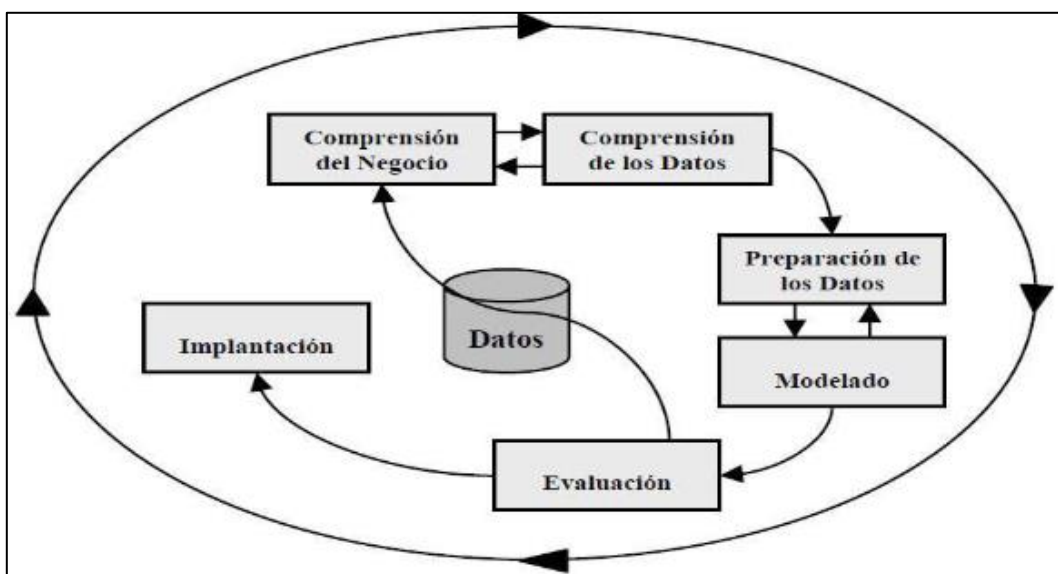
Nota. Adaptado de “MeISE: Metodología de Ingeniería de Software Educativo” (Volumen 2), por Abud, 2009, Revista Internacional de Educación en Ingeniería.

2.9. METODOLOGÍA CRISP-DM

Existen varios modelos de proceso diseñados para el desarrollo de proyectos de minería de datos, entre los cuales se destacan SEMMA (Muestrear, Explorar, Modificar, Modelar, Evaluar), DMAMC (Definir, Medir, Analizar, Mejorar, Controlar) y CRISP-DM (Proceso Estándar de la Industria para la Minería de Datos). Sin embargo, entre ellos, el modelo CRISP-DM se destaca como uno de los más utilizados tanto en entornos académicos como industriales (Wirth y Hipp, 2000).

Figura 13

Modelo de procesos CRISP-DM



Nota. Metodología CRISP-DM (Cross Industry Standard Process For Data Mining). Adaptado de “Metodología para el Desarrollo de Proyectos de Minería de Datos CRISP-DM”, por Wirth y Hipp, 2000.

2.9.1. Fase de Comprensión del Negocio o Problema

En esta fase, resulta fundamental la capacidad de traducir el conocimiento adquirido sobre el negocio y sus problemas relacionados con la minería de datos en un plan preliminar que tenga como objetivo alcanzar las metas definidas por la empresa. Las tareas principales de esta fase son:

- **Determinación de los objetivos del negocio:** Esta primera tarea busca identificar el problema específico que se quiere resolver, establecer por qué es necesario aplicar la minería de datos y definir criterios de éxito. Los problemas pueden ser variados, tales como la detección de fraudes en el uso de tarjetas de crédito, la identificación de intentos de acceso no autorizado a un sistema, o el aseguramiento del éxito de una campaña publicitaria. Con respecto a los criterios de éxito, estos pueden ser cualitativos, donde un experto evalúa los resultados del proceso de minería de datos, o cuantitativos.
- **Evaluación de la situación:** En esta tarea se analiza el estado actual antes de iniciar el proceso de minería de datos, tomando en cuenta factores como el conocimiento previo sobre el problema, la disponibilidad de datos suficientes para abordarlo y la relación costo-beneficio de aplicar minería de datos. En esta etapa se establecen los requisitos del problema.
- **Determinación de los objetivos de DM:** Esta tarea busca traducir los objetivos del negocio en metas concretas para el proyecto de minería de datos. Por ejemplo, si el objetivo comercial es lanzar una campaña publicitaria para aumentar la asignación de créditos hipotecarios, el objetivo de la minería de datos podría ser identificar el perfil de clientes según su capacidad de endeudamiento.

- Producción de un plan del proyecto: Como tarea final de la primera fase de CRISP-DM, esta actividad consiste en crear un plan de proyecto que detalle los pasos a seguir y las técnicas específicas que se emplearán en cada uno (Wirth y Hipp, 2000).

2.9.2. Fase de Comprensión de Los Datos

La segunda fase del proceso, conocida como fase de comprensión de los datos, implica una primera recopilación de datos con el propósito de establecer un contacto inicial con el problema, familiarizarse con los datos, evaluar su calidad y detectar relaciones evidentes que permitan formular las primeras hipótesis. Esta fase, junto con las dos siguientes, suele requerir la mayor inversión de tiempo y esfuerzo en un proyecto de minería de datos, ya que es de vital importancia recopilar todos los datos necesarios.

Las tareas principales de esta fase del proceso incluyen:

- Recolección de datos iniciales: La primera tarea en esta fase de CRISP-DM consiste en recolectar los datos iniciales y adecuarlos para su procesamiento posterior.
- Descripción de los datos: Tras adquirir los datos iniciales, estos deben describirse en términos de volumen (número de registros y campos por registro), identificación, el significado de cada campo y el formato inicial.
- Exploración de datos: Seguidamente, los datos son explorados para descubrir su estructura general. Esta tarea implica aplicar pruebas estadísticas básicas para identificar propiedades en los datos, además de generar tablas de frecuencia y gráficos de distribución. El resultado es un informe de exploración de los datos.
- Verificación de la calidad de los datos: Por último, se realizan verificaciones para evaluar la consistencia de los valores de cada campo, la cantidad y distribución de valores nulos, y la identificación de valores fuera de rango, los cuales podrían ser considerados como ruido. El propósito es asegurar la integridad y precisión de los datos (Wirth y Hipp, 2000).

2.9.3. Fase de Preparación de los Datos

Una vez completada la recolección inicial de datos, se procede a su preparación para adaptarlos a las técnicas de minería de datos que se utilizarán en fases posteriores, como las técnicas de visualización de datos, la búsqueda de relaciones entre variables y otras medidas exploratorias. La preparación de los datos abarca tareas generales como la selección de datos relevantes para las técnicas de modelado, limpieza de los datos, creación de nuevas variables, integración de diferentes fuentes de datos y cambio de formatos.

Esta fase está estrechamente relacionada con la fase de modelado, ya que los datos necesitan ser procesados de manera diferente dependiendo de la técnica de modelado seleccionada. Las fases de preparación y modelado interactúan de manera continua. Las tareas involucradas en esta fase incluyen:

- Selección de datos: En esta etapa, se selecciona un subconjunto de los datos adquiridos en la fase anterior, basándose en criterios previamente establecidos, como la calidad de los datos (completitud y corrección) y las limitaciones de volumen o tipo de datos relacionadas con las técnicas de minería de datos elegidas.
- Limpieza de los datos: Esta tarea, que complementa la selección de datos, es una de las más laboriosas, ya que se requieren diversas técnicas para mejorar la calidad de los datos y prepararlos para la fase de modelado. Algunas de las técnicas aplicadas incluyen la normalización de los datos, discretización de campos numéricos, manejo de valores ausentes, eliminación de ruido y reducción del volumen de datos.
- Estructuración de los datos: En esta tarea, se llevan a cabo operaciones como la generación de nuevos atributos a partir de los existentes, la integración de nuevos registros y la transformación de valores para los atributos existentes, con el fin de optimizar los datos para las siguientes fases del proceso.

- Integración de los datos: Esta tarea implica la creación de nuevas estructuras de datos a partir de los datos seleccionados. Esto puede incluir la generación de nuevos campos a partir de los ya existentes, la creación de nuevos registros o la fusión de tablas, campos o nuevas tablas que resuman las características de múltiples registros o de otros campos.
- Formateo de los datos: El formateo de los datos se refiere a realizar transformaciones sintácticas sobre los datos sin alterar su significado. El propósito de esta tarea es preparar los datos para que sean compatibles con las herramientas de modelado de datos y otras técnicas de minería de datos. Entre las acciones típicas se incluyen la reordenación de campos o registros, y la adaptación de los valores de los campos a las limitaciones de las herramientas, como la eliminación de caracteres especiales, comas, tabuladores, o la normalización de los rangos de valores (Wirth y Hipp, 2000).

2.9.4. Fase de Modelado

En la fase de modelado de CRISP-DM, se seleccionan las técnicas de modelado más adecuadas para el proyecto de minería de datos específico.

En esta fase, además de la selección y construcción del modelo, se toman decisiones clave relacionadas con la evaluación y validación del rendimiento del modelo. La descripción de las tareas principales en esta fase es la siguiente:

- Selección de la técnica de modelado: En esta tarea, se selecciona la técnica de minería de datos más adecuada según el tipo de problema a resolver. El objetivo principal del proyecto debe guiar la elección de la herramienta. Por ejemplo, para problemas de clasificación, se podrían elegir técnicas como árboles de decisión, k-nearest neighbors o razonamiento basado en casos (CBR). Para tareas de predicción, podrían usarse análisis de regresión o redes neuronales, y para segmentación, redes neuronales o técnicas de visualización.

- **Generación del plan de prueba:** Una vez seleccionado el modelo, se debe definir un procedimiento para evaluar su calidad y validez. En tareas supervisadas, como clasificación, la calidad del modelo generalmente se mide a través de la tasa de error. Para esto, se divide el conjunto de datos en dos partes: un conjunto de entrenamiento, que se utiliza para construir el modelo, y un conjunto de prueba, que permite evaluar la precisión del modelo generado.
- **Construcción del modelo:** Después de elegir la técnica, se aplica a los datos previamente preparados para generar uno o más modelos. Cada técnica de modelado tiene parámetros específicos que determinan sus características. El proceso de selección de los mejores parámetros es iterativo, y se realiza ajustando esos parámetros basándose en los resultados obtenidos.
- **Evaluación del modelo:** Esta tarea implica la interpretación de los modelos generados con base en el conocimiento previo del dominio y los criterios de éxito previamente definidos. Los expertos en el área del dominio del problema evalúan los modelos en su contexto, mientras que los expertos en minería de datos aplican sus propios criterios para asegurar que el modelo sea adecuado y efectivo. Esto incluye la validación de la consistencia del conjunto de prueba, la evaluación de la ganancia o pérdida en las tablas, y otros criterios de calidad relevantes para el problema en cuestión (Wirth y Hipp, 2000).

2.9.5. Fase de Evaluación

En la fase de evaluación, se examina el modelo considerando el cumplimiento de los criterios de éxito establecidos para el problema. Es importante tener en cuenta que la fiabilidad calculada del modelo se aplica únicamente a los datos sobre los cuales se realizó el análisis. Es necesario revisar el proceso y, si es necesario, repetir algún paso anterior donde pudiera haberse cometido un error. Además, se pueden utilizar diversas herramientas para interpretar los resultados. Las matrices de confusión, como las de Edelstein (1999), son comúnmente

usadas en problemas de clasificación y consisten en una tabla que muestra cuántas clasificaciones se realizaron para cada tipo. La diagonal de la tabla representa las clasificaciones correctas. Las tareas de esta fase incluyen:

- Evaluación de los resultados: En las fases de evaluación previas se analizaron factores como la precisión y generalidad del modelo. Esta tarea evalúa el modelo en relación con los objetivos del negocio y busca identificar si hay algún aspecto del negocio que pueda hacer que el modelo sea deficiente, o si es recomendable probarlo en un escenario real, si las restricciones de tiempo lo permiten.
- Proceso de revisión: El proceso de revisión implica calificar todo el proceso de minería de datos (DM) para identificar aspectos que podrían mejorarse.
- Determinación de futuras fases: Si se ha comprobado que los resultados hasta este momento son satisfactorios, se puede avanzar a la siguiente fase. En caso contrario, se puede optar por realizar otra iteración desde la fase de preparación de datos o modelado con nuevos parámetros. En algunos casos, podría considerarse iniciar un nuevo proyecto de minería de datos desde cero (Wirth y Hipp, 2000).

2.9.6. Fase de Implementación

En la fase de implementación, una vez que el modelo ha sido construido y validado, el conocimiento obtenido se transforma en acciones dentro del proceso de negocio. Esto puede implicar que el analista recomiende acciones basadas en la observación del modelo y sus resultados, o que el modelo sea aplicado a diferentes conjuntos de datos o como parte de un proceso, como en análisis de riesgo crediticio o detección de fraudes, entre otros. Generalmente, un proyecto de minería de datos no concluye con la implantación del modelo, ya que es necesario documentar y presentar los resultados de manera comprensible para el usuario, con el fin de incrementar el conocimiento. Además, en la fase de explotación, es fundamental garantizar el mantenimiento de la aplicación y la difusión de los resultados. Las tareas de esta fase incluyen:

- Plan de implementación: Para integrar los resultados del proceso de minería de datos (DM) en la organización, esta tarea toma los resultados de la evaluación y establece una estrategia para su implementación.
- Monitorización y Mantenimiento: Si los modelos generados en el proceso de Data Mining son implementados en el dominio del problema como parte de la rutina diaria, es recomendable preparar estrategias de monitorización y mantenimiento para ser aplicadas sobre los modelos (Wirth y Hipp, 2000).

2.10. TECNOLOGÍAS PARA EL DESARROLLO

2.10.1. Django

Los frameworks son herramientas fundamentales en el desarrollo de aplicaciones web, ya que simplifican el trabajo de los arquitectos de software, permitiendo agilizar el proceso de desarrollo mediante la reutilización de código existente. Además, fomentan la adopción de buenas prácticas de desarrollo, como la implementación de patrones de diseño.

Django es un framework web de código abierto desarrollado en Python, diseñado para facilitar la creación de aplicaciones web de manera más rápida y con menor cantidad de código. Originalmente, fue creado para gestionar aplicaciones web de sitios orientados a noticias como World Online, pero posteriormente se lanzó bajo la licencia BSD. Este framework sigue el patrón de arquitectura modelo-vista-controlador y proporciona una serie de componentes para el desarrollo de aplicaciones web (Molina Ríos et al., 2016).

2.10.2. Django Rest Framework

Django REST Framework es un conjunto de herramientas utilizado para desarrollar aplicaciones web, facilitando la creación de APIs REST. Aunque es capaz de gestionar niveles de API menos avanzados, no está diseñado para soportar un nivel específico de hipermedios. El uso de la implementación HATEOAS depende del programador o de otros proyectos adicionales, como srf-hal-json, si desean seguir este tipo de implementación en su API. or lo

tanto, es posible integrar un enfoque HATEOAS en una API utilizando este framework (eBook, 2019).

2.10.3. Vue

Vue es una herramienta JavaScript desarrollada por Evan You, quien es reconocido en la comunidad por su participación en el desarrollo de Meteor y por haber trabajado en Google durante varios años. Evan You describe su herramienta como un framework progresivo, ya que está compuesto por varias librerías pequeñas y bien definidas, cada una con una función específica. Esto permite que el desarrollador agregue solo los módulos necesarios según las necesidades del proyecto, evitando la necesidad de incluir toda la funcionalidad desde el principio, como ocurre con frameworks como AngularJS.

Vue es similar en su enfoque modular al sistema de ReactJS, Facebook creó un núcleo básico para trabajar con vistas, y a partir de ahí, se han desarrollado diversas librerías, tanto por Facebook como por la comunidad, que permiten trabajar de manera eficiente en una SPA (Single Page Application). Todo esto forma parte del proyecto VueJS creado por Evan You.

El núcleo principal de Vue facilita el desarrollo de componentes de UI mediante JavaScript. La librería se enmarca dentro de arquitecturas basadas en componentes, que son muy populares en la actualidad, y gestiona internamente los modelos utilizando el patrón MVVM. Esto implica que los componentes tienen mecanismos de doble "data-binding" para manipular el estado de la aplicación (Sánchez, 2018).

VueJS se distingue por diversas características que lo hacen una opción popular y poderosa dentro del mundo de los frameworks JavaScript. A continuación, se detallan algunos de sus aspectos más destacados:

- **Componentes Visuales Reactivos:** VueJS permite la creación de componentes visuales que responden de manera reactiva a los eventos. Estos componentes están bien encapsulados, exponiendo una API simple que facilita la entrada de

propiedades y la emisión de eventos. Gracias a su naturaleza reactiva, los componentes gestionan los eventos masivos sin que el rendimiento se vea afectado.

- **Conceptos Claros de Directivas, Filtros y Componentes:** VueJS organiza y estructura bien sus elementos clave, como las directivas, filtros y componentes, facilitando la comprensión y uso del framework. Cada uno de estos elementos cumple con un propósito específico y es fácil de entender a medida que se profundiza en la plataforma.
- **Uso de Virtual DOM:** VueJS emplea un Virtual DOM, lo cual es clave para mejorar el rendimiento. Dado que las operaciones con el DOM real son costosas, VueJS mantiene una copia cacheada del DOM.
- **Externalización de Ruteo y Gestión de Estado:** Aunque VueJS ofrece un núcleo potente, delega el ruteo y la gestión de estado a otras librerías especializadas, como Vue Router y Vuex, lo que permite una mayor modularidad y flexibilidad en las aplicaciones.
- **Sistema de Efectos de Transición y Animación:** VueJS cuenta con un sistema robusto para manejar transiciones y animaciones, lo que facilita la implementación de efectos visuales de manera sencilla y eficiente, mejorando la experiencia del usuario en la aplicación (Sanchez, 2018).

2.10.4. Pytorch

PyTorch es una biblioteca de Python diseñada para facilitar la creación de proyectos de aprendizaje profundo. Su flexibilidad y enfoque intuitivo la convierten en una herramienta ideal tanto para principiantes como para profesionales en la industria. Desde su lanzamiento inicial, PyTorch se ha consolidado como una de las principales opciones en el ámbito del aprendizaje profundo, abarcando aplicaciones como traducción automática, juegos de estrategia e identificación de objetos en entornos complejos (Stevens et al., 2020).

Características principales de PyTorch:

- Su sintaxis clara y diseño "Pythonic" permiten que los desarrolladores que ya conocen Python la adopten rápidamente.
- PyTorch facilita la depuración e iteración incremental, lo que lo hace ideal tanto para aprendizaje como para investigación y producción.
- Los tensores, estructura central de datos de PyTorch, son matrices multidimensionales similares a las de NumPy, pero con soporte integrado para operaciones aceleradas por GPU.
- Esto permite manejar de forma eficiente grandes cantidades de datos y ejecutar operaciones matemáticas complejas.
- Utiliza unidades de procesamiento gráfico (GPU), logrando aceleraciones significativas (hasta 50 veces más rápidas que en CPU para ciertos cálculos).
- Su diseño modular hace que se adapte fácilmente a diversas aplicaciones, desde investigación hasta despliegues en producción.
- PyTorch hereda muchas características de NumPy, lo que facilita a los desarrolladores la transición y reutilización de conocimientos (Stevens et al., 2020).

2.11.PRUEBAS DE SOFTWARE

2.11.1. Pruebas Funcionales

Las pruebas funcionales se basan en las funcionalidades de un sistema que están descritas en la especificación de requisitos, es decir, en lo que el sistema debe hacer. Estas funcionalidades también pueden no estar documentadas, pero requieren un nivel elevado de experiencia para ser interpretadas correctamente. Según la ISO 25010, las características de la funcionalidad incluyen idoneidad, exactitud, interoperabilidad y seguridad. En la misma norma se señala que "la funcionalidad representa la capacidad del producto de software para proporcionar funciones que satisfacen las necesidades declaradas e implícitas, cuando el

producto se usa en condiciones específicas". Esta funcionalidad se desglosa en las siguientes características:

- **Completitud funcional:** el grado en que las funcionalidades cubren todas las tareas y objetivos del usuario especificados.
- **Corrección funcional:** la capacidad del producto o sistema para proporcionar resultados correctos con el nivel de precisión requerido.
- **Pertenencia funcional:** la capacidad del producto de software para ofrecer un conjunto adecuado de funciones para las tareas y objetivos especificados por el usuario.

Estas pruebas pueden realizarse en todos los niveles de prueba, como las pruebas de componentes basadas en una especificación. Las pruebas funcionales suelen asociarse con técnicas de diseño de caja negra, ya que se enfocan en el comportamiento externo del software (Peño Sánchez, 2015).

2.11.2. Pruebas de Integración

Las pruebas de integración se utilizan para verificar que las unidades de prueba, que han pasado las pruebas unitarias, funcionen correctamente al ser integradas. El objetivo principal de estas pruebas es evaluar la arquitectura del software. Durante el proceso de integración, se emplean principalmente técnicas de caja negra, aunque también pueden realizarse algunas pruebas de caja blanca para asegurar que se cubren los flujos principales de comunicación entre las unidades.

En el contexto de la programación orientada a objetos, las pruebas de integración buscan garantizar que los mensajes enviados desde los objetos de una clase o componente se entreguen y reciban en el orden correcto en el objeto receptor, y que estos mensajes generen los cambios de estado esperados en el receptor (Polo Usaola, 2008).

2.11.3. Pruebas Unitarias

Las pruebas unitarias se enfocan en lo que se denomina la "unidad de prueba", que, dependiendo del contexto, puede ser una clase, un método o un subsistema. El estándar ANSI/IEEE 1008/1987 define la unidad de prueba como un conjunto de uno o más módulos de un programa, junto con los datos de control asociados (como tablas), procedimientos de uso y procedimientos de operación, siempre que cumplan las siguientes condiciones:

- Todos los módulos pertenecen a un único programa.
- Al menos uno de los módulos nuevos o modificados del conjunto no ha pasado las pruebas unitarias (ya que una unidad de prueba puede incluir uno o más módulos previamente probados).
- El conjunto de módulos junto con sus datos y procedimientos asociados son el único objetivo del proceso de pruebas.

En el contexto de la programación orientada a objetos, generalmente se considera que la unidad de prueba es la clase. Se verifica que el estado de la instancia de la clase bajo prueba sea correcto para los datos proporcionados como entrada. De esta manera, las pruebas unitarias de caja negra tratan a la clase como una "caja" cuyo contenido interno no es relevante; lo único que importa en este enfoque son las entradas suministradas y las salidas obtenidas (Polo Usaola, 2008).

2.12. MÉTRICAS DE CALIDAD

2.12.1. ISO/IEC 25010

La norma ISO/IEC 25010 forma parte de la familia de normas ISO 25000 y está orientada a evaluar la calidad del producto software. Su enfoque principal es la usabilidad, pero abarca una gama completa de características de calidad que deben considerarse al evaluar las propiedades de un software terminado. Esta norma establece que la calidad del software se define como el grado en que satisface los requisitos de sus usuarios, proporcionando valor al cumplir con sus expectativas y necesidades.

- **Adecuación Funcional:** Hace referencia a la completitud, corrección y pertinencia funcional.
- **Eficiencia de desempeño:** se enfoca en el comportamiento temporal, la utilización de recursos y la capacidad del sistema.
- **Compatibilidad:** Evalúa la coexistencia y la interoperabilidad del software con otros sistemas.
- **Usabilidad:** Mide aspectos como la capacidad de reconocer su adecuación, la facilidad de aprendizaje, la facilidad de uso, la protección contra errores de usuario, la estética de la interfaz y la accesibilidad.
- **Fiabilidad:** Analiza la madurez, la disponibilidad, la tolerancia a fallos y la capacidad de recuperación.
- **Seguridad:** Abarca la confidencialidad, la integridad, el no repudio, la responsabilidad y la autenticidad (Mera Paz et al., 2017).

Figura 14

ISO/IEC 25010



Nota. Adaptado de *Garantía de Calidad del Software*, por Invensity, 2024.

2.13. MÉTRICAS DE SEGURIDAD

2.13.1. ISO/IEC 27002

La norma ISO 27002 es un estándar internacional que ofrece directrices detalladas para la implementación de controles de seguridad de la información. A diferencia de la ISO 27001, que se enfoca en los requisitos para establecer un Sistema de Gestión de la Seguridad de la Información (SGSI), la ISO 27002 se presenta como un complemento a esta última.

Esta norma proporciona un conjunto exhaustivo de directrices y mejores prácticas para implementar los controles de seguridad. Se posiciona como un recurso esencial para aquellas organizaciones que buscan adoptar un enfoque sólido y alineado con las mejores prácticas en seguridad de la información, facilitando la implementación efectiva de medidas de protección.

La norma ISO 27002 es aplicable a todas las organizaciones, independientemente de su tamaño, tipo o industria. Su propósito es ayudar a las organizaciones a seleccionar e implementar controles de seguridad adecuados, de acuerdo con los riesgos que enfrentan (globalsuitesolutions, 2024).

CAPÍTULO III

MARCO APLICATIVO

**INGENIERÍA
DE SISTEMAS**
UNIVERSIDAD PÚBLICA DE EL ALTO



CAPÍTULO III

3. MARCO APLICATIVO

3.1. INTRODUCCIÓN

En este capítulo realizaremos el análisis, diseño y desarrollo del Tutor Inteligente, utilizando la metodología MeISE el cual tiene un enfoque educativo, así como también estaremos realizando la combinación con la metodología CRISP-DM el cual nos permitirá trabajar con redes neuronales artificiales. A partir de ahora el Sistema Tutor Inteligente será denominado InternAI.

Tabla 8

Combinación de la metodología MeISE y CRISP-DM

Fases	Metodología MeISE	Metodología CRISP-DM	Tareas
Fase de Definición	Etapa Conceptual	Comprensión del Negocio	Realizar el análisis tanto del sistema web como del modelo de IA.
Análisis, Diseño y Comprensión de Datos	Análisis y Diseño Inicial	Comprensión de Datos	Establecer los requerimientos funcionales y no funcionales del sistema. Establecer la arquitectura del software. Seleccionar los datos para el modelo de red neuronal.
Planeación de Iteraciones y Preparación de Datos	Plan de Iteraciones	Preparación de Datos	Preparar iteraciones del sistema. Estructurar, formatear y limpiar los datos.

Fases	Metodología MeISE	Metodología CRISP-DM	Tareas
Diseño computacional y modelado	Diseño Computacional	Modelado	Elaborar el diseño computacional (diagramas de clases y secuencias, diagrama de base de datos). Establecer los parámetros del modelo, para su posterior entrenamiento.
Desarrollo y evaluación	Desarrollo	Evaluación	Desarrollar el sistema y evaluar las predicciones del modelo.
Despliegue	Despliegue	Implantación	Entregar el sistema funcional a los usuarios.

3.2. FASE DE DEFINICIÓN

En esta primera fase, se comienza con el análisis de las necesidades educativas de los estudiantes en el Internado Rotatorio, trabajando de la mano con el Dr. Luis Flores, Coordinador del Internado Rotatorio, quien facilitará la selección de textos de diferentes temas del internado rotatorio.

Al mismo tiempo, siguiendo CRISP-DM, se especifican y analizan los criterios de éxito, como la precisión en el contenido médico, la capacidad de respuesta del modelo y la necesidad de datos (textos de medicina) de alta calidad.

3.2.1. *Análisis de Necesidades Educativas*

Se establecen los temas esenciales de cada área del internado rotatorio (Medicina Interna, Cirugía, Ginecología y Obstetricia, Pediatría, y Salud Pública) en forma de módulos de estudio que contengan PDFs y cuestionarios generados por el modelo. Los temas esenciales y disponibilidad de datos (textos de medicina) se definió con el Dr. Luis Flores

3.2.2. Revisión de Alternativas de Solución

Se evalúan las herramientas y plataformas de desarrollo necesarias para cumplir con los requisitos del sistema, seleccionando tecnologías óptimas como Vue y Vuetify como frontend, Django como backend y PostgreSQL como gestor de base de datos. Estas opciones facilitarán la implementación de un sistema interactivo y estable para el contexto educativo médico.

3.2.3. Estudio de Riesgos

Se documentan los riesgos técnicos y pedagógicos, tales como la precisión de respuestas generadas por la red neuronal y la usabilidad del sistema.

3.2.4. Identificación de Funcionalidades del Software

Se definirá los tipos de usuario principales (estudiantes y personal institucional) y sus casos de uso específicos.

Figura 15

Arquitectura STI del sistema

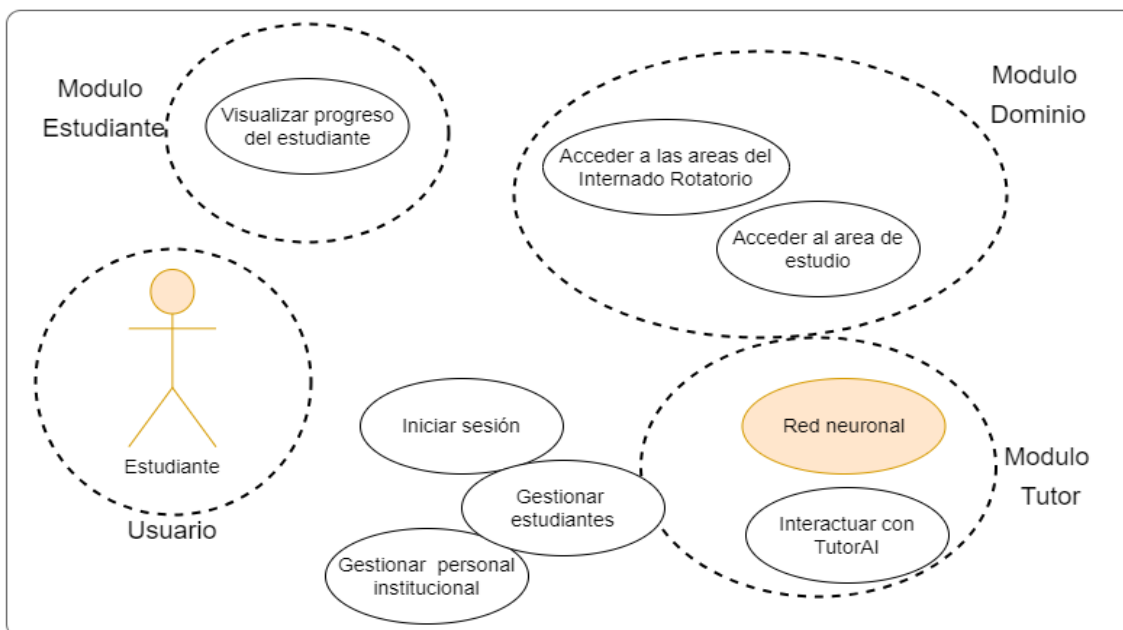


Figura 16

Caso de Uso de Alto Nivel

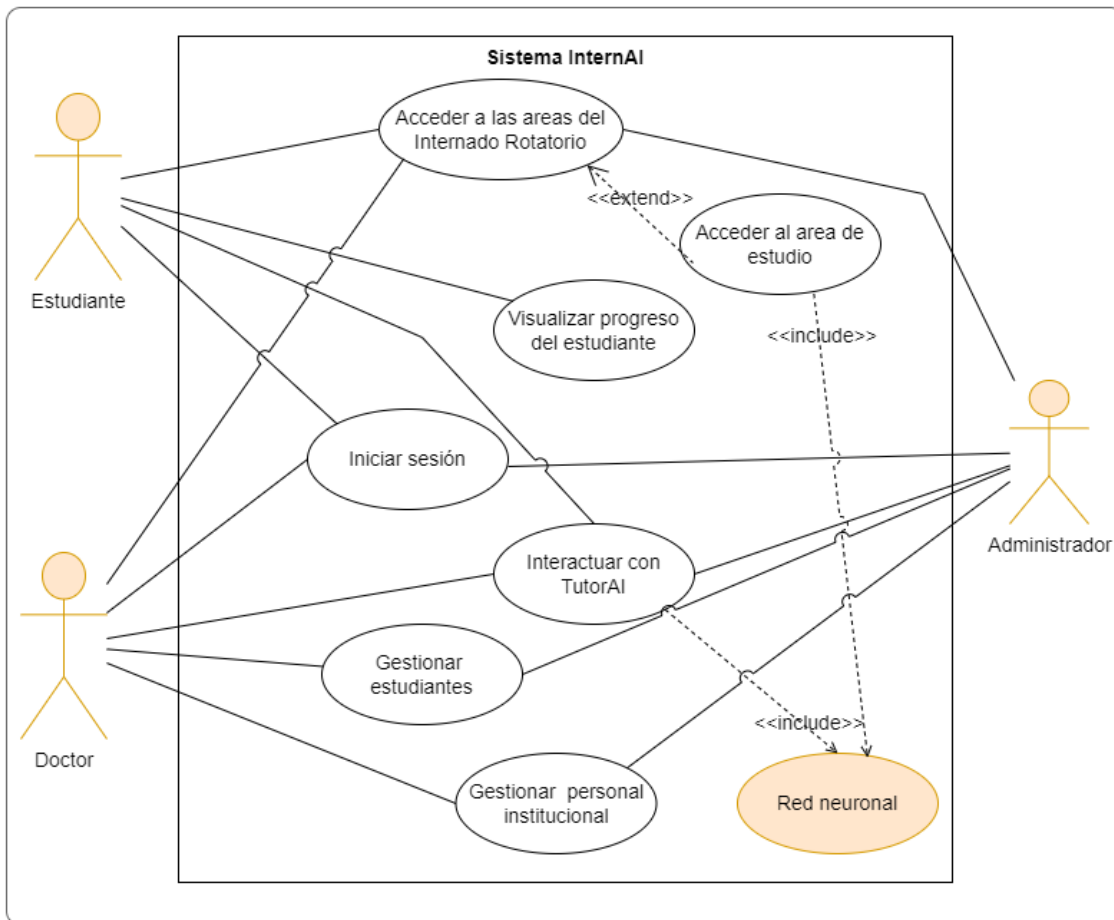
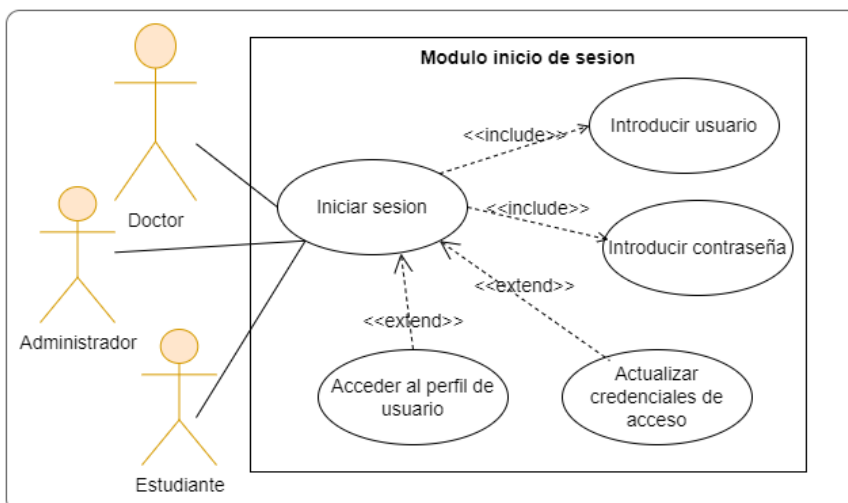


Figura 17

Caso de Uso Expandido Inicio de sesión



El usuario accede al sistema a través de un formulario de inicio de sesión. Tras ingresar sus credenciales, el sistema valida la información y permite el acceso a la interfaz principal de InternAI, donde podrá interactuar con los distintos módulos disponibles según sus permisos de usuario. El usuario tiene la opción de acceder al perfil de usuario y actualizar las credenciales de inicio de sesión.

Tabla 9

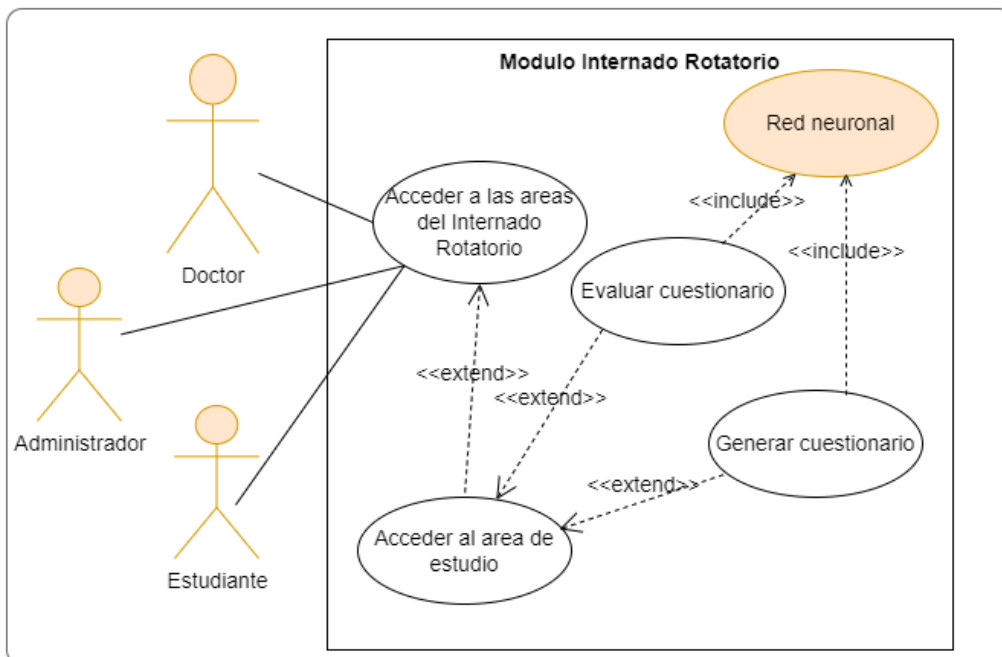
Especificaciones de Caso de Uso Expandido Inicio de sesión

Caso de uso expandido	
Caso de uso:	Inicio de sesión
Fecha de Creación:	01/09/2024
Actores:	Doctor, Administrador, Estudiante
Descripción:	Ingresar al sistema
Restricción:	Ninguna
Precondiciones:	Debe estar registrado en el sistema.
Flujo normal:	<ol style="list-style-type: none"> 1. Introducir el usuario. 2. Introducir la contraseña. 3. Ingresar.
Flujo Alternativo:	<ol style="list-style-type: none"> 1. Introducir el usuario. 2. Introducir la contraseña. 3. Ingresar. 4. Contraseña incorrecta.
Excepciones	
Incluye:	Ninguno
Frecuencia de uso:	Alta
Prioridad:	Alta

Caso de uso expandido

Regla del negocio

 Ninguno

Figura 18
Caso de Uso Expandido Módulo Internado Rotatorio


El estudiante navega hasta la sección de áreas del internado rotatorio, donde puede seleccionar el área (medicina interna, cirugía, ginecología obstetricia, pediatría, o salud pública). Dentro del área seleccionada, el estudiante elige un tema específico para estudiar y accede al material de apoyo en formato PDF. Al finalizar el estudio del tema, el estudiante tiene la opción de responder un cuestionario generado por la red neuronal. Una vez completado, el estudiante debe solicitar la respectiva evaluación del cuestionario, automáticamente la red neuronal procederá con la evaluación.

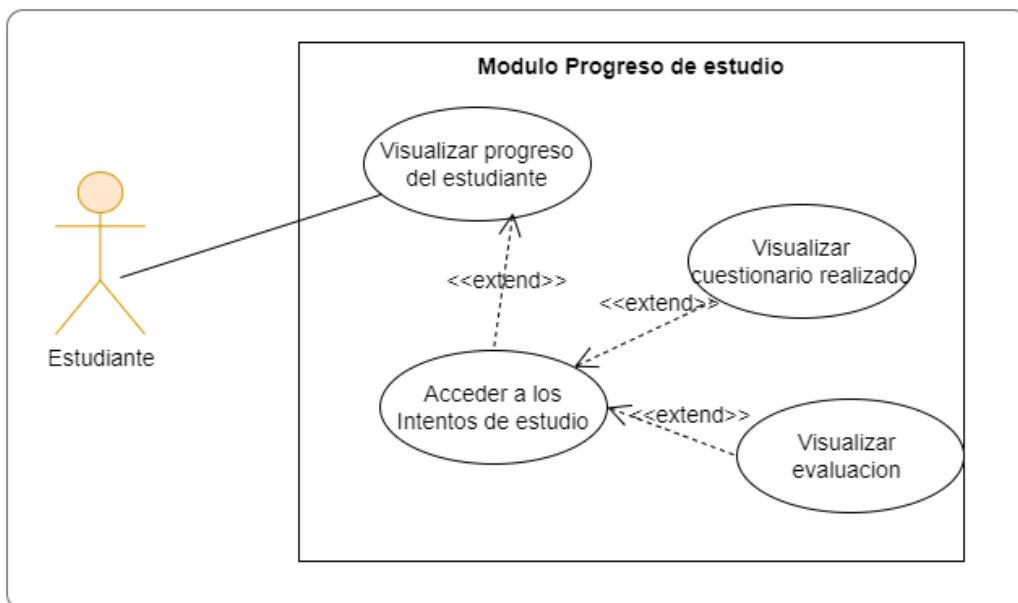
Tabla 10

Especificaciones de Caso de Uso Módulo Internado Rotatorio

Caso de uso expandido	
Caso de uso:	Módulo Internado Rotatorio
Fecha de Creación:	01/09/2024
Actores:	Doctor, Administrador, Estudiante
Descripción:	Acceder a las áreas del Internado Rotatorio, se visualizará, temas del I.R. por área.
Restricción:	Ninguna
Precondiciones:	Debe estar registrado en el sistema.
Flujo normal:	<ol style="list-style-type: none"> 5. Ir al menú, opción "Internado Rotatorio" 6. Elegir un tema de una determinada área del I.R. 7. Acceder al área de estudio. 8. Generar cuestionario. 9. Solicitar evaluación del cuestionario a la red neuronal.
Flujo Alternativo:	<ol style="list-style-type: none"> 1. Opción "Internado Rotatorio". 2. Elegir un tema. 3. Estudiar. 4. Responder cuestionario. 5. Solicitar evaluación del cuestionario. 6. La red neuronal procesará y evaluará el cuestionario.
Excepciones	
Incluye:	Ninguno
Frecuencia de uso:	Alta
Prioridad:	Alta
Regla del negocio	Usuarios registrados

Figura 19

Caso de Uso Expandido Módulo progreso de estudio



El estudiante puede ver el progreso de estudio en curso donde se muestran sus intentos y el avance en cada área y tema de estudio.

Tabla 11

Especificaciones de Caso de Uso Expandido Progreso de estudio

Caso de uso expandido	
Caso de uso:	Módulo progreso de estudio
Fecha de Creación:	01/09/2024
Actores:	Estudiante
Descripción:	Visualizar progreso de estudio, donde se encuentra el avance y los intentos de los temas estudiados.
Restricción:	Ninguna
Precondiciones:	Solo usuarios "Estudiantes" tiene acceso a este módulo.
Flujo normal:	<ol style="list-style-type: none"> 1. Ir menú, opción "Progreso de estudio". 2. Ver detalles de un determinado tema en progreso.

Caso de uso expandido

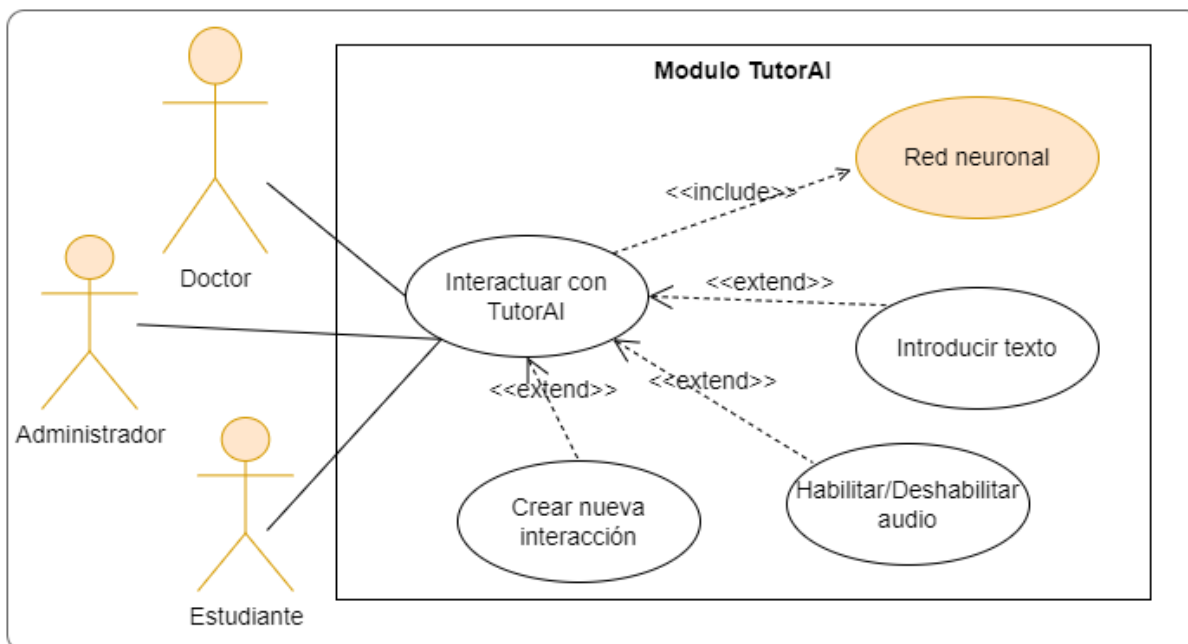
Flujo Alternativo:	3. Visualizar intentos de los cuestionarios realizados.
	1. Opción "Progreso de estudio".
	2. Ver detalles de un determinado tema en progreso.
	3. Visualizar intentos de los cuestionarios realizados.

Excepciones

Incluye:	Ninguna
Frecuencia de uso:	Alta
Prioridad:	Alta
Regla del negocio	Solo usuarios de tipo "Estudiante" tiene acceso

Figura 20

Caso de Uso Expandido Módulo TutorAI



El usuario interactúa con TutorAI, donde puede realizar preguntas directamente al TutorAI sobre temas de medicina.

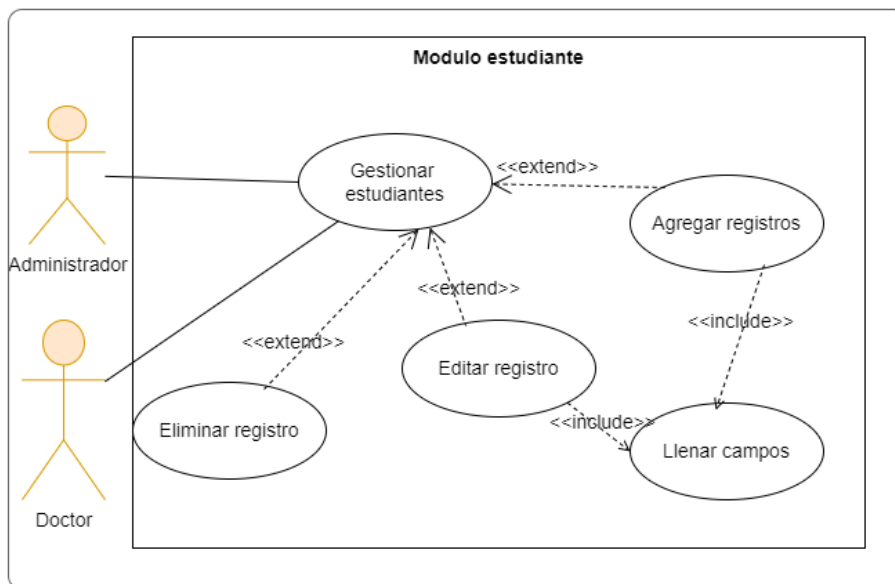
Tabla 12

Especificaciones de Caso de Uso Expandido Módulo TutorAI

Caso de uso expandido	
Caso de uso:	Módulo TutorAI
Fecha de Creación:	01/09/2024
Actores:	Estudiantes, Doctor, Administrativo
Descripción:	Interactuar con TutorAI, donde se puede realizar preguntas sobre temas de medicina.
Restricción:	Ninguna
Precondiciones:	Debe estar registrado en el sistema.
Flujo normal:	<ol style="list-style-type: none"> 1. Ir menú, opción "TutorAI". 2. Escribir el prompt (texto, pregunta o instrucción). 3. TutorAI (red neuronal) recibirá como entrada el texto del usuario y retornará una respuesta en tiempo real. 4. Activar/Desactivar audio si es necesario. 5. Crear nueva interacción.
Flujo Alternativo:	<ol style="list-style-type: none"> 1. Opción "TutorAI". 2. Escribir el prompt (texto, pregunta o instrucción). 3. TutorAI (red neuronal) recibirá como entrada el texto del usuario y retornará una respuesta en tiempo real.
Excepciones	
Incluye:	Escribir los prompts de forma correcta.
Frecuencia de uso:	Alta
Prioridad:	Alta
Regla del negocio	TutorAI se debe usar con responsabilidad y verificar la información proporcionada, el modelo puede cometer errores.

Figura 21

Caso de Uso Expandido Modulo estudiante



El personal autorizado puede gestionar estudiantes en el sistema, proporcionando información básica que permitirá al estudiante acceder al sistema.

Tabla 13

Especificaciones de Caso de Uso Expandido Módulo estudiante

Caso de uso expandido	
Caso de uso:	Módulo estudiante
Fecha de Creación:	01/09/2024
Actores:	Administrador, Doctor
Descripción:	Gestionar estudiantes. Agregar, editar, eliminar estudiantes.
Restricción:	Solo usuarios con los permisos suficientes para administrar el módulo estudiante.
Precondiciones:	Debe estar registrado en el sistema.
Flujo normal:	1. Opción "Estudiantes".

Caso de uso expandido

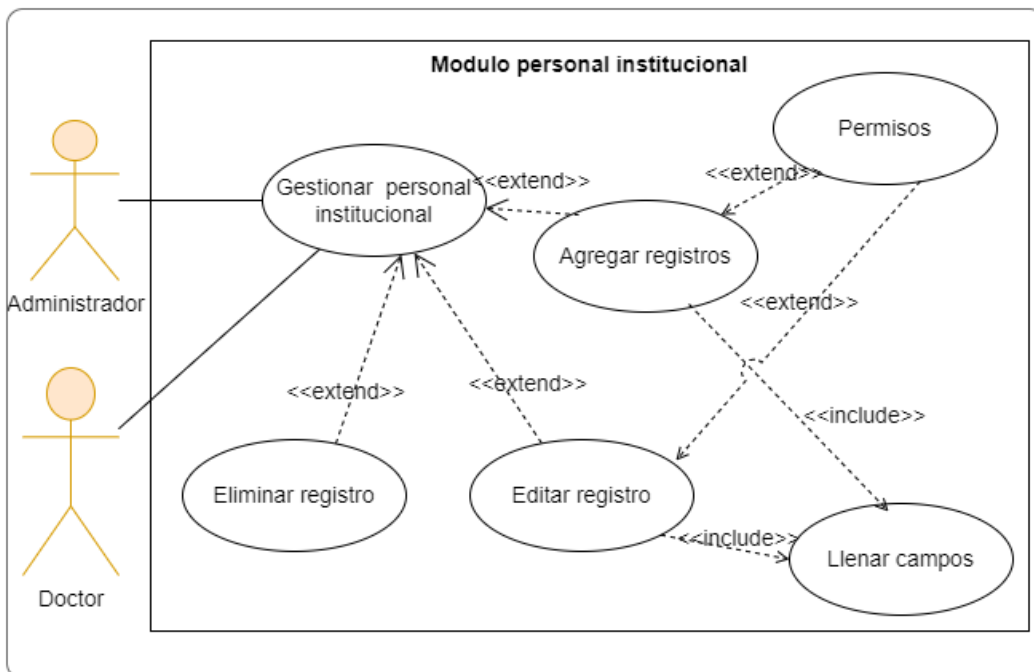
2. Esperar la carga de datos.
3. Agregar / Editar.
4. Enviar.
5. El sistema valida los campos si es necesario.
6. Corregir campos si es necesario.
7. Enviar.

Excepciones

Incluye:	Buscar registro
Frecuencia de uso:	Alta
Prioridad:	Alta
Regla del negocio	La información para rellenar los campos debe ser válido.

Figura 22

Caso de Uso Expandido Modulo Personal institucional



Los administradores del sistema pueden registrar nuevos miembros del personal, asignándoles permisos específicos de acceso y gestión dentro del sistema InternAI.

Tabla 14

Especificaciones de Caso de Uso expandido Módulo personal

Caso de uso expandido	
Caso de uso:	Módulo personal institucional
Fecha de Creación:	01/09/2024
Actores:	Administrador, Doctor
Descripción:	Gestionar personal institucional. Agregar, editar, eliminar y personal.
Restricción:	Solo usuarios con los permisos suficientes para administrar el módulo "institucional".
Precondiciones:	Debe estar registrado en el sistema.
Flujo normal:	<ol style="list-style-type: none"> 1. Ir menú, opción "Institucional". 2. Esperar la carga de datos. 3. Opción Agregar / Editar. 4. Llenar campos si es necesario. 5. Conceder permisos si es necesario. 6. Enviar. 7. El sistema valida los campos si es necesario. 8. Corregir campos si es necesario. 9. Enviar.
Flujo Alternativo:	<ol style="list-style-type: none"> 1. Opción "Institucional". 2. Eliminar registro. 3. Visualizar mensaje de respuesta del sistema.
Excepciones	
Incluye:	Buscar registro

Caso de uso expandido	
Frecuencia de uso:	Media
Prioridad:	Alta
Regla del negocio	La información para rellenar los campos debe ser válido para que el sistema procese de forma adecuada.

3.2.5. Criterios de Medición

Se establece criterios que aseguren una experiencia de usuario fluida, una interfaz intuitiva, y una funcionalidad precisa en la generación y evaluación de preguntas y respuestas generadas por el modelo.

3.2.6. Determinación de los Objetivos del Negocio

Se define que la red neuronal basado en un LLM debe generar contenido educativo en medicina para estudiantes en el internado rotatorio. Esto incluye preguntas y retroalimentación en temas como Medicina Interna, Cirugía, Pediatría, Ginecología y Obstetricia, y Salud Pública. El éxito del modelo se medirá en función de su precisión al generar preguntas relevantes y su capacidad para evaluar a los estudiantes.

Para el entrenamiento y procesamiento de datos se utilizará Python, ya que este es potente en el campo de la Inteligencia Artificial.

3.2.7. Determinación de los Objetivos de Data Mining

Los objetivos de Data Mining se enfocan en obtener una red neuronal que sea capaz de generar preguntas de evaluación y retroalimentación coherente en los temas de cada área. Esto implica transformar los objetivos de negocio en metas técnicas específicas para el entrenamiento del modelo.

3.3. ANALISIS, DISEÑO Y COMPRENSIÓN DE DATOS

En esta fase, realizamos el diseño de la estructura del sistema web y el flujo educativo. Esto incluye el desarrollo de los módulos y la interfaz que permitirá a los estudiantes acceder

a áreas específicas del internado, como Medicina Interna y Ginecología, y ver temas y cuestionarios de cada área.

Simultáneamente, utilizando la fase de comprensión de datos de CRISP-DM, se recolectan y se exploran los datos médicos (textos de diferentes áreas del internado rotatorio), evaluando su calidad y relevancia conjuntamente con el Dr. Luis Flores. Esto garantiza que el modelo tenga acceso a datos de alta calidad para su posterior entrenamiento.

3.3.1. *Requerimientos Funcionales y No Funcionales*

A continuación se describe los requerimientos funcionales y no funcionales

Tabla 15

Requerimientos funcionales

Id único	Requerimientos funcionales	Descripción	Prioridad
RF01	Iniciar sesión	Permitir el inicio de sesión.	Alta
RF02	Perfil de usuario	Visualizar los datos del usuario, cambiar credenciales de acceso al sistema	Alta
RF03	Registrar estudiantes	Permitir la creación de cuentas para nuevos estudiantes con información básica y de contacto.	Alta
RF04	Editar estudiantes	Permitir la actualización de los datos de los estudiantes.	Alta
RF05	Eliminar estudiantes	Permitir la eliminación de cuentas.	Alta
RF06	Visualizar datos del estudiante	Visualizar los datos del estudiante.	Media
RF07	Registrar personal institucional	Permitir la creación de cuentas para nuevos administrativos con información básica y de contacto.	Alta

Id único	Requerimientos funcionales	Descripción	Prioridad
RF08	Editar personal institucional	Permitir la actualización de los datos del personal institucional.	Alta
RF09	Eliminar personal institucional	Permitir la eliminación de cuentas.	Alta
RF10	Visualizar datos del personal institucional	Visualizar los datos del personal institucional.	Media
RF11	Permisos de acceso	Limitar el acceso a ciertas funcionalidades según los permisos asignado a cada usuario.	Alta
RF12	Entrenar la red neuronal	Seleccionar textos de las diferentes áreas del Internado Rotatorio, procesar textos y entrenar la red neuronal.	Alta
RF13	Visualizar áreas disponibles del Internado Rotatorio	Mostrar las áreas disponibles (medicina interna, cirugía, ginecología y obstetricia, pediatría, salud pública), y permitir el acceso a temas específicos en cada área.	Alta
RF14	Visualizar temas de estudio	En cada área, permitir el acceso a temas de estudio que incluyan PDFs.	Alta
RF15	Generar cuestionarios	Generar cuestionarios dinámicos basados en los temas de estudio mediante el modelo de red neuronal.	Alta
RF16	Evaluar cuestionarios	Evaluar las respuestas de los estudiantes utilizándola red neuronal, proporcionando retroalimentación inmediata.	Alta

Id único	Requerimientos funcionales	Descripción	Prioridad
RF17	Interactuar con el modelo	Permitir al estudiante realizar preguntas abiertas sobre los temas de estudio y recibir respuestas generadas por el modelo.	Alta
RF18	Generación de audio	Una vez generado el texto, se procede a generar audio en base al texto del modelo.	Media
RF19	Visualizar el progreso del estudiante	Visualizar el progreso de estudio.	Alta
RF20	Almacenamiento de progresos	Guardar los datos como tiempo, fecha y progreso de estudio.	Alta

Figura 23*Requerimientos no funcionales*

Id único	Requerimientos no funcionales	Descripción	Prioridad
RNF01	Rendimiento del sistema	Optimizar tiempos de carga y procesamiento, especialmente en la generación y evaluación de preguntas.	Alta
RNF02	Escalabilidad	Permitir el crecimiento en el número de usuarios y datos, asegurando el rendimiento constante del sistema.	Alta
RNF03	Seguridad de datos	Proteger la información sensible de los usuarios, utilizando cifrado y protocolos seguros de autenticación y autorización.	Alta
RNF04	Soporte para dispositivos móviles	Adaptar la interfaz para dispositivos móviles y tablets, asegurando una experiencia de	Alta

Id único	Requerimientos no funcionales	Descripción	Prioridad
RNF05	Usabilidad	usuario fluida en diferentes tamaños de pantalla. Garantizar la usabilidad del sistema, brindando una interfaz intuitiva.	Media
RNF06	Visualización de cargas	Mientras se carga un recurso del backend, visualizar progresos (progress circular y progress linear).	Media

3.3.2. Establecimiento de la Arquitectura del Software

Se establece una arquitectura de software modular, en la cual el frontend, desarrollado con Vue.js y estilizado con Vuetify para proporcionar una experiencia de usuario visualmente atractiva y dinámica, interactúa de manera eficiente con el backend, implementado en Django Rest Framework. La comunicación entre estas dos capas se realiza a través de API REST diseñadas bajo principios de buenas prácticas en desarrollo de software, asegurando seguridad, eficiencia y escalabilidad.

El backend no solo actúa como el núcleo lógico de la aplicación, sino que también gestiona el acceso a los datos mediante una conexión estable y segura con el gestor de base de datos PostgreSQL, que almacena toda la información estructurada y necesaria para el funcionamiento del sistema.

3.3.3. Diseño Educativo y Diseño de Comunicación

Se diseña una estructura navegable para que el estudiante pueda avanzar fácilmente en cada área.

Figura 24

Modelo de Navegación

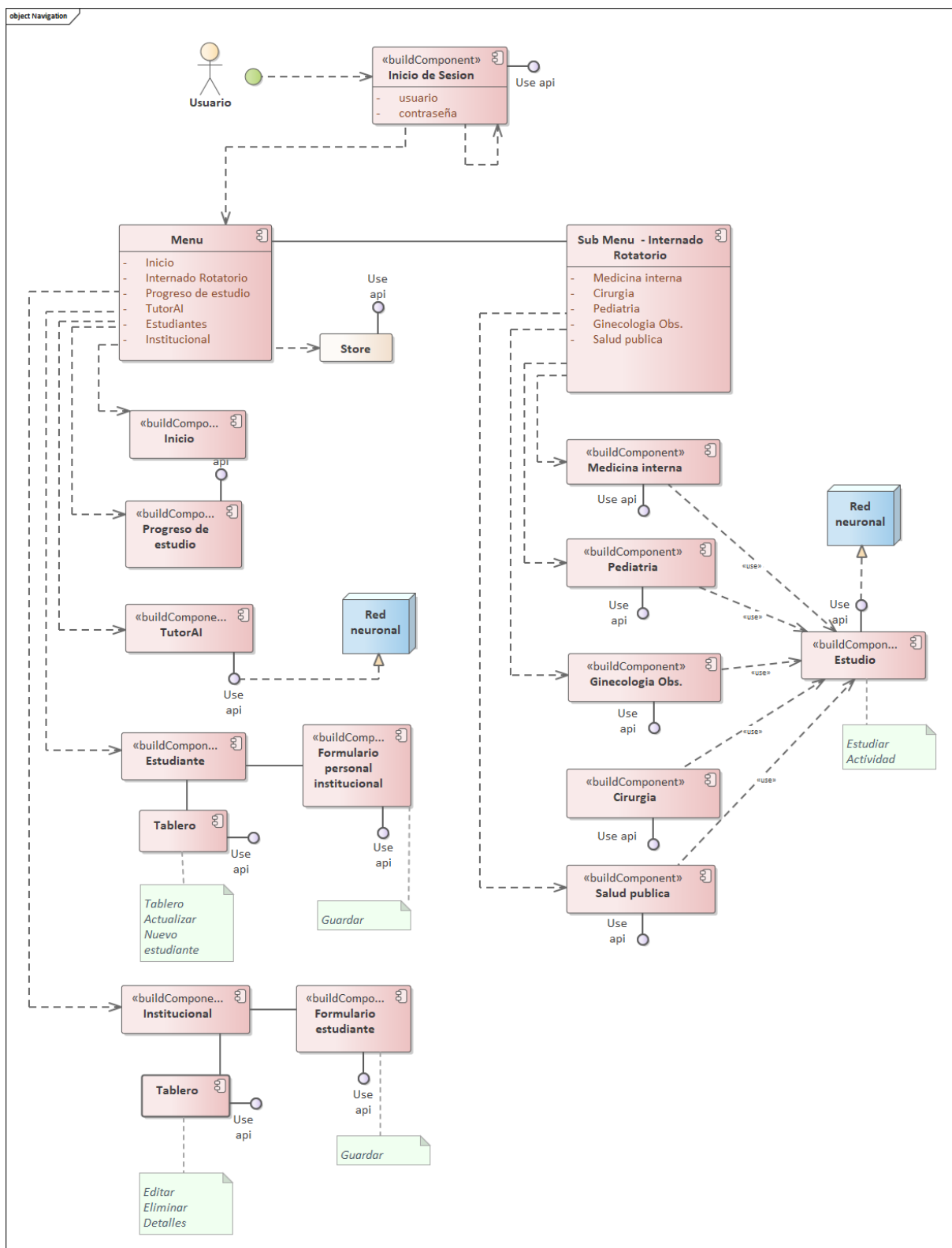


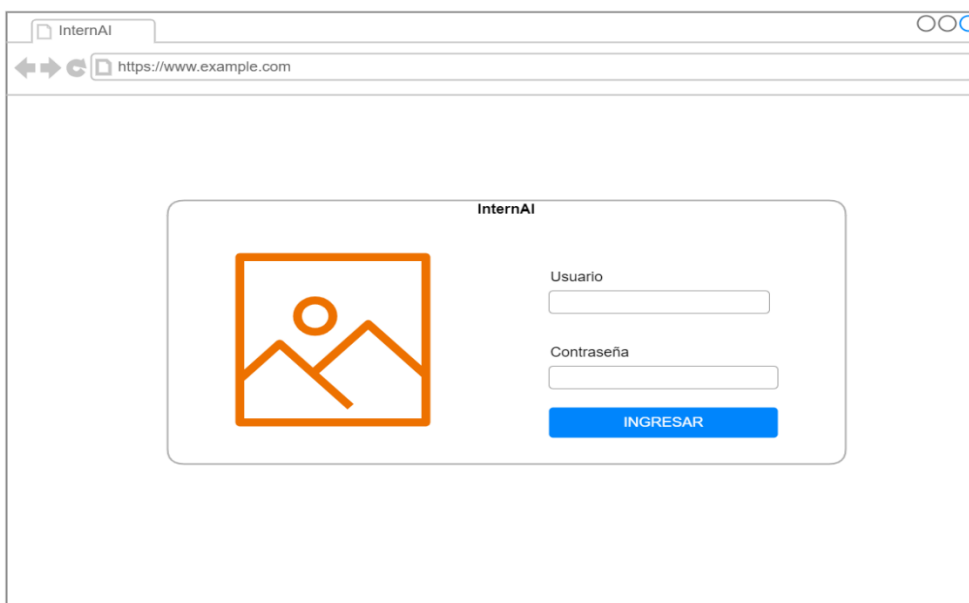
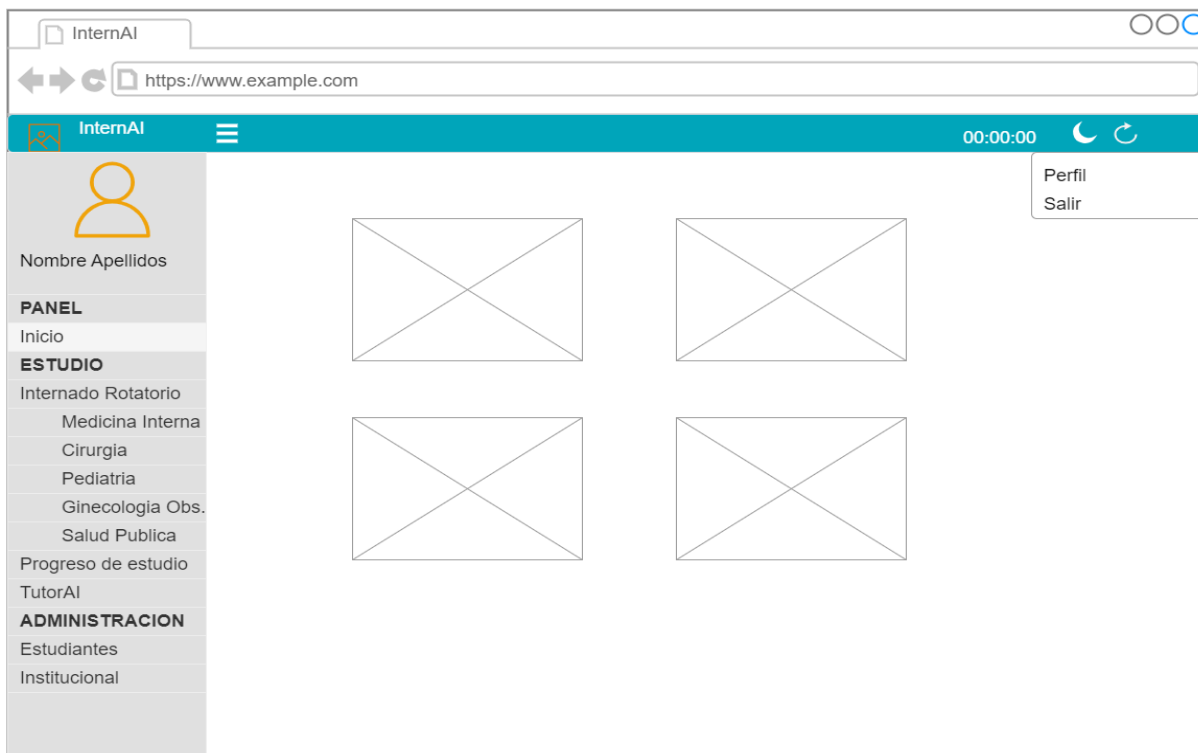
Figura 25*Prototipo, Inicio de sesión***Figura 26***Prototipo, Vista inicio*

Figura 27

Prototipo, Vista Internado Rotatorio

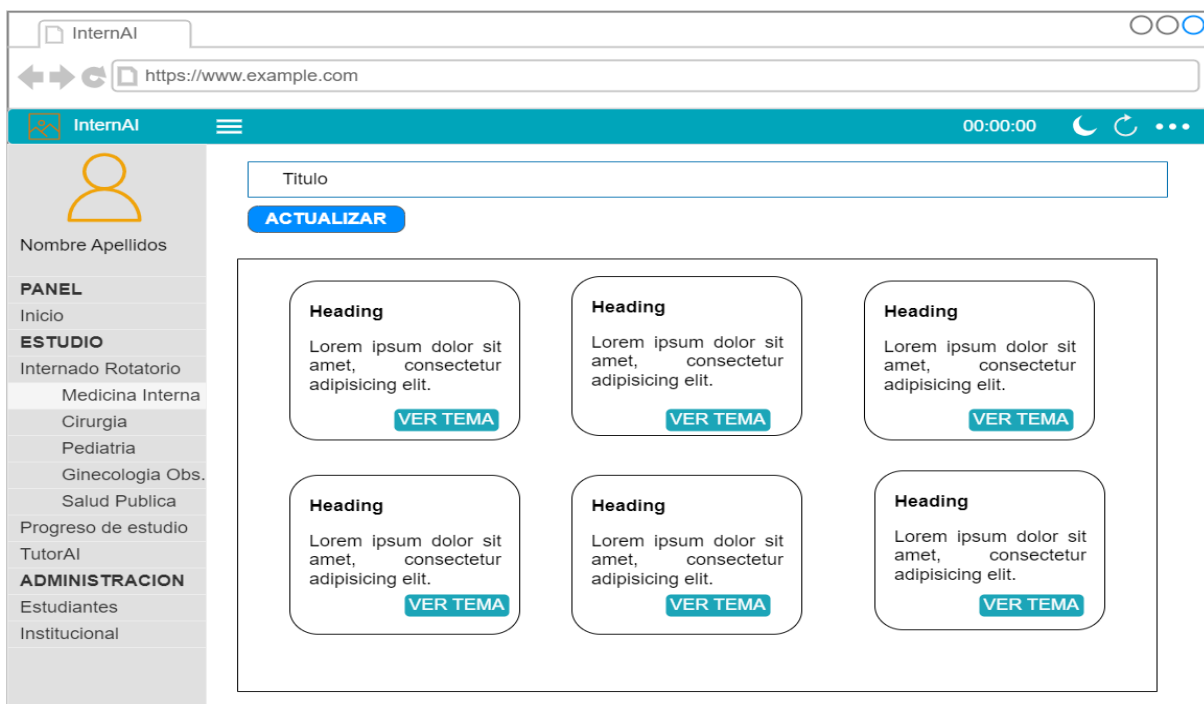


Figura 28

Prototipo, Vista área de estudio

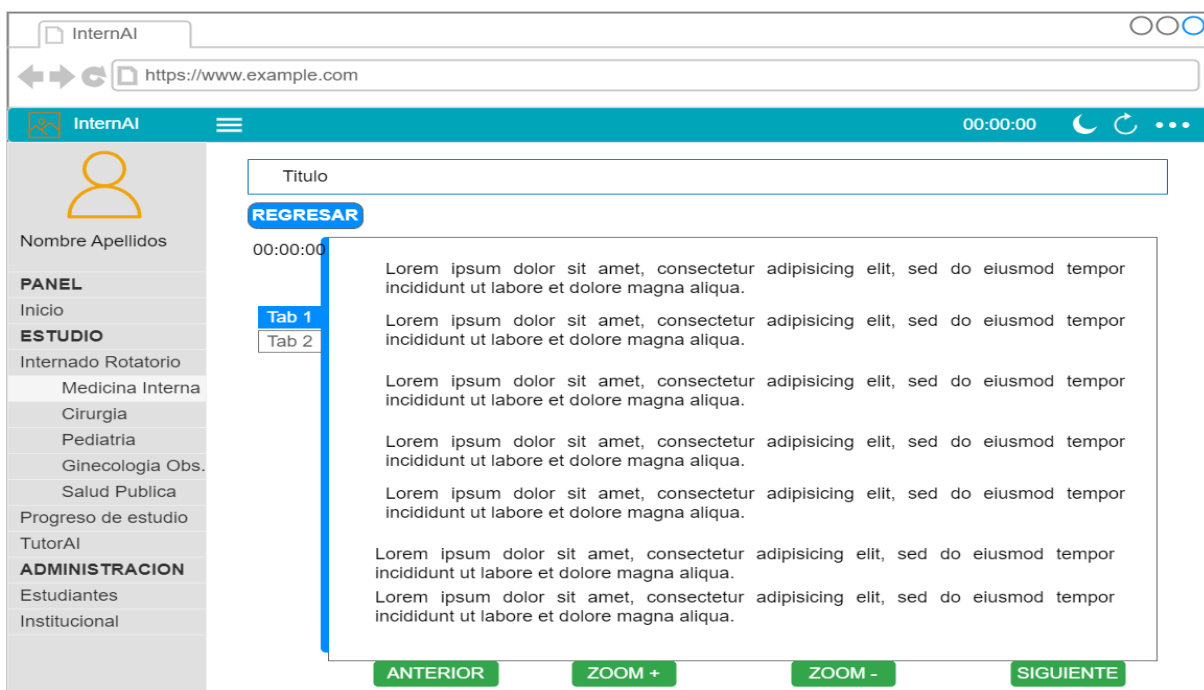


Figura 29

Prototipo, Vista progreso de estudio

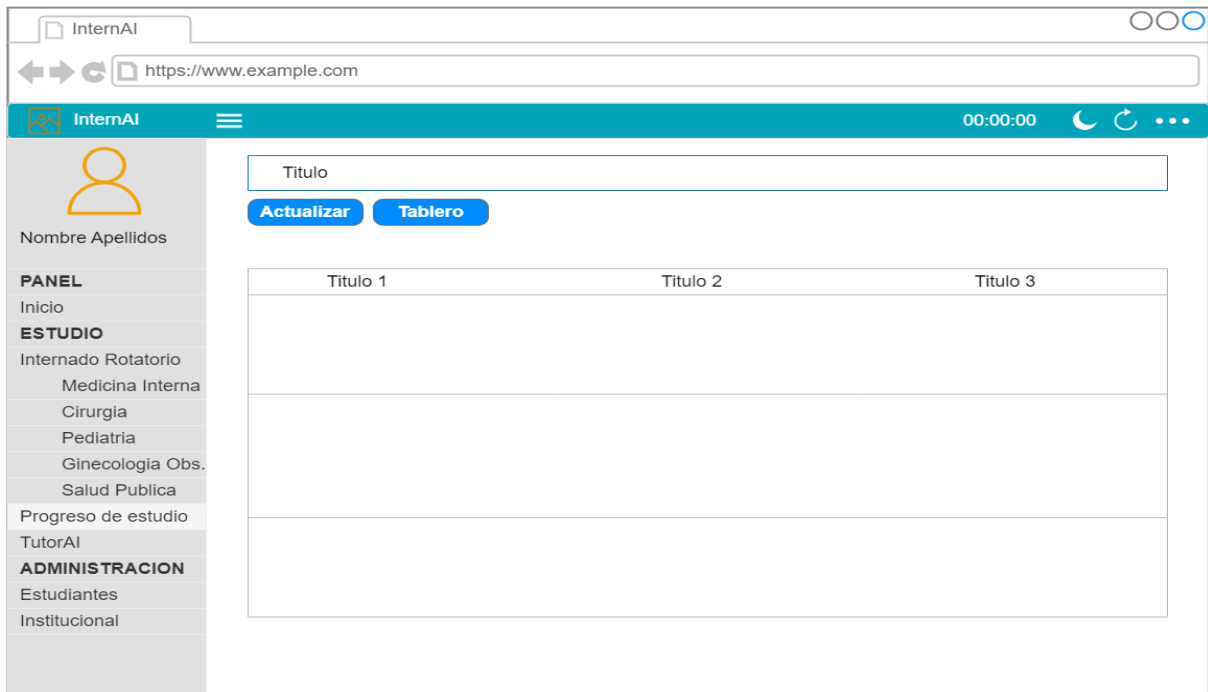


Figura 30

Prototipo, Vista TutorAI

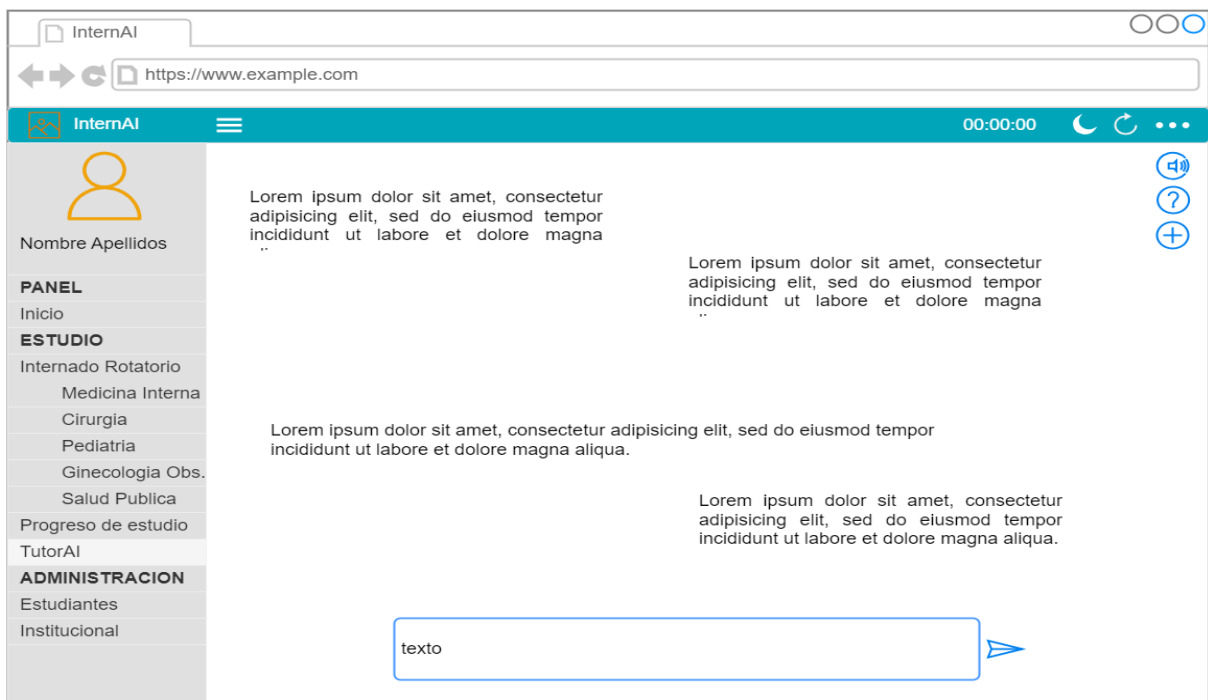


Figura 31

Prototipo, Vista estudiantes

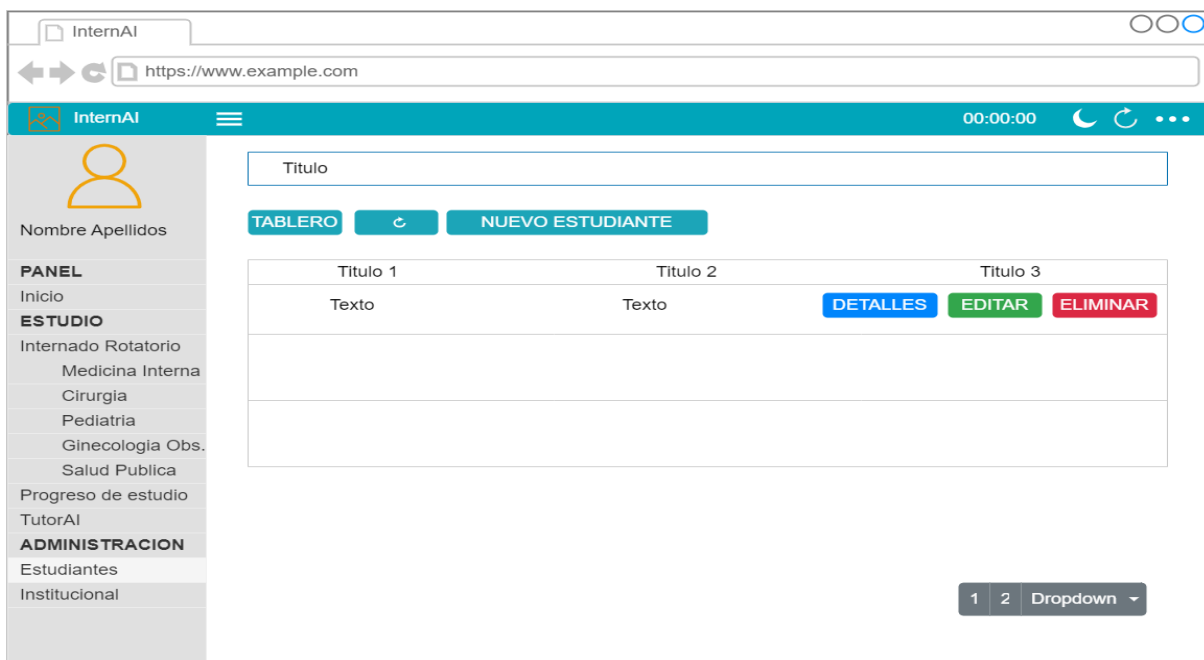


Figura 32

Prototipo, Vista formulario de estudiantes

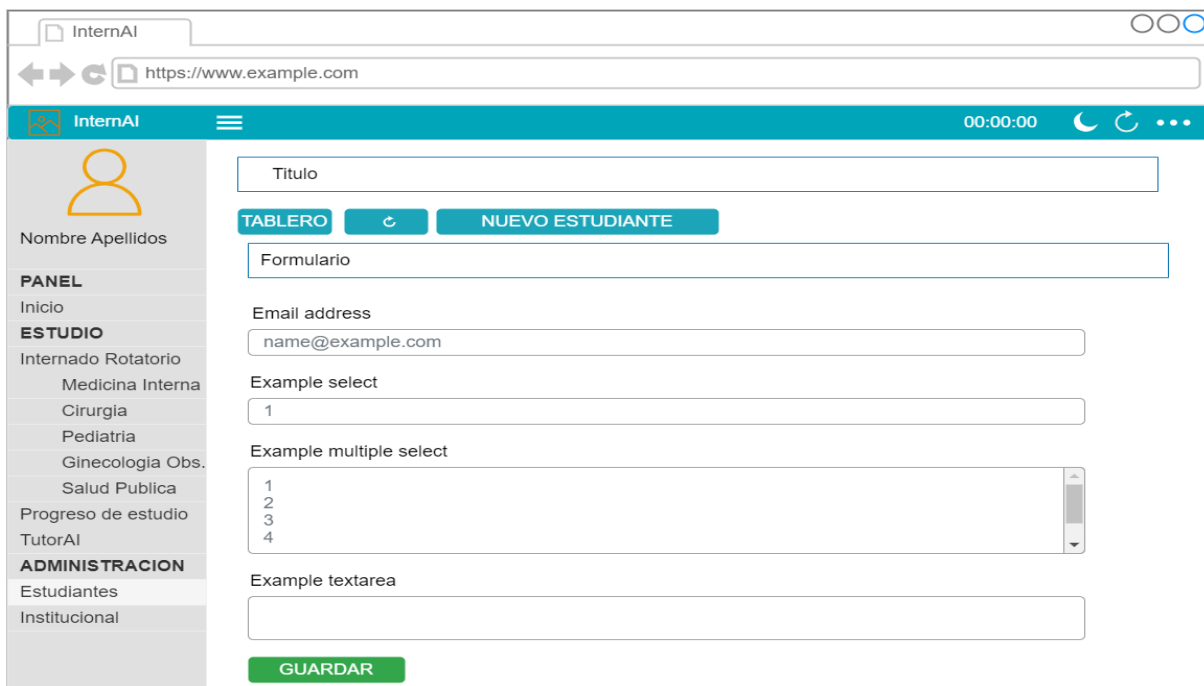


Figura 33

Prototipo, vista personal institucional

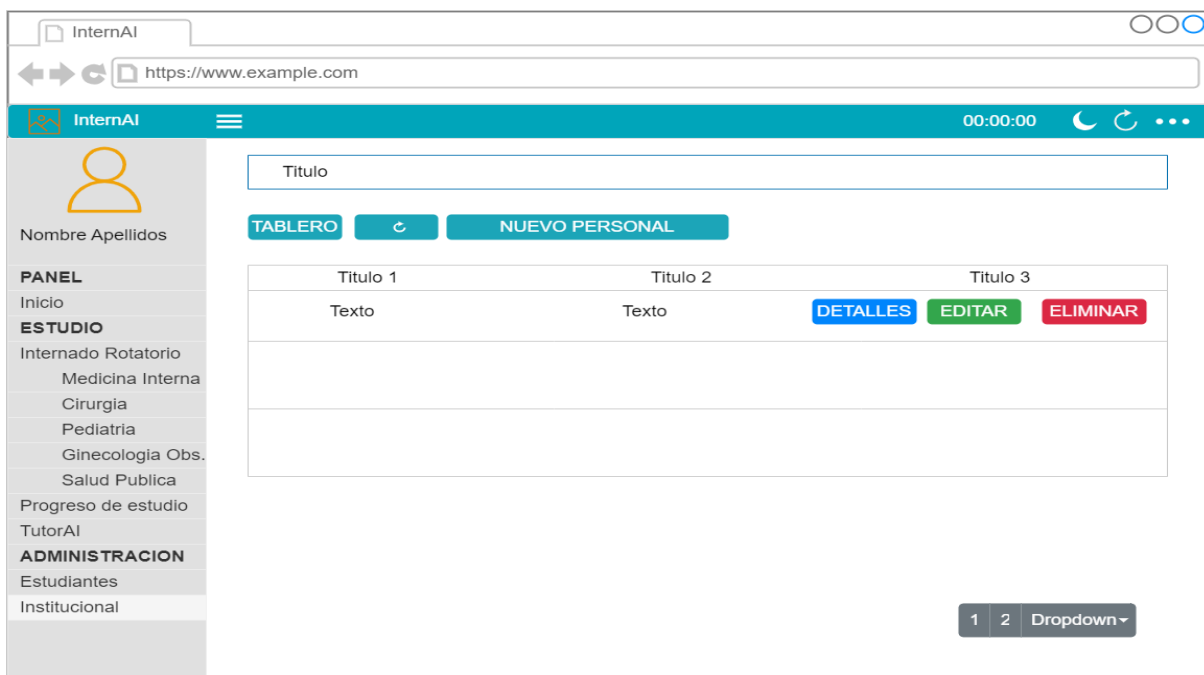
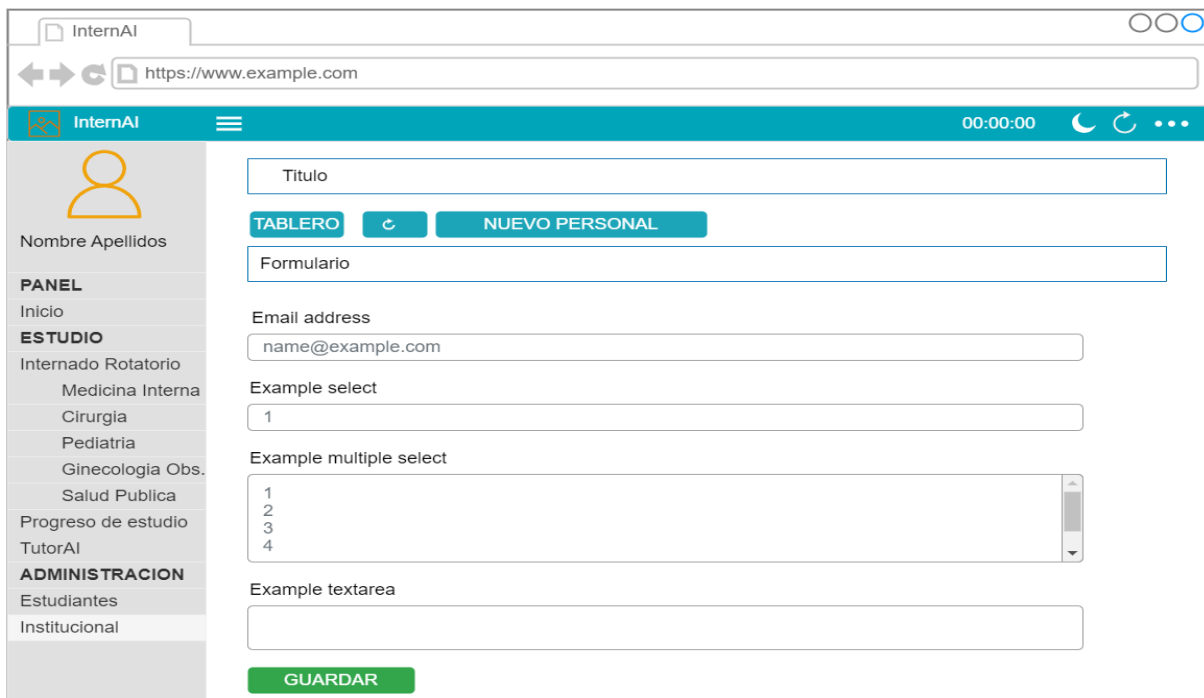


Figura 34

Prototipo, formulario personal institucional



3.3.4. *Recolección de Datos Iniciales*

Se recopilan los datos para el entrenamiento, principalmente en formato PDF de fuentes académicas y textos de cada área del internado, en colaboración del Dr. Luis Flores.

3.3.5. *Descripción de los Datos*

Se detallan la cantidad de datos, su estructura y su relevancia en temas específicos de medicina.

Tabla 16

Cantidad de datos, textos PDFs.

Salud publica	Pediatría	Medicina interna	Ginecología obstetricia	Cirugía
7	32	129	16	7
Total 191				

Nota. 191 textos en formato PDF del Internado Rotatorio.

3.3.6. *Exploración de los Datos*

Se realiza un análisis exploratorio para identificar patrones y estructuras en los datos. Esto incluye la verificación de que los datos están distribuidos uniformemente en cada tema del internado y que los textos tienen contenido útil para el modelo (como definiciones).

Se efectúan revisiones para asegurar la consistencia de los valores en los textos y la calidad general de los datos.

3.4. PLANEACIÓN DE ITERACIONES Y PREPARACIÓN DE DATOS

Se divide el proyecto en iteraciones, de forma que permita un desarrollo gradual, la cantidad de iteraciones dependerá si se cumplió con el desarrollo del producto en cada iteración. En cada iteración, se diseñan los módulos educativos, así como también los módulos administrativos.

Se estructura, formatea y limpia el conjunto de datos para el modelo, procesando los textos médicos para que estén en un formato adecuado para el entrenamiento.

3.4.1. Plan de Iteraciones

El proyecto se dividió en 7 iteraciones, el cual nos permite realizar un desarrollo gradual.

Figura 35

Plan de iteraciones

Iteraciones	Nombre	Objetivo	Tareas
Primera iteración	Módulo Inicio de sesión	Desarrollar la funcionalidad inicio de sesión y el perfil de usuario	Configuración inicial de sistema. Implementación de autenticación. Integración con la base de datos en PostgreSQL para el almacenamiento de datos de los usuarios. Crear la interfaz para visualización del perfil de usuario. Crear interfaz para actualización de credenciales.
Segunda iteración	Módulo estudiante	Desarrollar el módulo estudiante	Desarrollar frontend y backend para gestionar estudiantes.
Tercera iteración	Módulo institucional	Desarrollar el módulo institucional.	Desarrollar frontend y backend para gestionar personal.
Cuarta iteración	Módulo Áreas del Internado Rotatorio	Desarrollar el módulo Internado Rotatorio por área.	Desarrollo del frontend y backend. El usuario debe acceder fácilmente al contenido específico de cada área.
Quinta iteración	Sub Módulo Estudio	Desarrollar el sub modulo Estudio	Implementación de un sistema de carga y visualización de archivos PDF. Conexión del modelo LLM mediante APIs REST.

Iteraciones	Nombre	Objetivo	Tareas
			Generación y evaluación de un cuestionario según el tema por el modelo de red neuronal.
Sexta iteración	Módulo de Progreso del Estudiante	Crear una funcionalidad que registre y visualice el progreso del estudiante en cada área y tema de estudio.	Desarrollo de la interfaz y parte lógica (backend) donde el estudiante pueda ver sus avances en cada área, el número de cuestionarios completados y su rendimiento general. Almacenamiento y actualización de datos de progreso en la base de datos.
Séptima iteración	Módulo de TutorAI	Integrar el modelo LLM con el sistema para interactuar con el estudiante	Conexión del modelo LLM mediante APIs REST. Creación de una interfaz de usuario para enviar y recibir textos, mensajes en tiempo real.

Figura 36

Prioridades de las Iteraciones

Iteraciones	Nombre	Prioridad
Primera iteración	Módulo Inicio de sesión	Alta
Segunda iteración	Módulo de registro de estudiantes	Alta
Tercera iteración	Desarrollar el registro del personal institucional.	Alta

Iteraciones	Nombre	Prioridad
Cuarta iteración	Módulo Áreas del Internado Rotatorio	Alta
Quinta iteración	Sub Módulo Estudio	Alta
Sexta iteración	Módulo de Progreso del Estudiante	Alta
Séptima iteración	Módulo de TutorAI	Alta

3.4.2. Selección de los Datos

Se selecciona un subconjunto de los datos recopilados, basado en la relevancia y la calidad de los contenidos. Se eligen textos específicos de cada tema para asegurarse de que el modelo aprenda de manera balanceada y esté bien preparado para generar contenido en todas las áreas.

3.4.3. Limpieza de los Datos

Se implementan diversas técnicas de limpieza de datos, enfocadas en garantizar la calidad y consistencia del contenido extraído. Estas técnicas incluyen la eliminación de ruido, como caracteres irrelevantes, metadatos innecesarios, espacios en blanco redundantes, líneas vacías y elementos no textuales, como imágenes o tablas que no son procesables en formato textual.

Posteriormente, se realiza un proceso de transformación y estandarización de los datos, asegurando que toda la información sea organizada de manera uniforme.

Figura 37

Función para limpieza de datos

```

# Función para limpiar y fragmentar texto
def limpiar_texto(texto):
    texto = texto.replace('\n', ' ').replace('\xa0', ' ') # Limpiar saltos de línea y caracteres no deseados
    texto = re.sub(r"[\d+]", "", texto) # Eliminar corchetes
    texto = re.sub(r"[^A-Za-zÁÉÍÓÚáéíóúñÑÚ\s]", "", texto) # Elimina caracteres no alfabéticos
    texto = re.sub(r"\s+", " ", texto).strip() # Elimina espacios adicionales
    # Eliminar números de línea al inicio de cada línea
    texto = re.sub(r"^\d+\s*", "", texto, flags=re.MULTILINE)
    # Eliminar tablas representadas por líneas de números
    texto = re.sub(r"\s*\d+(\s+\d+)+\s*", "", texto)
    # Eliminar números de página que están al final
    texto = re.sub(r"\s*\d+\s*$", "", texto)
    # Eliminar títulos de capítulos como "Capítulo 1", "Tema", etc.
    texto = re.sub(r"^(Capítulo \d+|Tema \d+|Sección \d+|Unidad \d+).*", "", texto, flags=re.MULTILINE)
    # Eliminar caracteres no imprimibles
    texto = re.sub(r"[\x00-\x1F\x7F]", "", texto)
    # Eliminar signos de puntuación innecesarios (puedes ajustar la lista según necesites)
    texto = re.sub(r"[.,;!?(){}[\]<>]", "", texto)
    # Eliminar texto de números de página (Ej: Página 1, Page 1, etc.)
    texto = re.sub(r"Página \d+|Page \d+", "", texto)

    # Corrige palabras mal escritas
    palabras = texto.split()
    palabras_corregidas = []

    for palabra in palabras:
        # Corregir la palabra solo si tiene una corrección válida
        corregida = spell.correction(palabra)
        if corregida: # Solo agregar la corrección si es válida
            palabras_corregidas.append(corregida)
        else:
            palabras_corregidas.append(palabra)

    texto_corregido = " ".join(palabras_corregidas)

    return texto_corregido

```

Nota. La función "limpiar_texto", elimina todo el ruido de los textos PDFs

3.4.4. Integración de los Datos

Se integran los diferentes documentos y textos seleccionados en un conjunto de datos único que abarca todos los temas del internado rotatorio.

3.4.5. Formateo de los Datos

Los datos se transforman al formato requerido por el proceso de entrenamiento de la red neuronal. Esto implica adaptar el texto para que el modelo pueda comprender las entradas y generar respuestas coherentes. Se realizan ajustes en el orden y sintaxis para optimizar el uso de herramientas de procesamiento, eliminando caracteres especiales o reordenando campos según los requisitos del modelo.

Figura 38

Formateo de los datos

```

# Procesar PDFs y crear datos estructurados
base_dir = '/content/drive/MyDrive/datos-internado-rotatorio/'
areas = ['medicina-interna', 'cirugia', 'ginecologia-obstetricia', 'salud-publica', 'pediatria']
data = []

for area in areas:
    area_dir = os.path.join(base_dir, area)
    for nombre_pdf in os.listdir(area_dir):
        ruta_pdf = os.path.join(area_dir, nombre_pdf)
        if ruta_pdf.endswith('.pdf'):
            with fitz.open(ruta_pdf) as pdf:
                texto_total = ""
                for pagina in pdf:
                    texto_total += pagina.get_text()
                texto_total = limpiar_texto(texto_total)
                fragmentos = fragmentar_texto(texto_total)

                for idx, fragmento in enumerate(fragmentos):
                    data.append({
                        "area": area,
                        "contenido": fragmento
                    })

# Crear el Dataset
dataset = Dataset.from_list(data)
dataset = dataset.train_test_split(test_size=0.1) # Dividir en entrenamiento y prueba

dataset.push_to_hub("dataset_internado_rotatorio") # Guarda el dataset en el repositorio

```

Figura 39

DataSet







area	nombre_pdf	contenido
string · classes 5 values	string · classes 190 values	string · lengths 8 5.16k
pediatria	AIEPI - Intervenciones basadas en evidencia 2ed.pdf	Geneva: pregnancy after antepartum stillbirth. A review. J Matern WHO Press. 2006. Fetal Neonatal...
medicina-interna	Enfermedades_glomerulares.pdf	responsable por emesis, el aumento de la excreción (quelantes o diuréticos) mientras persista la...
pediatria	casos_clinicos_de_patologia_infecciosa_pediatica_en_la_ucip.pdf	punción, asepsia total con método propofol se debe recambiar cada 6- dos de barrera completos. 12...
pediatria	El Nino Sordo - David M. Lutzerman.pdf	1992. Promoting emerging speech in birth to 3 year-old hear- Brown, A. S. and Yoshinaga-Itano, C. 1994...
ginecologia-obstetricia	Manual_obstetricia_ginecologia.pdf	7, luego comparte esta función con la placenta y comienza a declinar en la semana 8-10. Ambos ovario...
pediatria	Protocolo de Urgencias en Pediatrã-a.pdf	(sin - Infecciones complejos antígeno-anticuerpo) - Enf. vasculares del colágeno - Angioedema...

Figura 40

Verificar y cargar el DataSet

```
[3] from datasets import load_dataset

# Cargar el dataset desde el repositorio
dataset = load_dataset("DevCar/dataset_internado_rotatorio_v2")
```

 /usr/local/lib/python3.10/dist-packages/huggingface_hub/utils/_auth.py:94: UserWarning:
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your settings tab (<https://huggingface.co/settings/tokens>).
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access public models or datasets.
warnings.warn(
 warnings.warn(
 README.md: 100%  476/476 [00:00<00:00, 14.6kB/s]
 train-00000-of-00001.parquet: 100%  15.4M/15.4M [00:00<00:00, 23.1MB/s]
 test-00000-of-00001.parquet: 100%  1.71M/1.71M [00:00<00:00, 17.9MB/s]
 Generating train split: 100%  13089/13089 [00:00<00:00, 23065.76 examples/s]
 Generating test split: 100%  1455/1455 [00:00<00:00, 11925.44 examples/s]

```
import pandas as pd

df_train = pd.DataFrame(dataset["train"])

df_train.head(10)
```

	area	nombre_pdf	contenido
0	pediatria	AIEPI - Intervenciones basadas en evidencia 2e...	Geneva: pregnancy after antepartum stillbirth....
1	medicina-interna	Enfermedades_glomerulares.pdf	responsable por emesis, el aumento de la excre...
2	pediatria	casos_clinicos_de_patologia_infecciosa_pediatr...	punción, asepsia total con méto- do propofol s...
3	pediatria	El Nino Sordo - David M. Luterman.pdf	1992. Promoting emerging speech in birth to 3 ...
4	ginecologia-obstetricia	Manual_obstetricia_ginecologia.pdf	7, luego comparte esta función con la placenta...

3.5. DISEÑO COMPUTACIONAL Y MODELADO

En esta etapa, se elabora el diseño computacional de cada iteración. Se crean y configuran el frontend y backend, asegurando que cada parte del sistema cumpla con los requerimientos funcionales establecidos en las etapas anteriores.

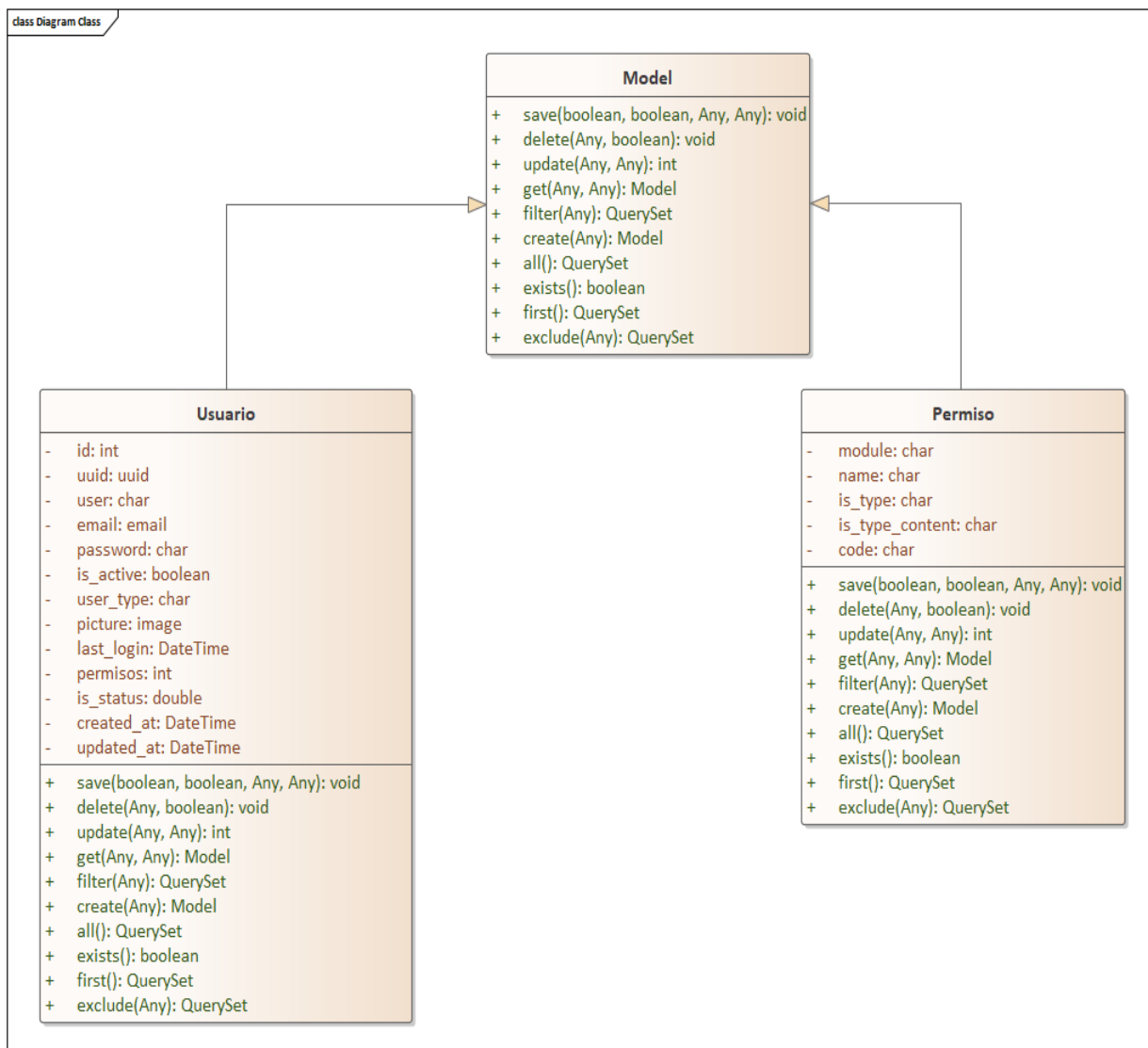
Siguiendo la fase de modelado de CRISP-DM, se establece los parámetros del modelo y posteriormente se realiza el entrenamiento del modelo LLM utilizando técnicas de peft y QLoRA. Se entrena iterativamente el modelo, asegurándose de que el LLM esté optimizado para generar texto en el ámbito del área de medicina, así como también calificar preguntas adecuadas al nivel de conocimiento del internado. Las pruebas iniciales de rendimiento ayudan a ajustar los parámetros del modelo y asegurar una alta precisión.

3.5.1. Primera Iteración

En esta primera iteración, se creó un diagrama de clases para representar la estructura de las clases Usuario y Permisos, los cuales heredan de la clase padre Model.

Figura 41

Diagrama de clases primera iteración



Se desarrolló un diagrama de secuencias que describe el proceso de autenticación, desde la entrada de credenciales hasta la verificación en la base de datos PostgreSQL.

Figura 42

Diagrama de secuencia primera iteración

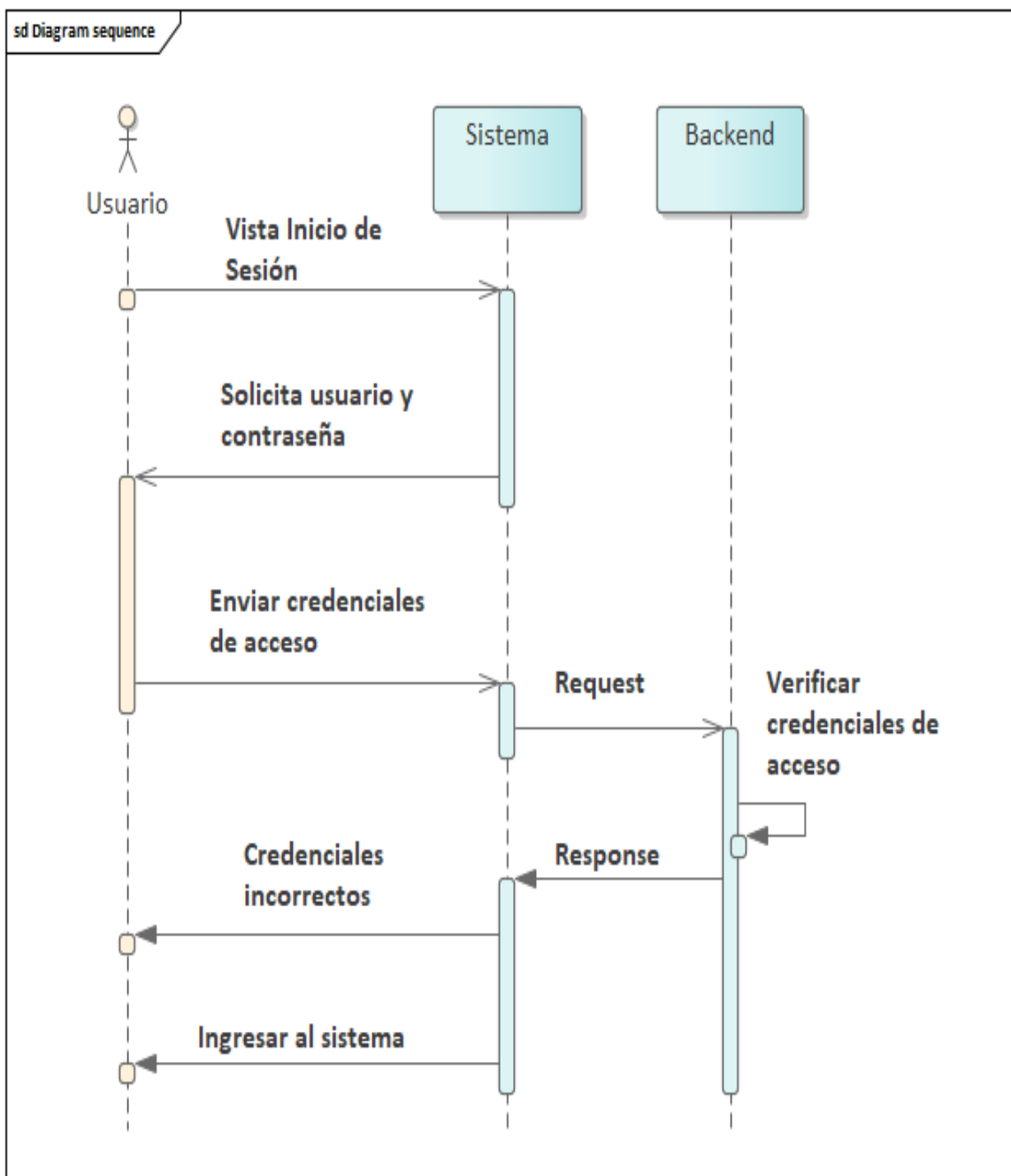
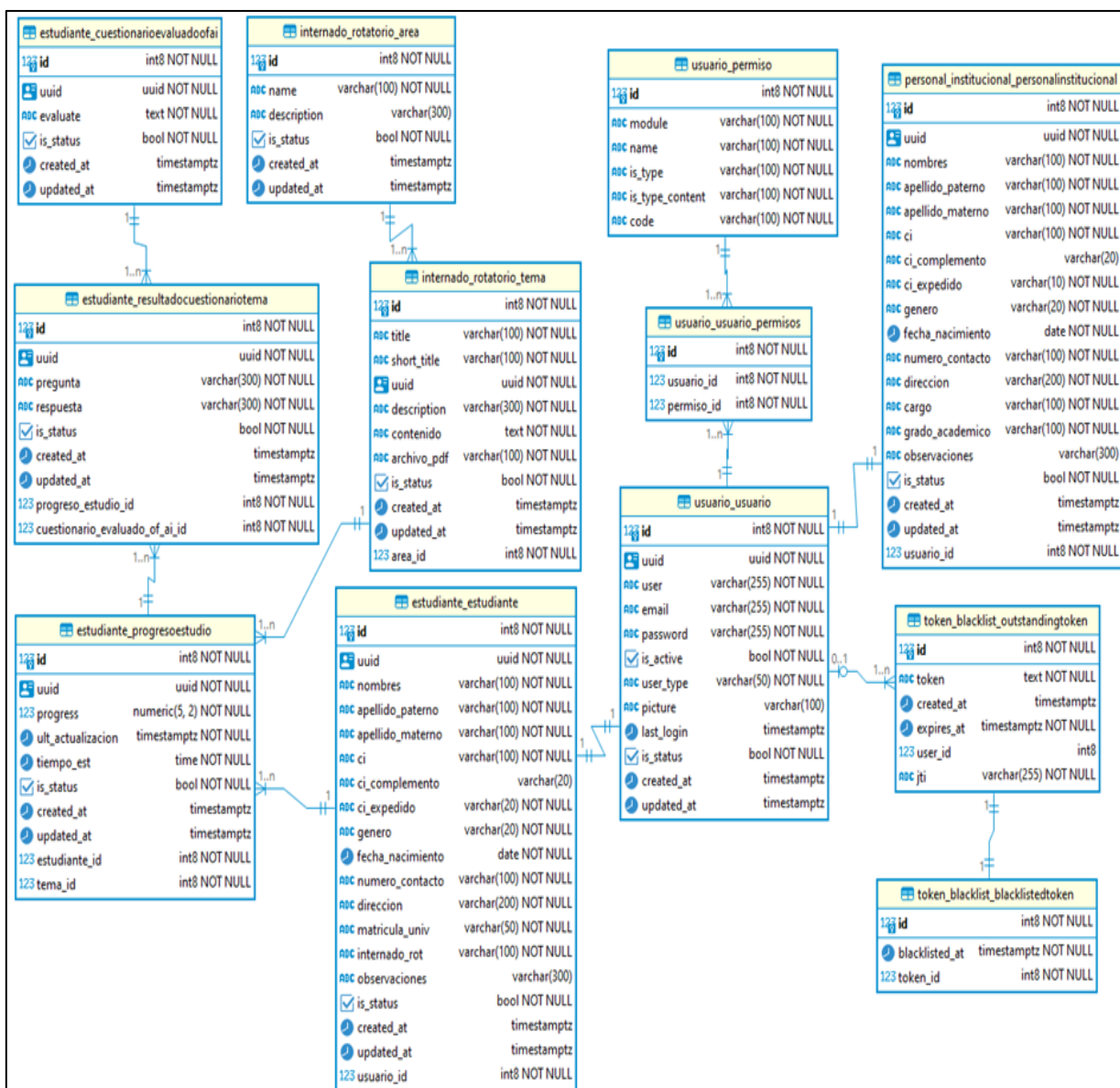


Figura 43

Diagrama conceptual de la base de datos

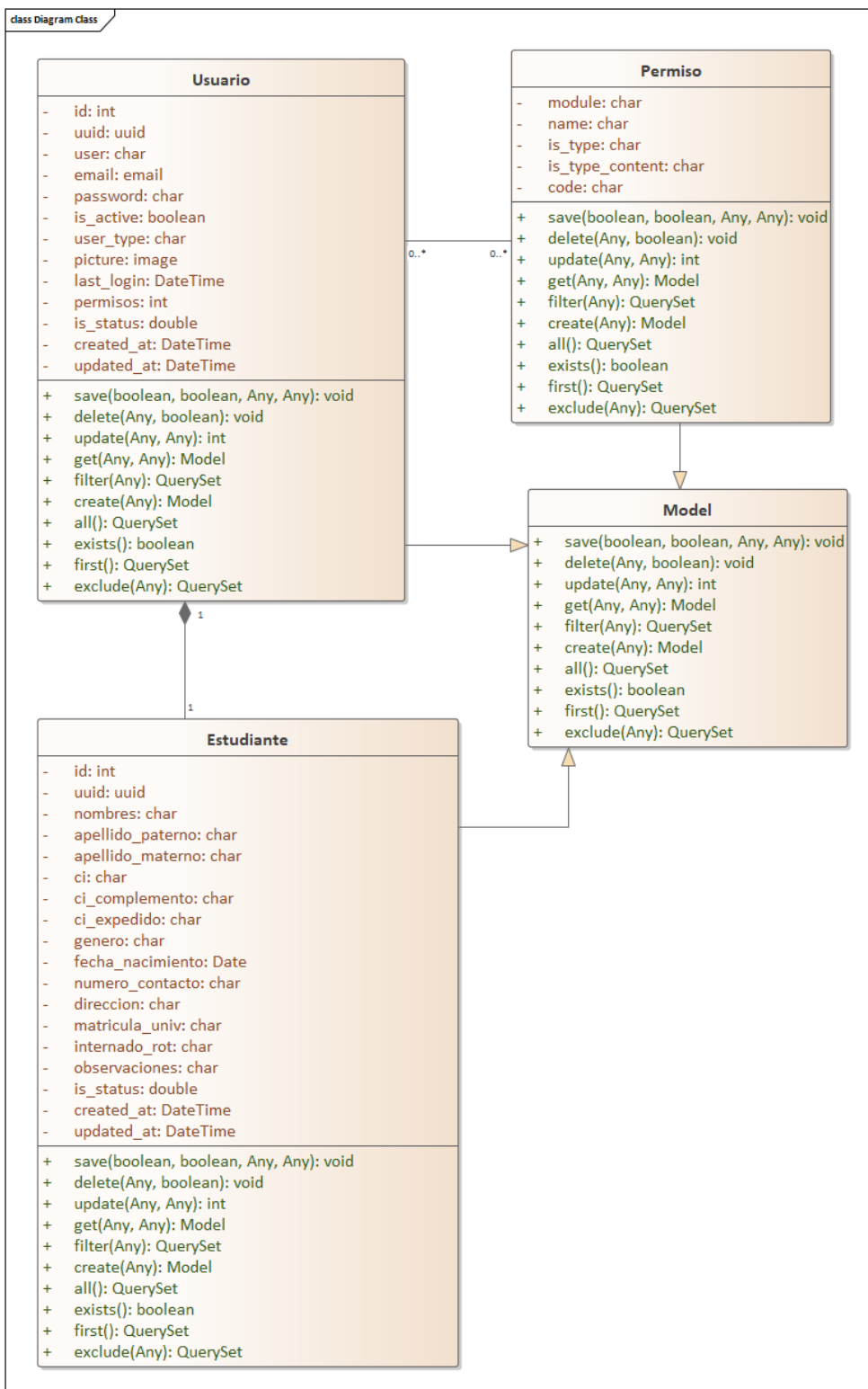


3.5.2. Segunda Iteración

En el diagrama de clases de la primera iteración se agrega la clase Estudiante, con sus atributos y métodos, estos métodos heredan de la clase padre Model. Así como también tiene una relación con la clase Usuario.

Figura 44

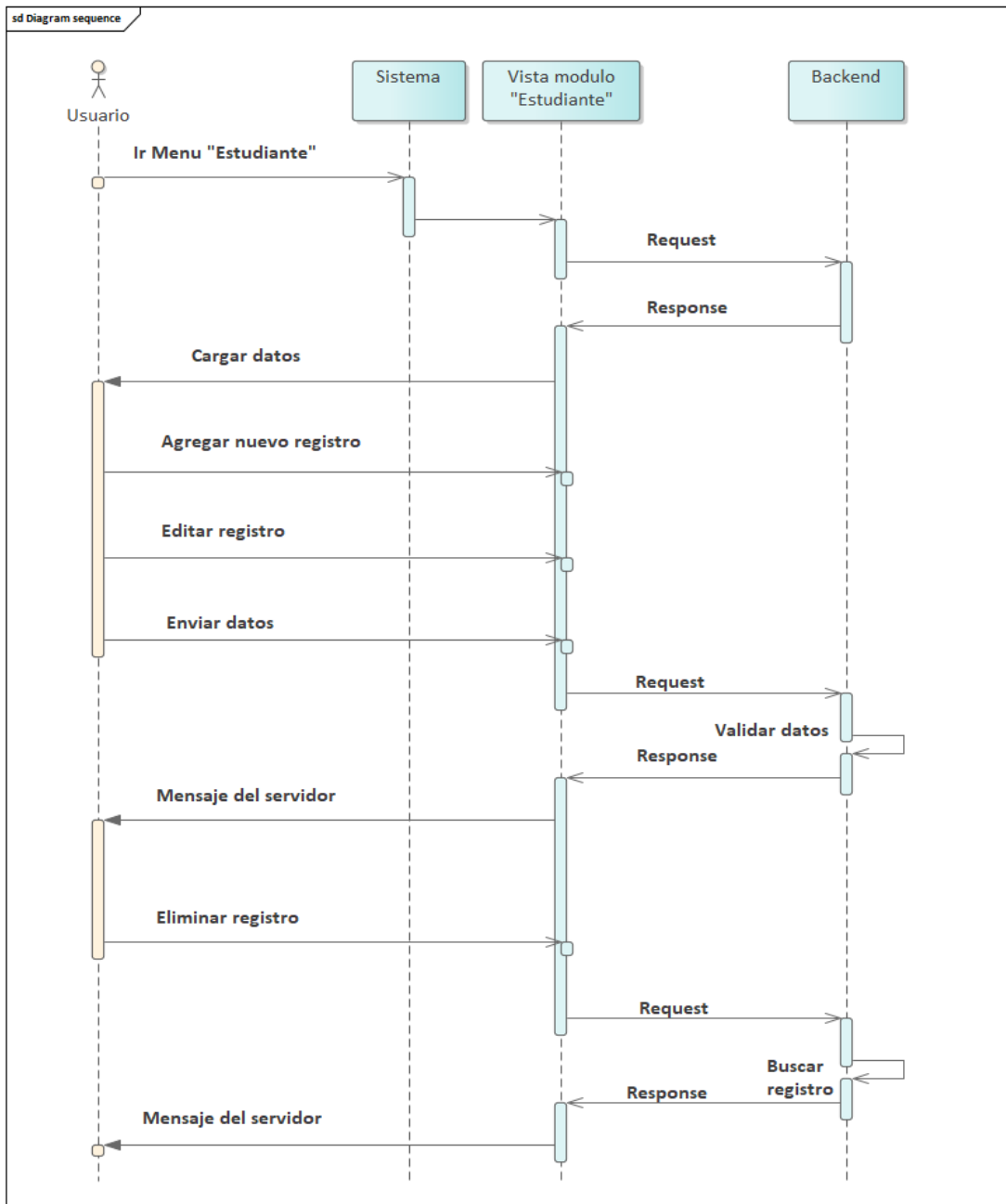
Diagrama de clases segunda iteración



Se creó un diagrama de secuencias para representar la interacción entre el frontend y el backend en la gestión de estudiantes, mostrando procesos como la creación, edición y la eliminación de registros.

Figura 45

Diagrama de secuencia, segunda iteración

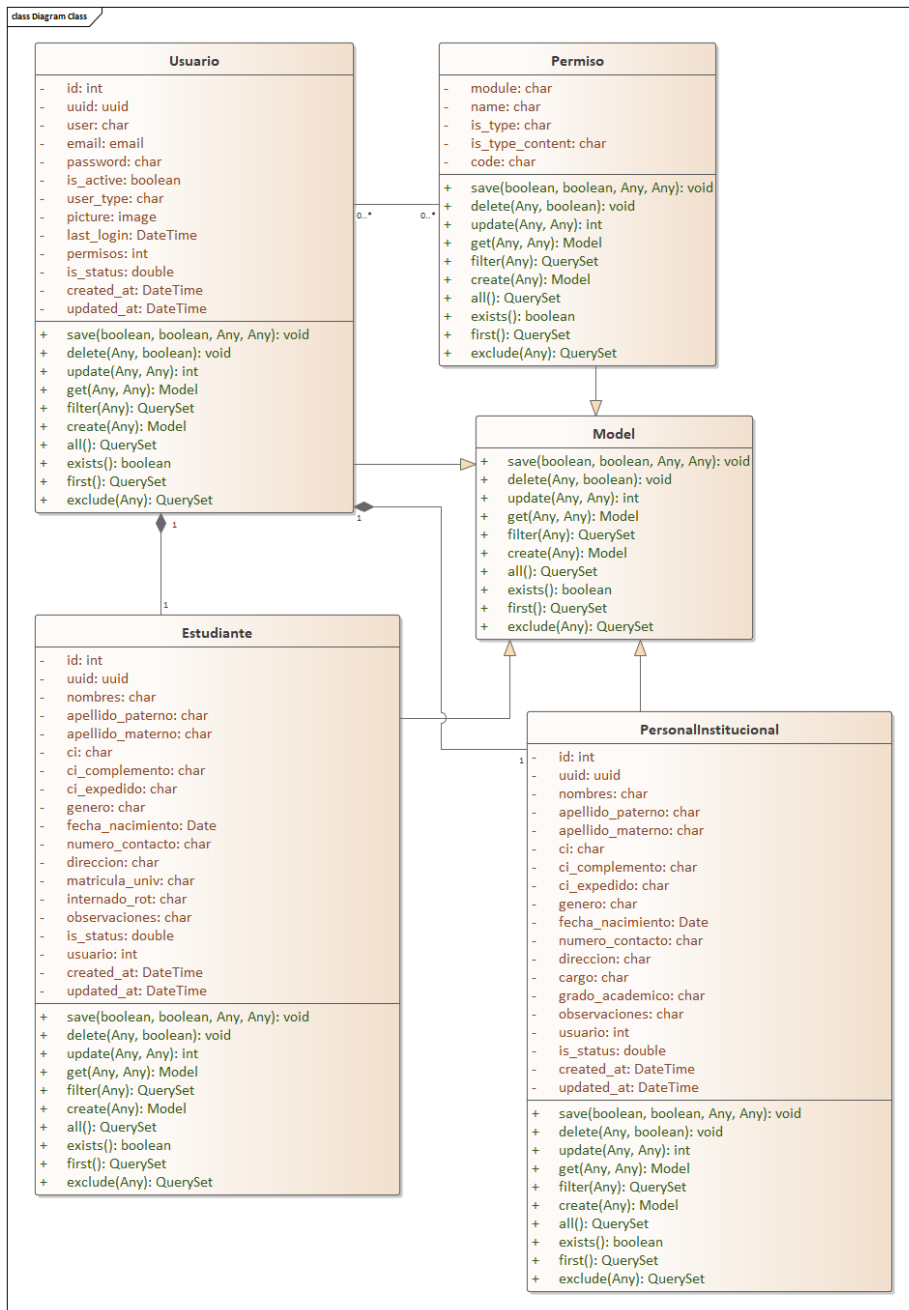


3.5.3. Tercera Iteración

En el diagrama de clases se detalla la clase PersonalInstitucional, modelando la gestión del personal y sus roles (Doctor o Administrador) dentro de la institución.

Figura 46

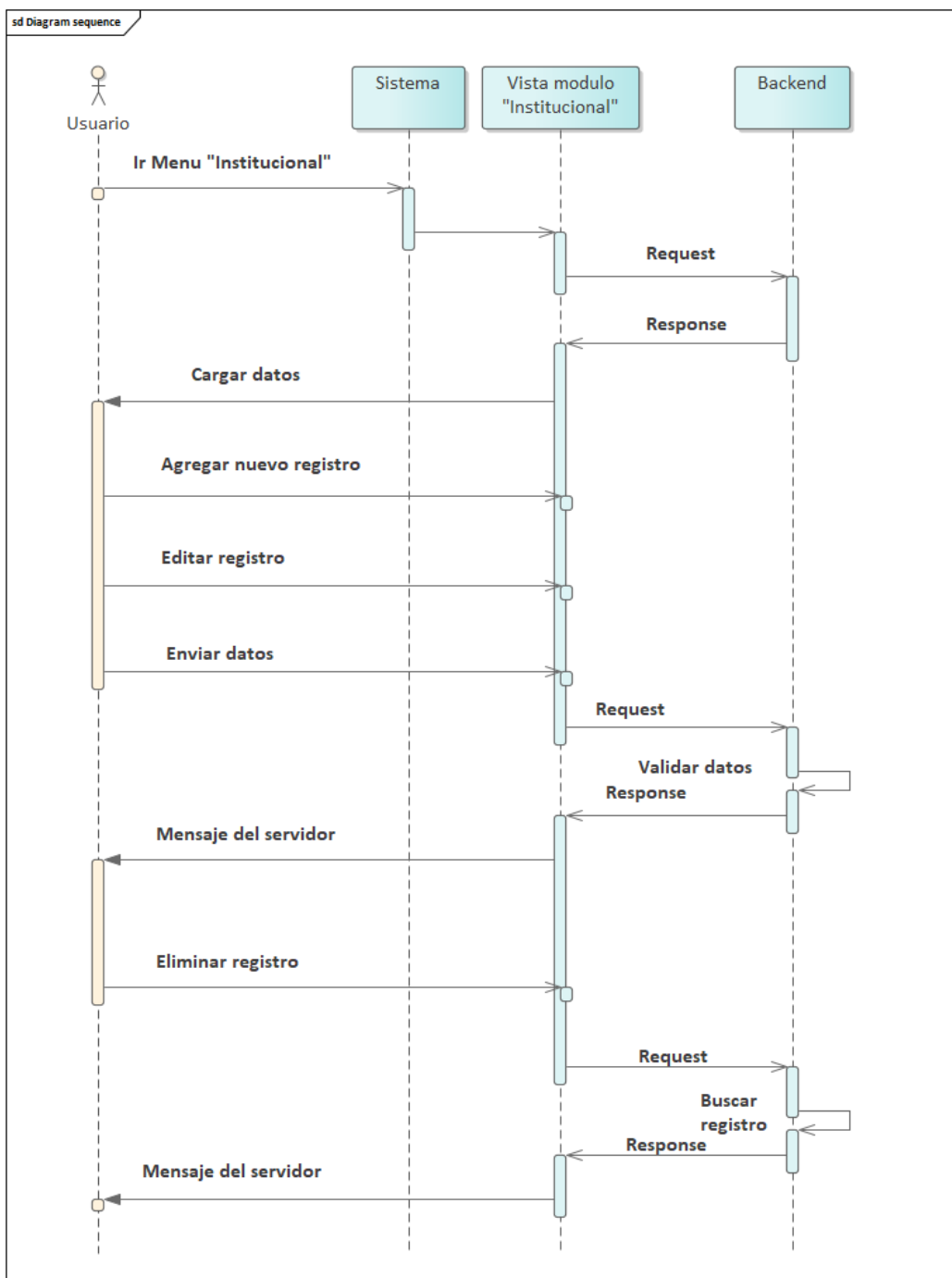
Diagrama de clases, tercera iteración



En el diagrama de secuencias se ilustra los procesos de lectura, creación, edición y eliminación del personal institucional, desde el frontend hasta las peticiones http al backend.

Figura 47

Diagrama de secuencia, tercera iteración



3.5.4. Cuarta Iteración

Se creó un diagrama de clases que representaba las entidades Área y Tema, el cual nos permite organizar el material de estudio del Internado Rotatorio.

Figura 48

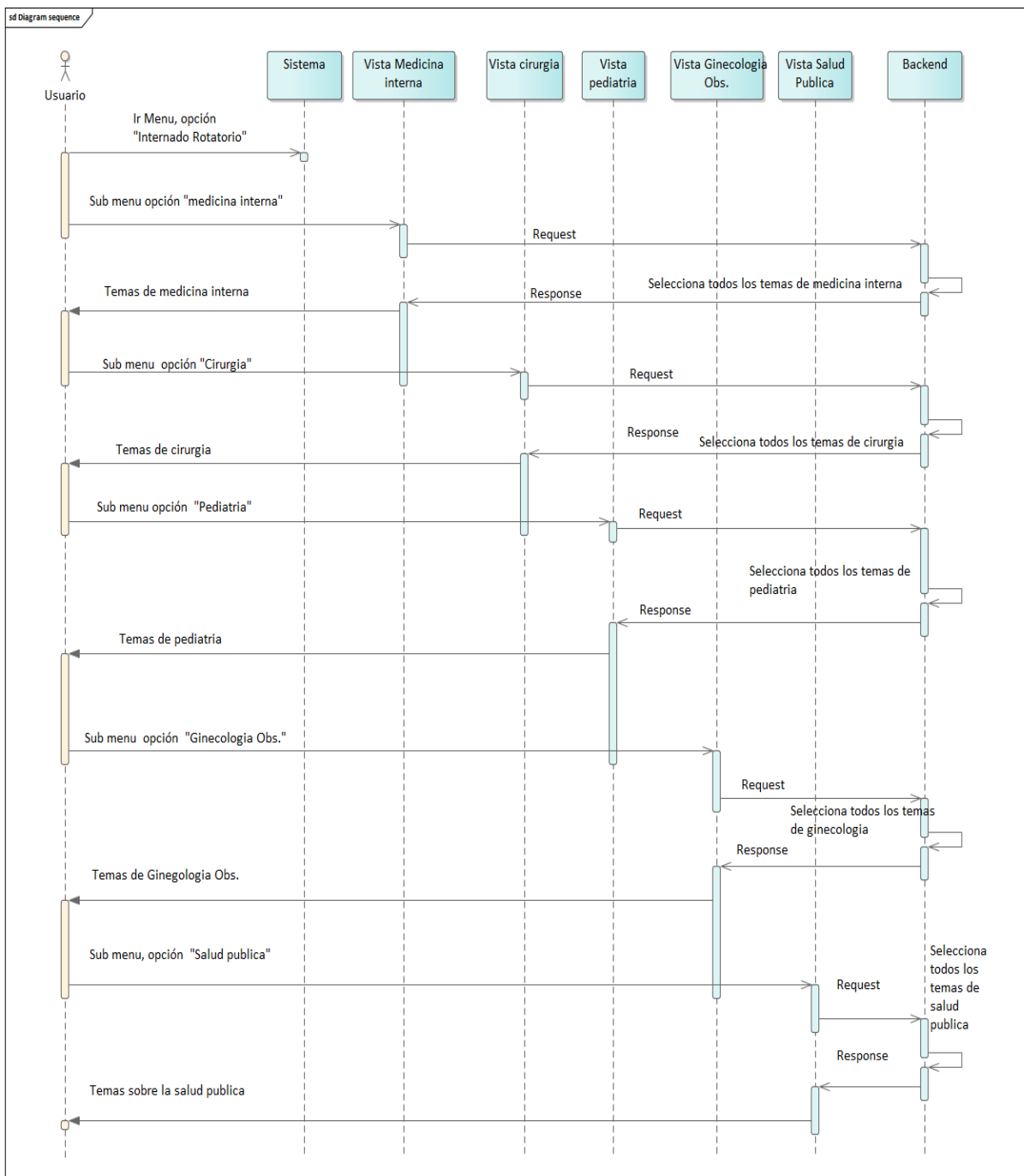
Diagrama de clases, cuarta iteración



El diagrama de secuencias muestra el flujo de acceso al contenido específico de cada área, desde la selección del área del Internado Rotatorio hasta la visualización de los temas.

Figura 49

Diagrama de clases cuarta iteración

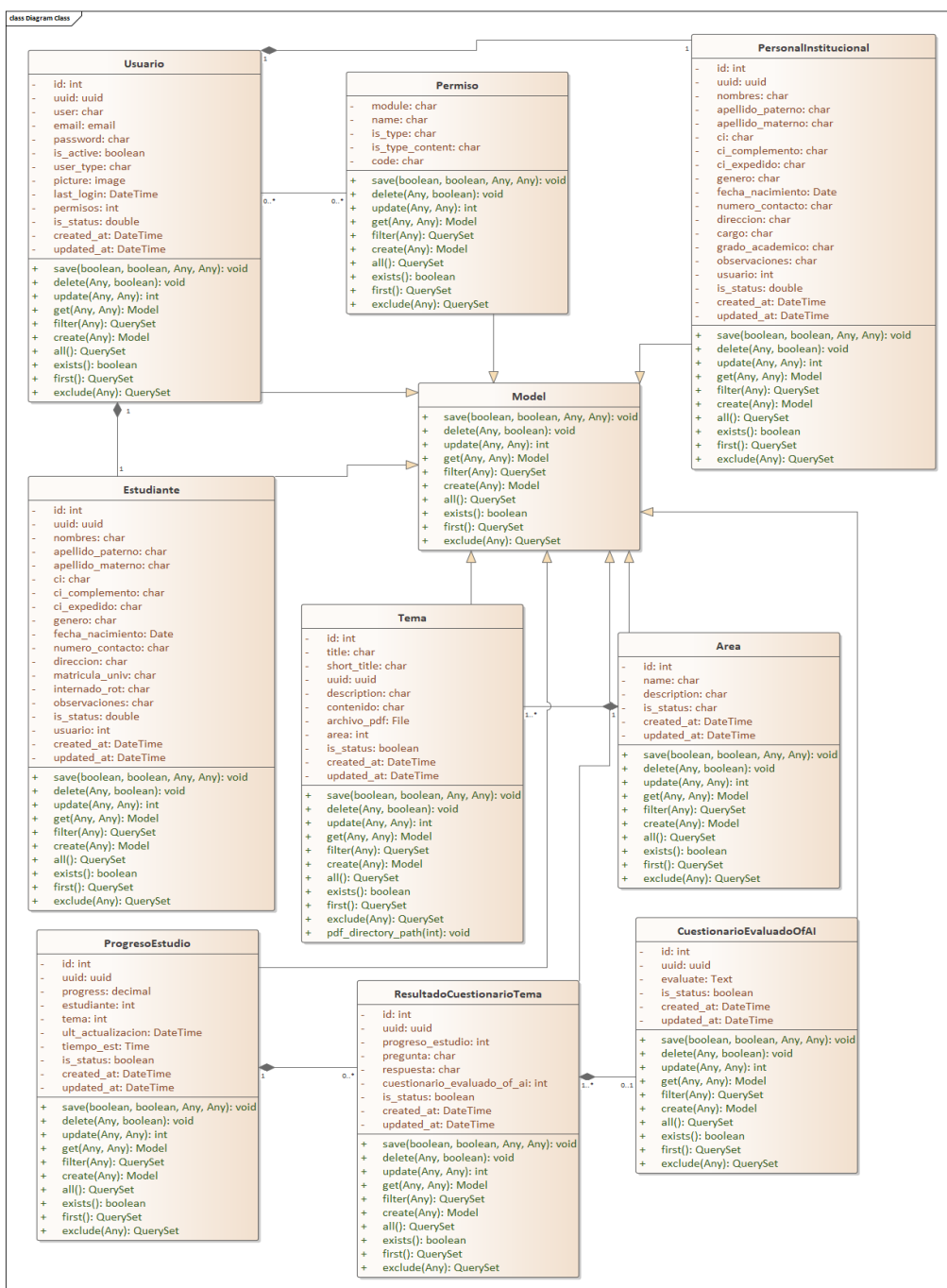


3.5.5. Quinta Iteración

En el diagrama de clases se incluye las clases ProgresoEstudio, ResultadoCuestionarioTema y CuestionarioEvaluadoOfAI.

Figura 50

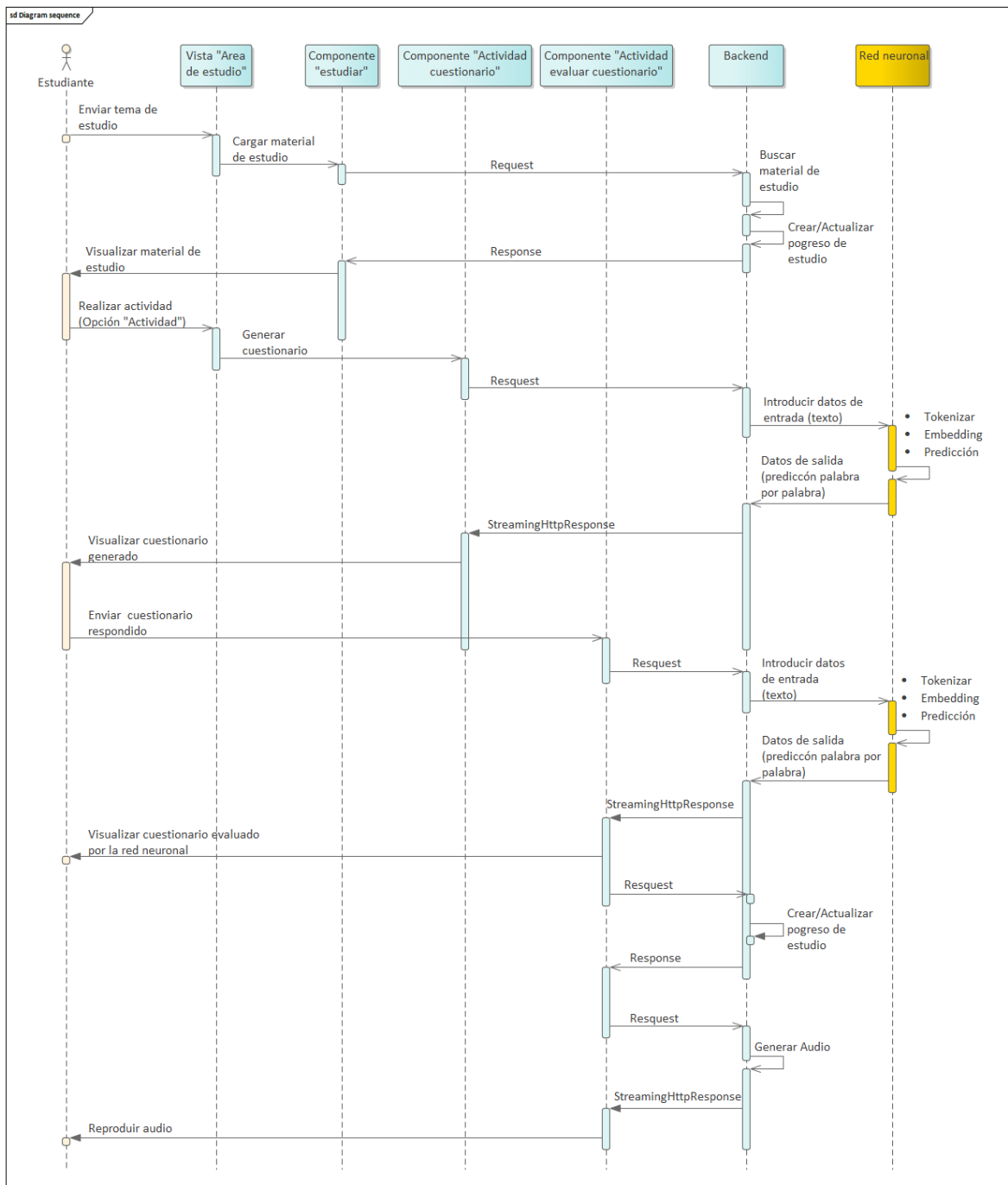
Diagrama de clases quinta iteración



El diagrama de secuencias detalla el proceso de carga del material de estudio e ilustra el uso de la red neuronal.

Figura 51

Diagrama de secuencia quinta iteración

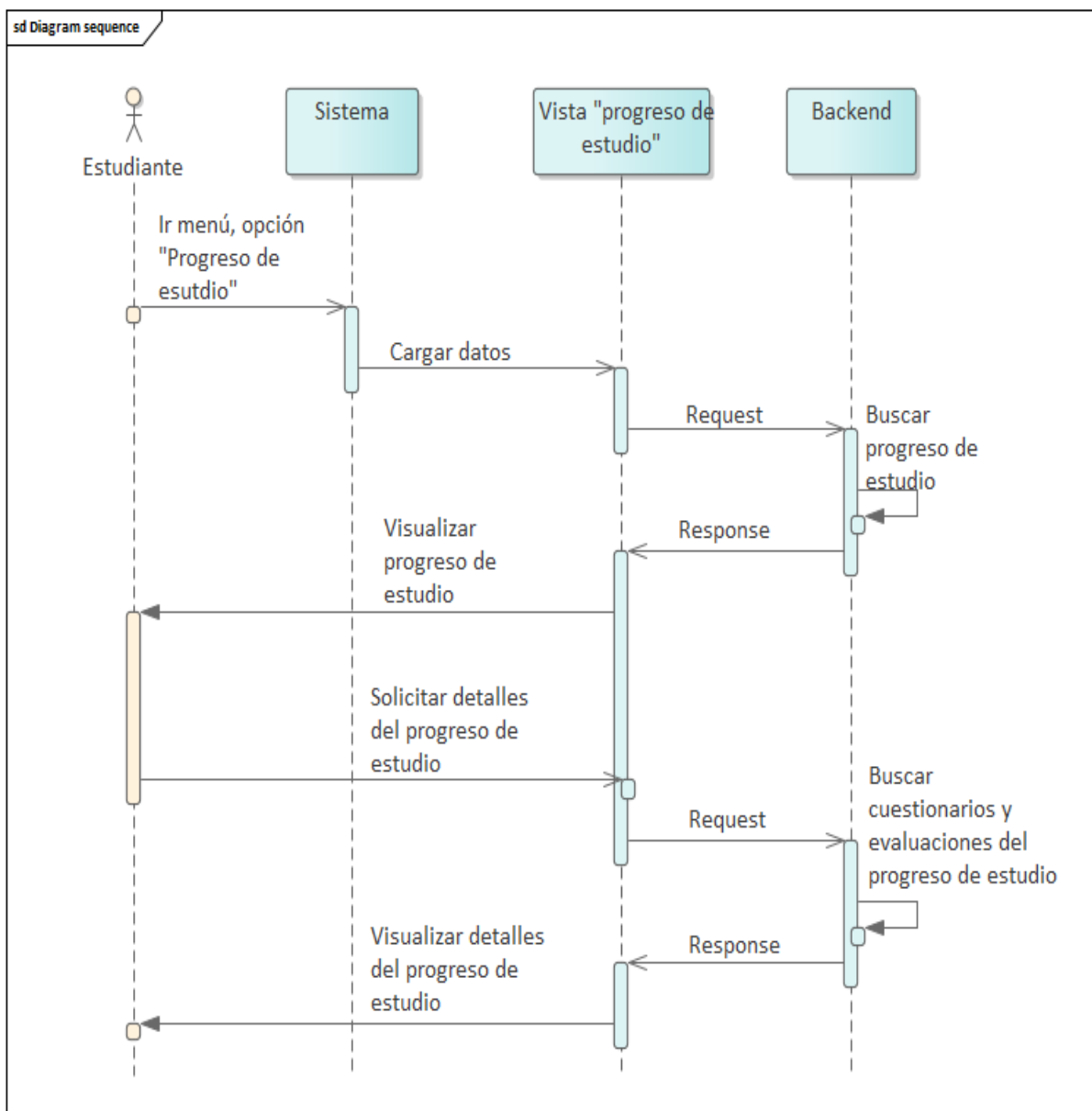


3.5.6. Sexta Iteración

En la quinta iteración ya se realizó el diseño del diagrama de clases del progreso de estudio, por lo tanto, ya no es necesario realizarlo. El diagrama de secuencias ilustra el proceso de visualizar el "Progreso de estudio".

Figura 52

Diagrama de secuencia sexta iteración

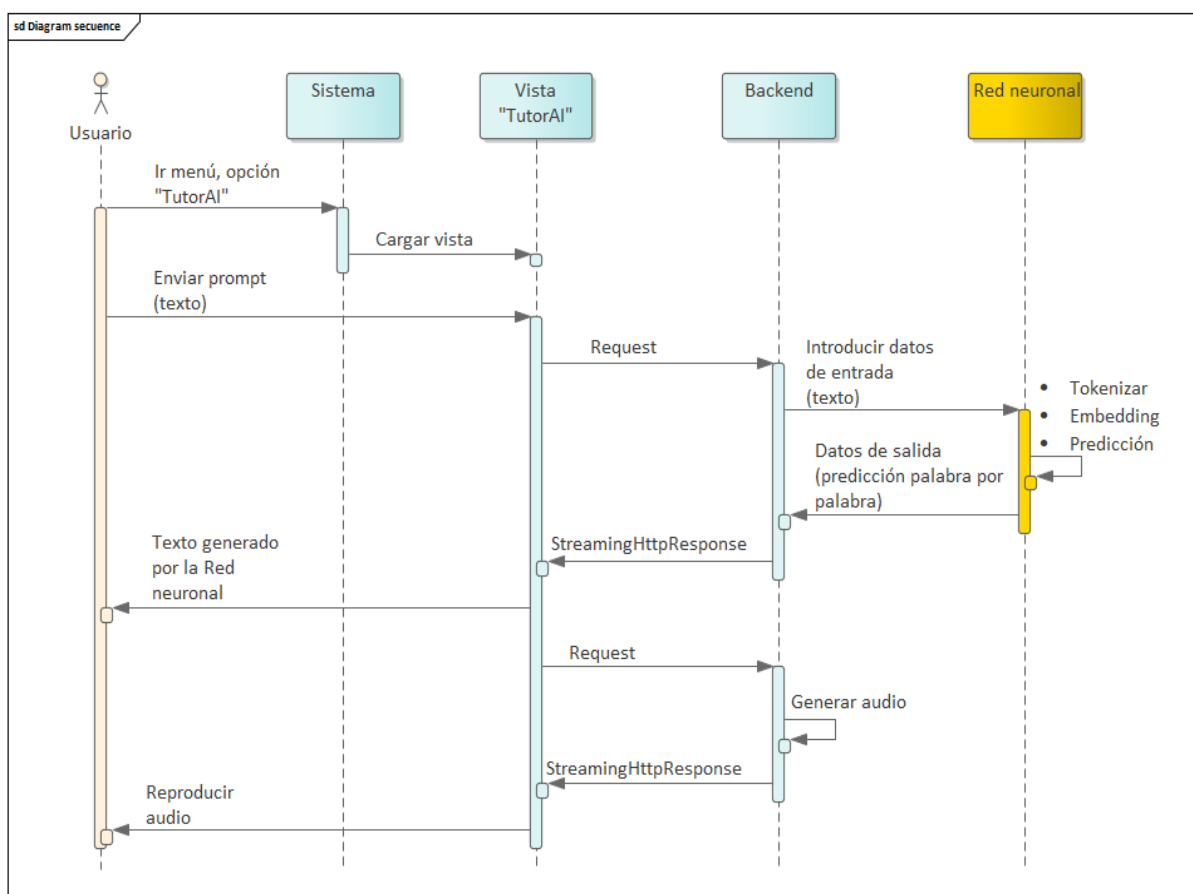


3.5.7. Séptima Iteración

En esta última iteración no fue necesario realizar el diagrama de clases, porque en la quinta iteración se realizó la versión final del diagrama de clases. Por lo tanto, solo se realiza el diagrama de secuencias, el cual ilustra la Interacción con TutorAI (Red neuronal).

Figura 53

Diagrama de secuencia séptima iteración



3.5.8. Selección de la Técnica de Modelado

Se selecciona la técnica de entrenamiento (QLoRA) para el modelo, en función de los requisitos de precisión y el tipo de contenido que se necesita generar. QLoRA permite optimizar el ajuste fino en un entorno con recursos limitados, asegurando que el modelo se adapte de forma eficiente al contexto médico.

3.5.9. Construcción del Modelo

Se configura el área de trabajo de Google Colab Pro, paralelamente se ajustan los parámetros iniciales para realizar el entrenamiento del modelo usando QLoRA, para no perder la precisión de la red neuronal. Durante el proceso, se ajustan los parámetros iterativamente para mejorar la precisión de las respuestas y la calidad de las preguntas generadas, interpretando continuamente los resultados para refinar el modelo. Por último, se realiza el entrenamiento de la red neuronal utilizando la versión paga de Google Colab.

Figura 54

Simulación de la red neuronal de forma tradicional

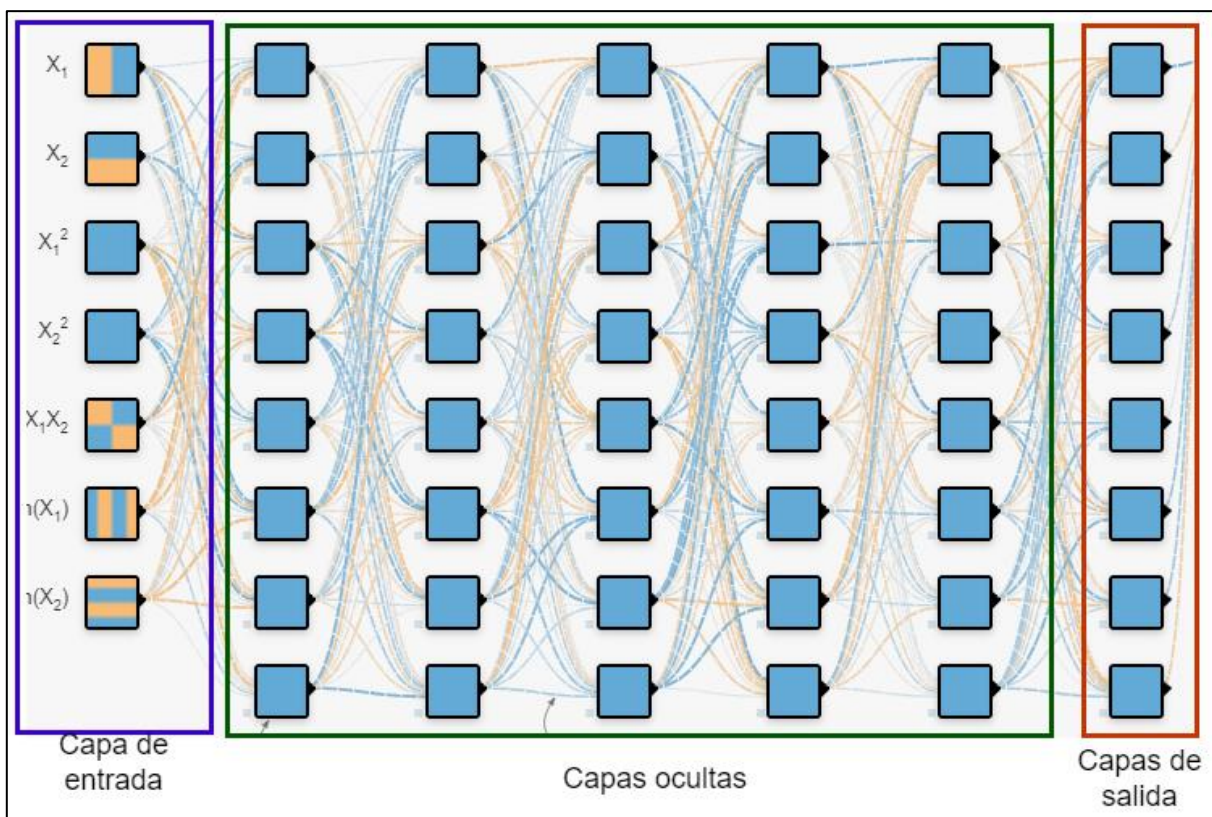


Figura 55

Arquitectura del modelo representado de forma tradicional

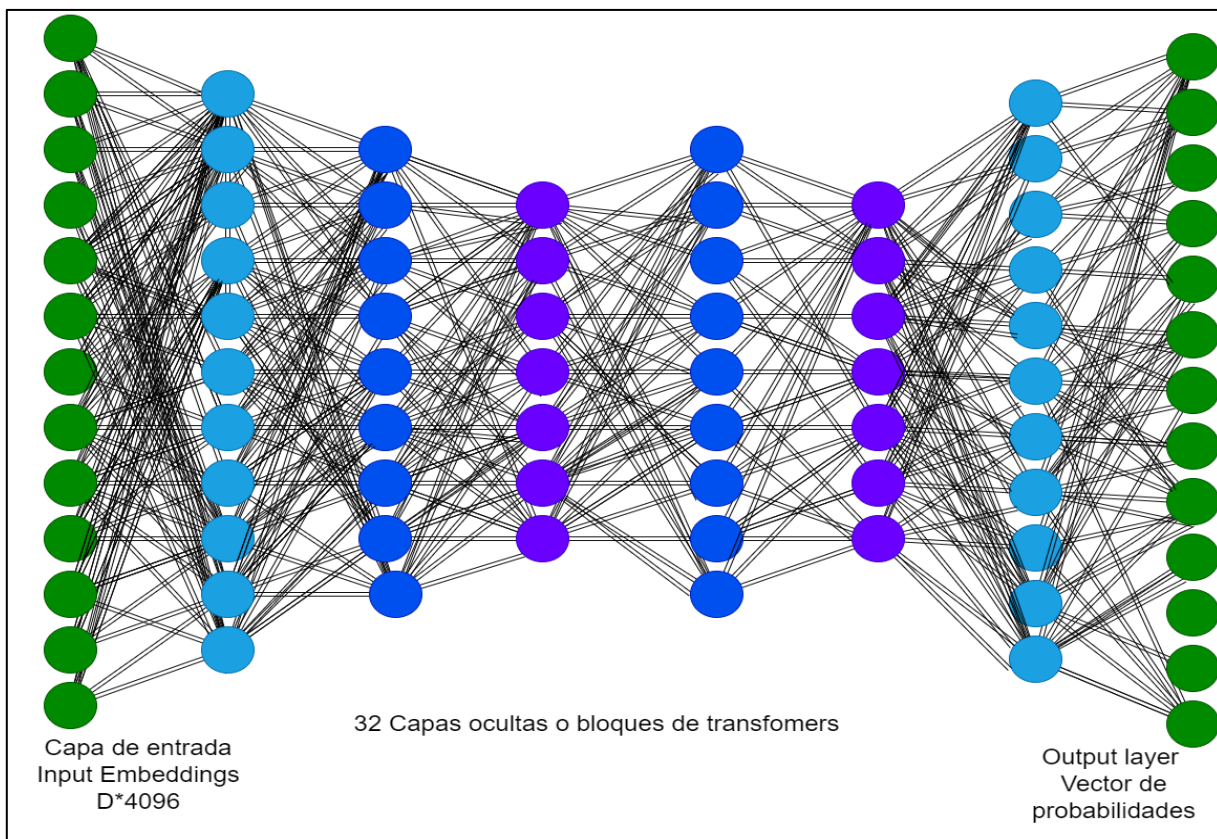


Figura 56

Preparar el modelo y configurar el tokenizador

```
# Preparar el tokenizador
from transformers import AutoModelForCausalLM, TrainingArguments, Trainer, AutoTokenizer
from peft import get_peft_model, LoraConfig, TaskType
from datasets import Dataset

model_name = "DevCar/red-neuronal-base"

# Cargar el modelo base
model = AutoModelForCausalLM.from_pretrained(
    model_name,
    load_in_8bit=True,
    device_map="auto",
)
tokenizer = AutoTokenizer.from_pretrained(model_name)

# Verificar si ya tiene un pad_token
print(f"Pad Token: {tokenizer.pad_token}")

if tokenizer.pad_token is None:
    tokenizer.pad_token = tokenizer.eos_token # Usa eos_token como pad_token

# Verificar que se ha asignado correctamente
print(f"Pad Token: {tokenizer.pad_token}")
```

Figura 57

Tokenizar el DataSet

```
# TOKENIZAR EL DATASETS

# Función para preprocesar los datos
def preprocess_function(examples):
    # Concatenamos 'area' y 'contenido' para crear el texto de entrada
    texts = [f"Área: {area}\nContenido: {contenido}" for area, contenido in zip(examples['area'], examples['contenido'])]
    return tokenizer(texts, padding=True, truncation=True, max_length=700)

# Preprocesar el dataset
train_dataset = dataset['train'].map(preprocess_function, batched=True)
test_dataset = dataset['test'].map(preprocess_function, batched=True)
```

Figura 58

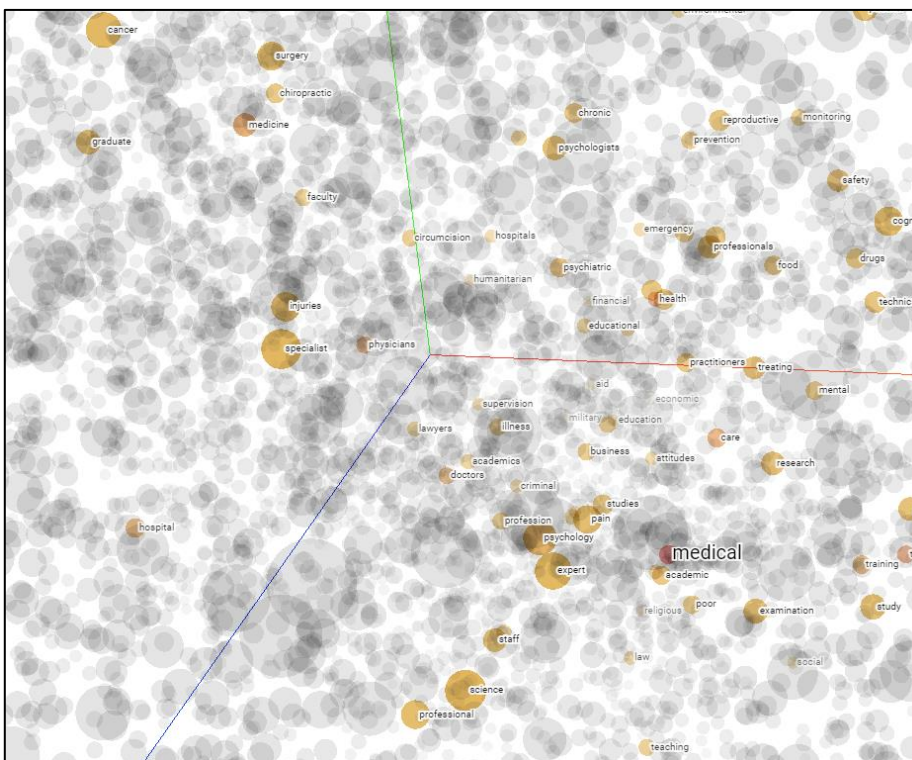
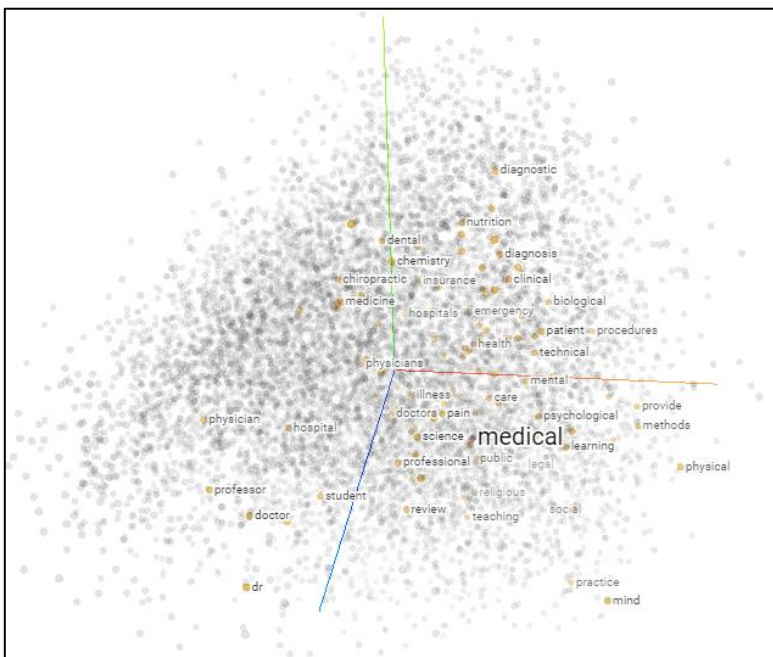
Entrenamiento de la red neuronal

```
# Configurar QLoRA
peft_config = LoraConfig(
    task_type=TaskType.CAUSAL_LM,
    r=16,
    lora_alpha=32,
    lora_dropout=0.05
)
model = get_peft_model(model, peft_config)

# Configuración de entrenamiento
training_args = TrainingArguments(
    output_dir="Internado-Rotatorio-Model", # Carpeta donde se guardará el modelo entrenado
    per_device_train_batch_size=4,
    gradient_accumulation_steps=4,
    num_train_epochs=6,
    learning_rate=2e-4,
    logging_dir='logs',
    logging_steps=10,
    fp16=True # Acelera el entrenamiento en GPUs compatibles
)

# Crear el Trainer
trainer = Trainer(
    model=model,
    args=training_args,
    train_dataset=train_dataset,
    eval_dataset=test_dataset,
    compute_metrics=compute_metrics
)

# Iniciar el entrenamiento
trainer.train()
```

Figura 59*Embedding, espacio vectorial de los tokens***Figura 60***Embedding*

3.6. DESARROLLO Y EVALUACIÓN

En esta fase se desarrolla el producto, se implementa la arquitectura del sistema definidas en las anteriores fases.

En cuanto al modelo, se lleva a cabo una evaluación de la precisión y efectividad del LLM en sus tareas de generación de texto. Se comparan los resultados con criterios en el ámbito de la medicina, asegurando que el modelo tenga una exactitud alta. Cualquier deficiencia en la precisión del modelo se ajusta mediante nuevas iteraciones de entrenamiento.

3.6.1. Primera Iteración

Se configuró la estructura básica del sistema, implementando la autenticación de usuarios y la conexión a la base de datos en PostgreSQL. Se desarrollaron las interfaces para el inicio de sesión, visualización del perfil y actualización de credenciales.

Figura 61

Módulo Inicio de sesión

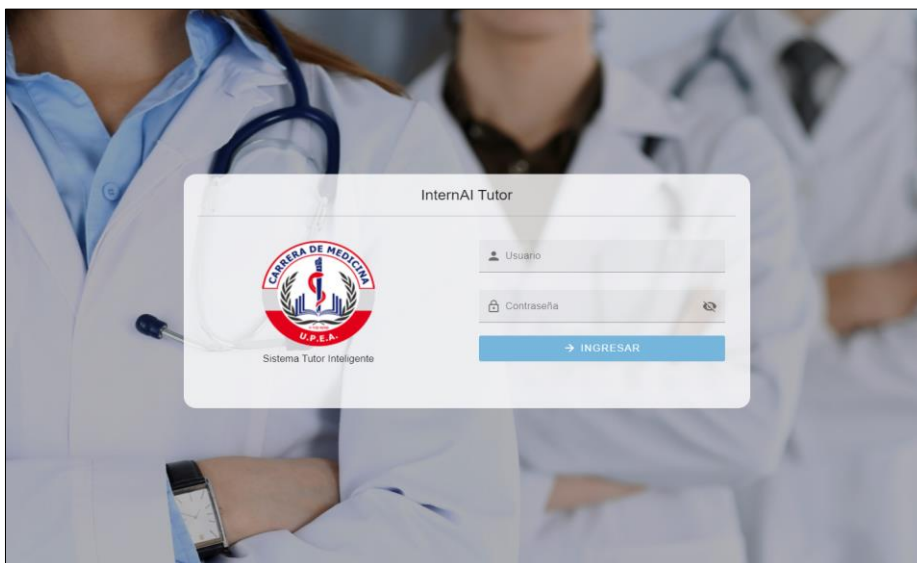
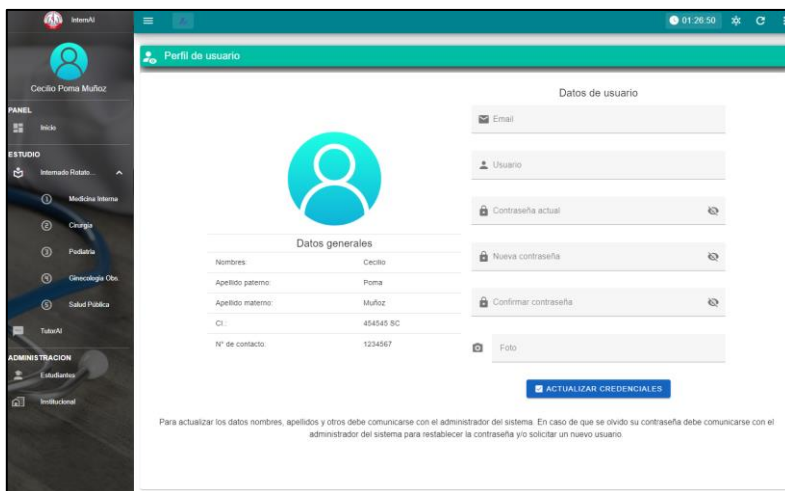


Figura 62

Interfaz perfil de usuario



3.6.2. Segunda Iteración

Se desarrollaron tanto el frontend como el backend para gestionar la información de los estudiantes, permitiendo al usuario registrar, editar y visualizar datos de los estudiantes.

Figura 63

Módulo Estudiante

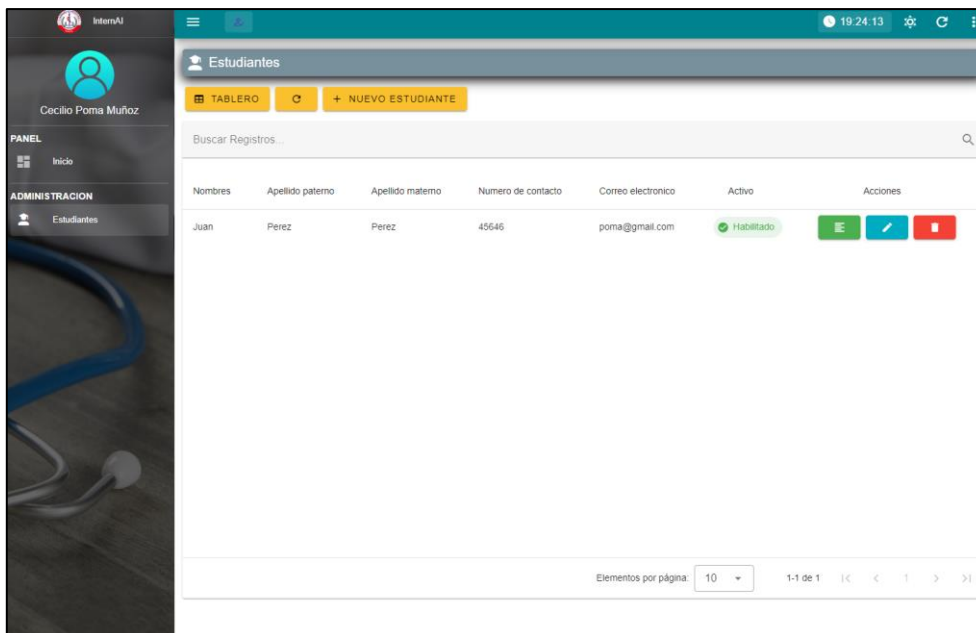


Figura 64

Módulo Estudiante, ver información del registro

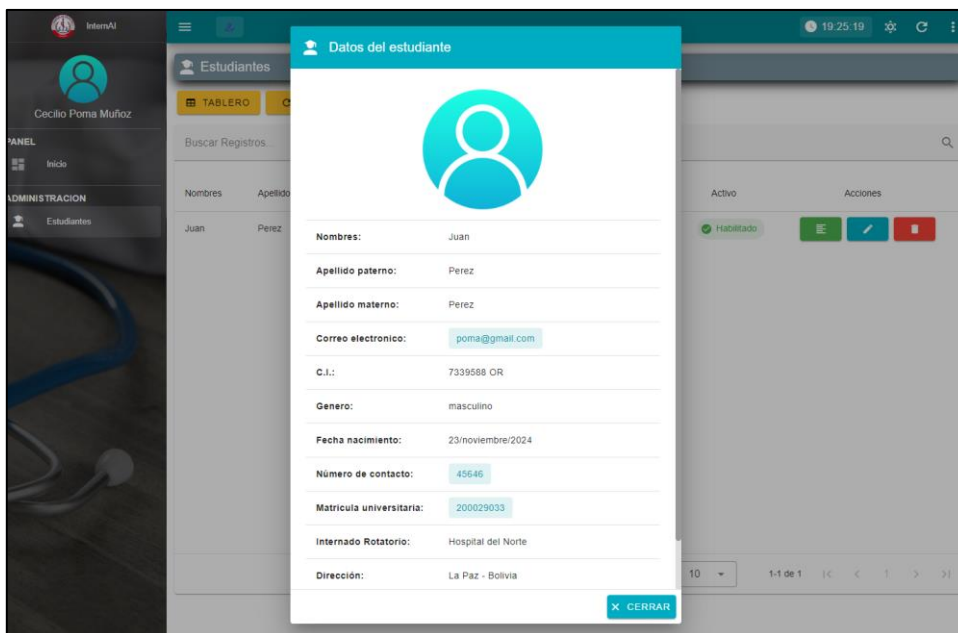
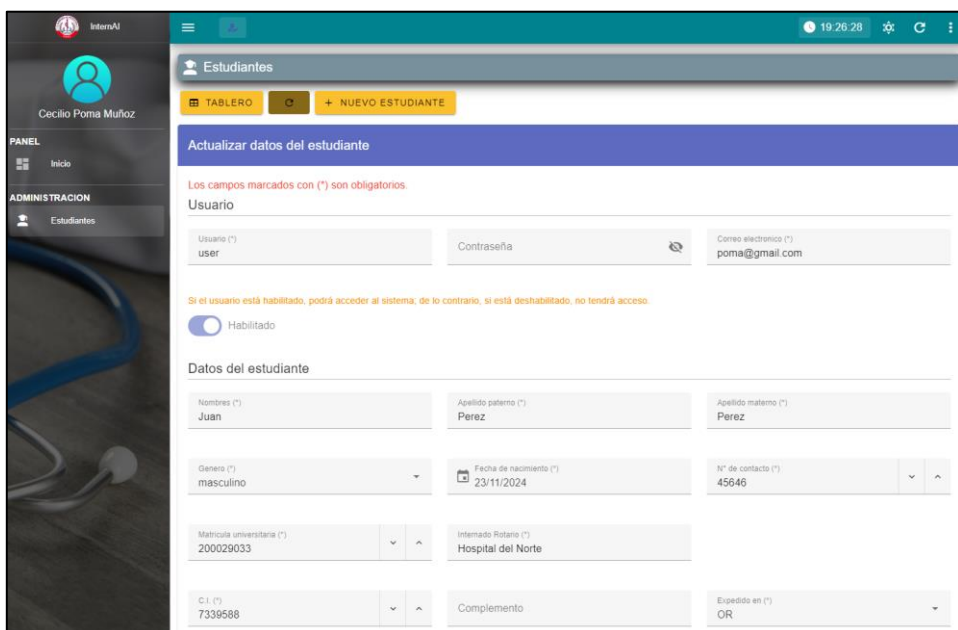


Figura 65

Módulo Estudiante, interfaz para crear nuevo registro



3.6.3. Tercera Iteración

Se implementaron las interfaces y la lógica de backend para gestionar al personal institucional, permitiendo el registro y asignación de permisos específicos en el sistema.

Figura 66

Módulo Institucional

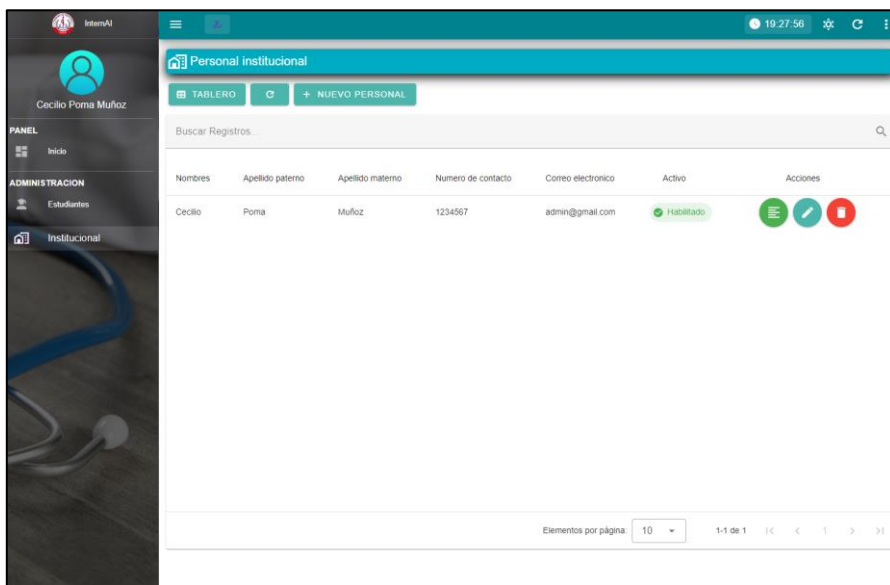


Figura 67

Módulo Institucional, interfaz para crear nuevo registro

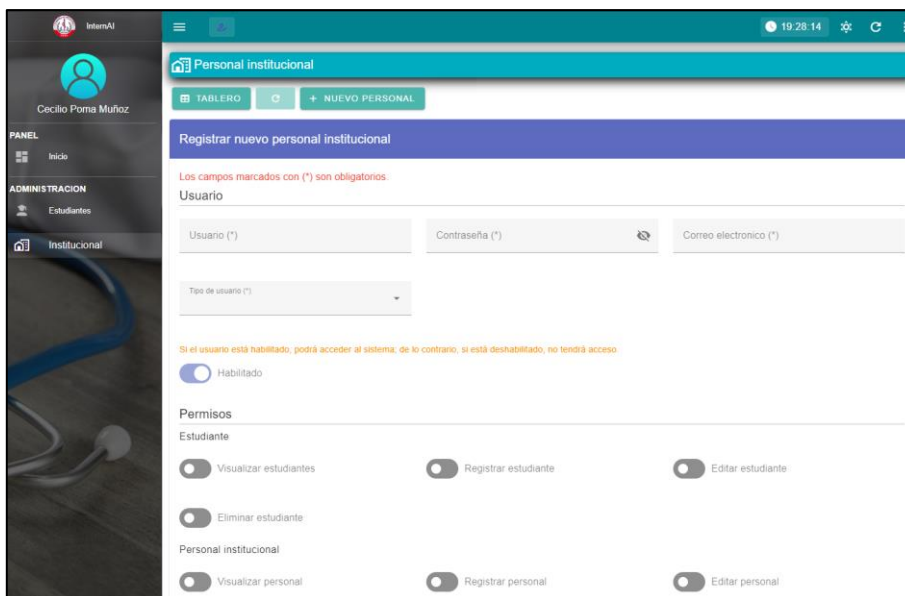
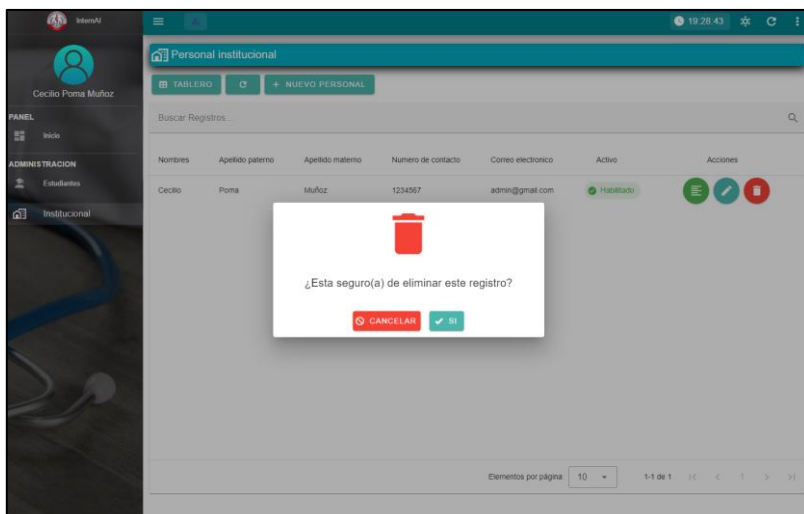


Figura 68

Modulo Institucional, interfaz para eliminar registro

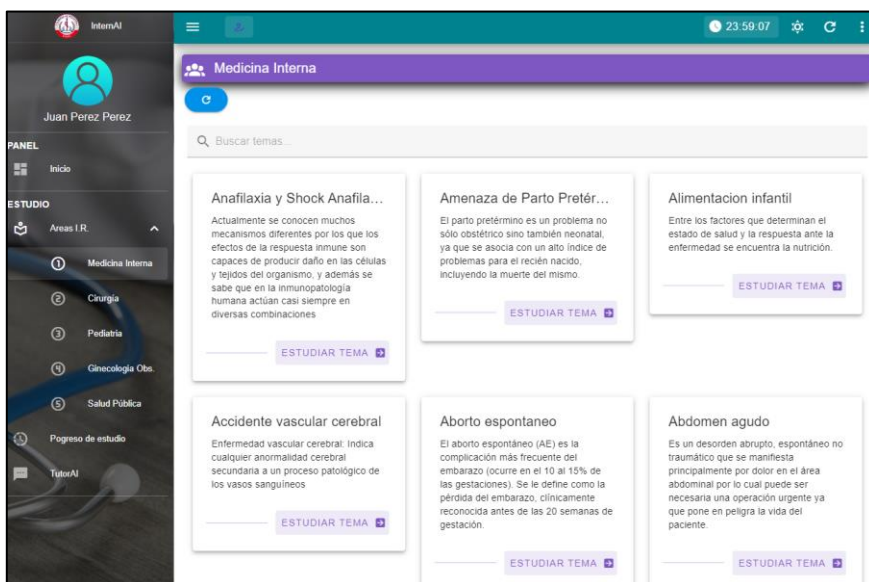


3.6.4. Cuarta Iteración

Se desarrollaron las interfaces para el acceso fácil a los materiales y temas de cada área del internado rotatorio, integrando el frontend con el backend para cargar los datos de cada área del Internado Rotatorio.

Figura 69

Modulo Áreas del Internado Rotatorio



3.6.5. Quinta Iteración

Se desarrolló una interfaz donde se carga el material de estudio, donde el frontend se comunica con el backend, así de esta manera la red neuronal genera y evalúa cuestionarios automáticamente según los temas de estudio.

Figura 70

Sub Módulo Estudio

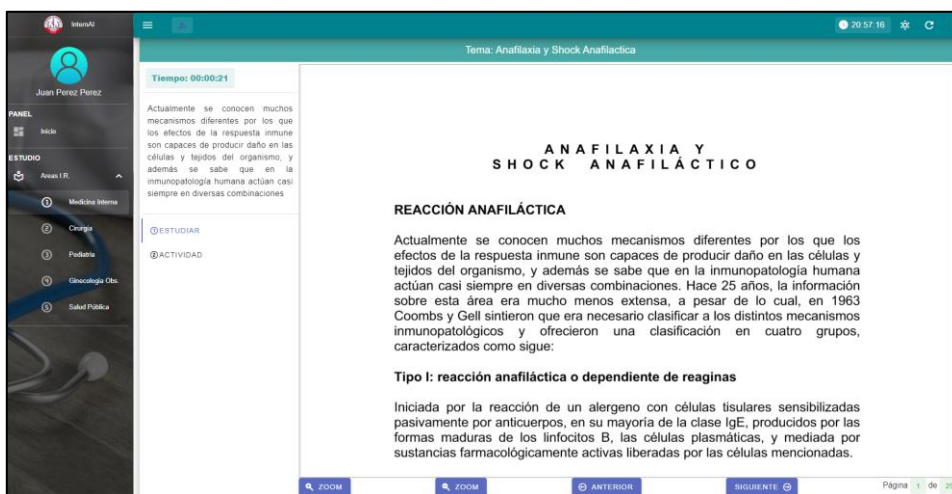
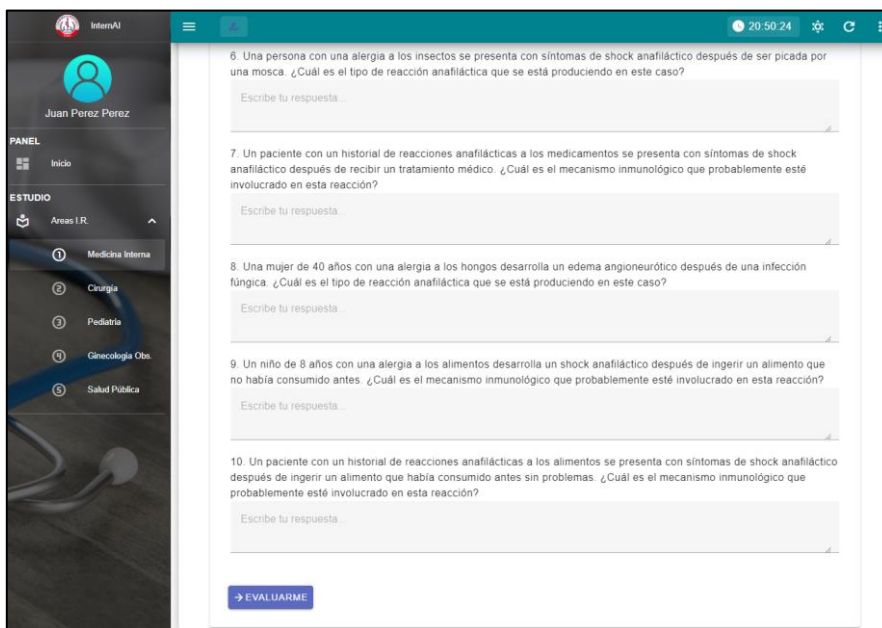


Figura 71

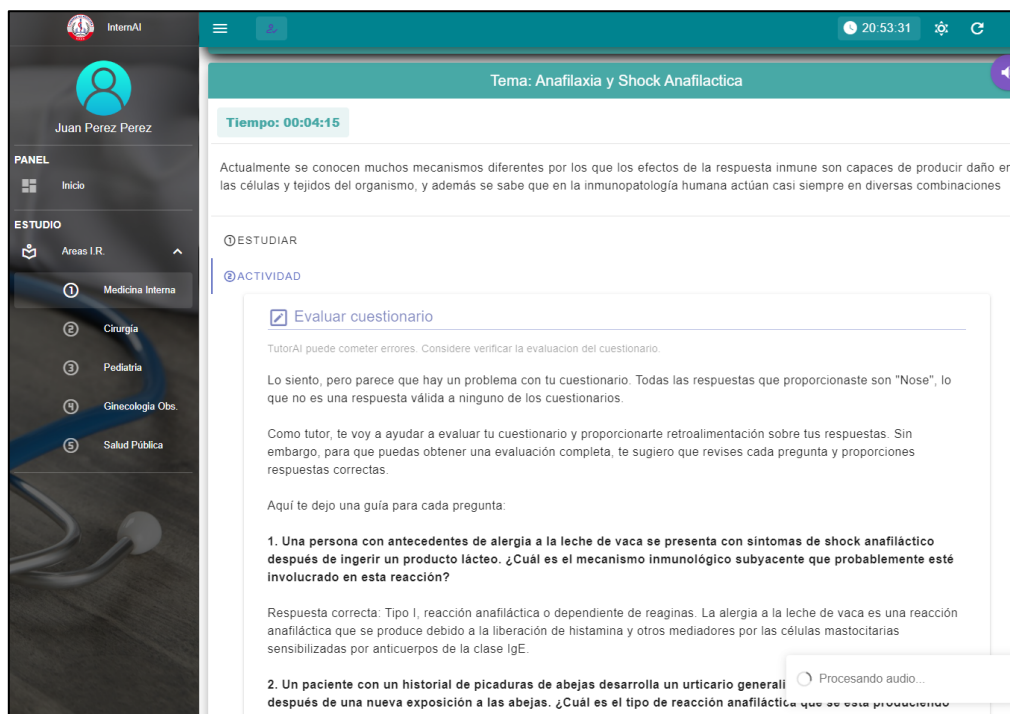
Sub Módulo Estudio, generación de cuestionario



Se envía desde el frontend las preguntas con su respectiva respuesta mediante una petición fetch de JavaScript al backend donde la red neuronal recibe como datos, posteriormente realiza la predicción y retorna un StreamingHttpResponse, para evitar los tiempos de espera y visualizar los datos de salida a medida que la red neuronal realiza la predicción.

Figura 72

Sub Módulo Estudio, evaluación de cuestionario

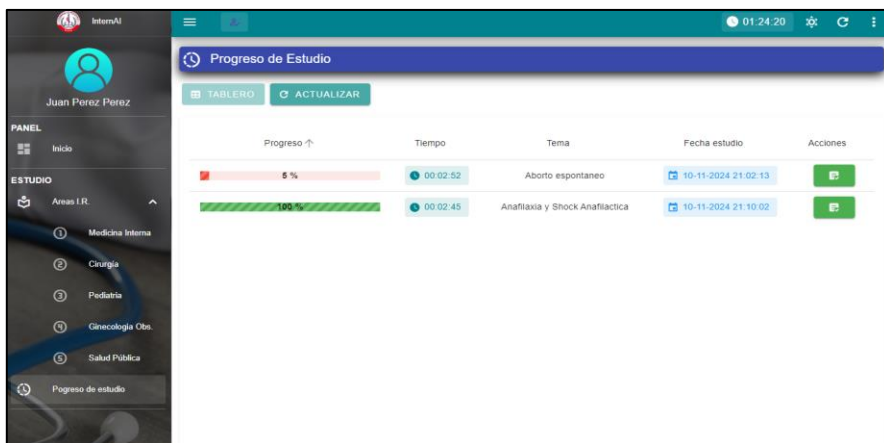


3.6.6. Sexta Iteración

Se implementó una interfaz y backend para que el estudiante visualice su avance (Progreso de estudio) en cada área y tema, registrando los cuestionarios completados y su rendimiento general.

Figura 73

Módulo Progreso de estudio

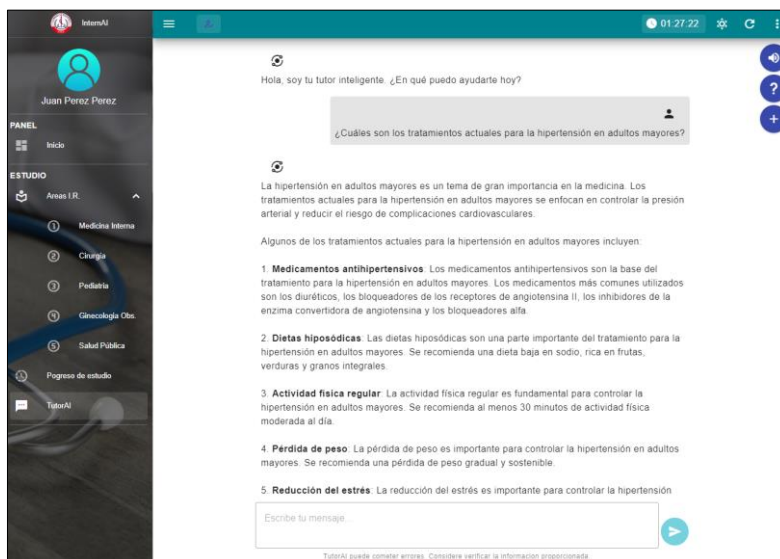


3.6.7. Séptima Iteración

A través de una solicitud fetch, el frontend envía un prompt (texto) al backend. Posteriormente la red neuronal procesa esta información y realiza una predicción. Para ofrecer una experiencia más fluida, el backend emplea StreamingHttpResponse, lo que permite visualizar los resultados de la predicción a medida que se generan. Por último, el backend procede a generar el audio y lo envía al frontend para una experiencia de usuario única.

Figura 74

Módulo TutorAI



3.6.8. Pruebas Primera Iteración

Tabla 17

Prueba funcional iniciar sesión con credenciales incorrectas

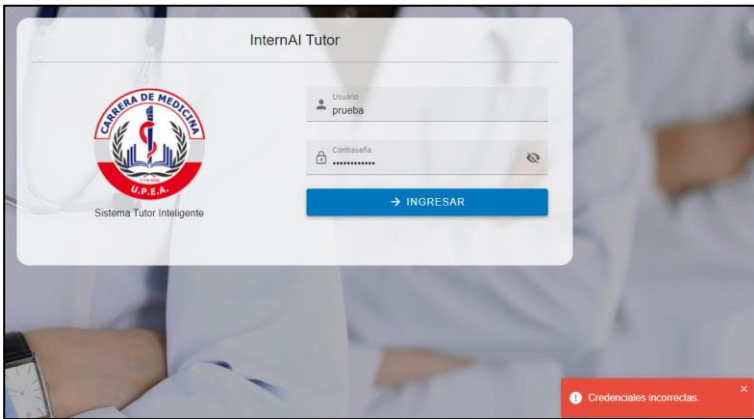
Prueba Funcional: Iniciar de Sesión	
Id único:	CP-001
Caso de prueba:	Iniciar sesión con credenciales incorrectas.
Prioridad:	Alta
Precondiciones:	<ol style="list-style-type: none"> 1. Navegador web. 2. Acceso a internet.
Datos de entrada:	<ul style="list-style-type: none"> • Usuario • Contraseña
Pasos:	<ol style="list-style-type: none"> 1. Ingresar al sistema mediante la dirección web. 2. Introducir las credenciales de acceso usuario y contraseña. 3. Click en “Ingresar”.
Resultado esperado:	El sistema no debe permitir el inicio de sesión, debe mostrar el siguiente mensaje “Credenciales incorrectas.”
Resultado obtenido:	
Estado de prueba:	OK

Tabla 18

Prueba funcional iniciar sesión con credenciales correctas

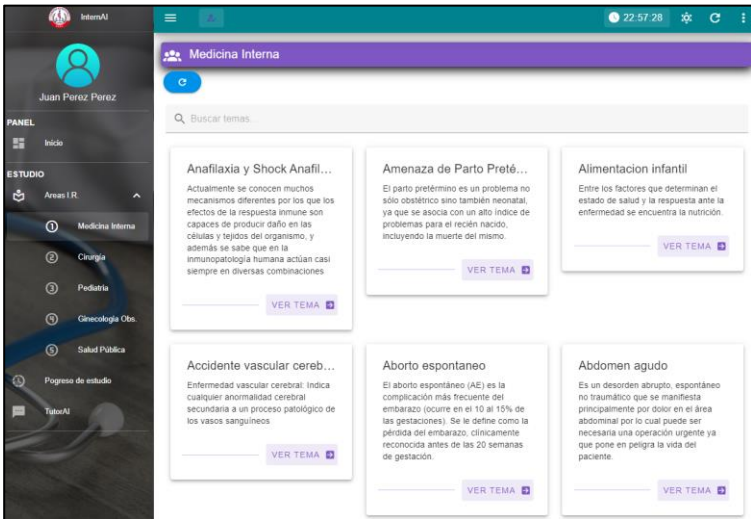
Prueba Funcional: Iniciar Sesión	
Id único:	CP-002
Caso de prueba:	Iniciar sesión con credenciales correctas.
Prioridad:	Alta
Precondiciones:	<ol style="list-style-type: none"> 1. Navegador web. 1. Acceso a internet.
Datos de entrada:	<ul style="list-style-type: none"> • Usuario • Contraseña
Pasos:	<ol style="list-style-type: none"> 1. Ingresar al sistema mediante la dirección web. 2. Introducir las credenciales de acceso usuario y contraseña. 3. Click en “Ingresar”.
Resultado esperado:	El sistema debe permitir el inicio de sesión.
Resultado obtenido:	
Estado de prueba:	OK

Tabla 19

Prueba de Integración, app authentication, endpoint n-login

Prueba de Integración: App authentication, endpoint n-login, método post	
Id único:	CP-003
Caso de prueba:	Enviar credenciales incorrectas a la app de authentication.
Prioridad:	Alta
Precondiciones:	<ol style="list-style-type: none"> 1. Instalar pytest-django. 2. Instalar Visual Studio Code.
Datos de entrada:	<ul style="list-style-type: none"> • user • password
Pasos:	<ol style="list-style-type: none"> 1. Crear una función con pytest-django 2. Hacer una petición post con APIClient
Resultado esperado:	El endpoint, retornar un estado de respuesta 401.
Resultado obtenido:	<pre> class LoginView(APIView): permission_classes = [AllowAny] # Permitir el acceso a todos def post(self, request): try: # Obtener los datos enviados request_user = request.data.get("user") request_password = request.data.get("password") </pre> <pre> @pytest.mark.django_db() def test_login_view_post(): print("\n") print("*****" * 5) datos_de_entrada = { "user": "admin", "password": "otro_password" } # Obtener la URL mediante el nombre de la ruta url = reverse('n-login') client = APIClient() # Hacemos una solicitud POST al endpoint de login response = client.post(url, datos_de_entrada, format='json') print("=> app authenticate, endpoint n-login") print(response.status_code) print("Prueba de integracion exitosa") assert response.status_code == status.HTTP_401_UNAUTHORIZED </pre>

Prueba de Integración: App authentication, endpoint n-login, método post

```
[DYL.CASA] > pytest testing/test_app_authenticate.py -s
===== test session starts =====
platform win32 -- Python 3.12.2, pytest-8.3.3, pluggy-1.5.0
django: version: 5.1.2, settings: core.settings (from ini)
rootdir: C:\Users\DYL\Documents\proyecto-medicina\backend-sti-medicina
configfile: pytest.ini
plugins: anyio-4.6.2.post1, django-4.9.0
collected 1 item

testing\test_app_authenticate.py

*****
=> app authenticate, ndpoint n-login
401
Prueba de integracion exitosa
.
===== 1 passed in 2.72s =====
```

Estado de prueba: OK

Tabla 20

Prueba Unitaria, App authentication, verificar contraseña

Prueba Unitaria: App authentication, verificar contraseña

Id único: CP-003

Caso de prueba: Verificar si la contraseña esta encriptada de forma correcta

Prioridad: Alta

Precondiciones:

1. Instalar pytest-django.
2. Instalar Visual Studio Code.

Datos de entrada:

- user

Pasos:

1. Crear una función con pytest-django
2. Verificar la contraseña con Auth

Resultado esperado: Validar contraseña en texto plano y contraseña encriptada

Prueba Unitaria: App authentication, verificar contraseña

**Resultado
obtenido:**

```
class Auth:
    @staticmethod
    def encrypt_password(raw_password):
        """Encripta la contraseña proporcionada."""
        return make_password(raw_password)

    @staticmethod
    def check_password(raw_password, hashed_password):
        """Verifica si la contraseña proporcionada coincide con la contraseña encriptada."""
        return check_password(raw_password, hashed_password)

    @staticmethod
    def user(request):
        return request.user
```

```
from apps.usuario.models import Usuario
from apps.authentication.utils import Auth
@pytest.mark.django_db()
def test_unit_authentication_password():
    print("\n")
    usuario = Usuario.objects.get(user="admin")

    if Auth.check_password("1234", usuario.password):
        print("Prueba unitaria exitosa")
        assert Auth.check_password("1234", usuario.password)
    else:
        print("Prueba unitaria fallida")
        assert Auth.check_password("1234", usuario.password)
```

```
[DYL.CASA] > pytest testing/test_unit_app_authentication.py -s
===== test session starts =====
platform win32 -- Python 3.12.2, pytest-8.3.3, pluggy-1.5.0
django: version: 5.1.2, settings: core.settings (from ini)
rootdir: C:\Users\DYL\Documents\proyecto-medicina\backend-sti
configfile: pytest.ini
plugins: anyio-4.6.2.post1, django-4.9.0
collected 1 item

testing\test_unit_app_authentication.py

Prueba unitaria exitosa
.

===== 1 passed in 1.87s =====
```

**Estado de
prueba:**

OK

3.6.9. Pruebas Segunda Iteración

Tabla 21

Prueba Funcional, Registrar estudiante

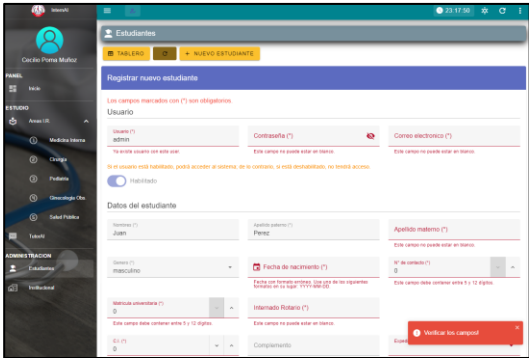
Prueba Funcional: Registrar estudiante	
Id único:	CP-003
Caso de prueba:	Registrar nuevo estudiante con datos no válidos.
Prioridad:	Alta
Precondiciones:	<ol style="list-style-type: none"> 1. Navegador web. 2. Acceso a internet. 3. Debe estar registrado en el sistema.
Datos de entrada:	<p>Datos del estudiante.</p> <ol style="list-style-type: none"> 1. Ingresar al sistema mediante la dirección web. 2. Iniciar sesión. 3. Ir al menú, opción “Estudiantes”. 4. Click en “Nuevo estudiante”. 5. Llenar todos los campos. 6. Click en “Guardar”.
Pasos:	
Resultado esperado:	El sistema no debe permitir el registro, porque los datos no son válidos.
Resultado obtenido:	
Estado de prueba:	OK

Tabla 22

Prueba de Integración, app estudiante, endpoint n-estudiante-list-create

Prueba de Integración: App estudiante, endpoint n-estudiante-list-create, método post	
Id único:	CP-003
Caso de prueba:	Enviar un token de acceso invalido al endpoint
Prioridad:	Alta
Precondiciones:	<ol style="list-style-type: none"> 1. Instalar pytest-django. 2. Instalar Visual Studio Code.
Datos de entrada:	Ninguno
Pasos:	<ol style="list-style-type: none"> 1. Crear una función con pytest-django 2. Solicitud usando APIClient
Resultado esperado:	El endpoint debe retornar un estado de respuesta 401.
Resultado obtenido:	<pre> from django.urls import reverse from rest_framework.test import APIClient @pytest.mark.django_db() def test_estudiante_endpoint_post(): print("\n") print("*****" * 5) # Obtener la URL mediante el nombre de la ruta url = reverse('n-estudiante-list-create') client = APIClient() client.credentials(HTTP_AUTHORIZATION=f'Bearer token-invalido') # Hacemos una solicitud POST response = client.post(url) print("=> app estudiante, endpoint n-estudiante-list-create") print(response.status_code) print("Prueba de integracion exitosa") assert response.status_code == status.HTTP_401_UNAUTHORIZED </pre>

Prueba de Integración: App estudiante, endpoint n-estudiante-list-create, método post

```
[DYL.CASA] > pytest testing/test_integ_app_estudiante.py -s
===== test session starts =====
platform win32 -- Python 3.12.2, pytest-8.3.3, pluggy-1.5.0
django: version: 5.1.2, settings: core.settings (from ini)
rootdir: C:\Users\DYL\Documents\proyecto-medicina\backend-sti-medicina
configfile: pytest.ini
plugins: anyio-4.6.2.post1, django-4.9.0
collected 1 item

testing\test_integ_app_estudiante.py

*****
=> app estudiante, endpoint n-estudiante-list-create
401
Prueba de integracion exitosa
.
===== 1 passed in 2.42s =====
```

Estado de prueba: OK

Tabla 23

Prueba Unitaria, App estudiante, serializador

Prueba Unitaria: App estudiante, serializador

Id único: CP-003

Caso de prueba: Verificar si el serializador valida correctamente los campos

Prioridad: Alta

Precondiciones:

1. Instalar pytest-django.
2. Instalar Visual Studio Code.

Datos de entrada: Diccionario

Pasos:

1. Crear una función con pytest-django
2. Instanciar el serializador de la app estudiant

Prueba Unitaria: App estudiante, serializador

Resultado esperado: El serializador debe indicar que los datos del diccionario no son válidos.

```
class UsuarioEstudianteSerializer(serializers.ModelSerializer):
    usuario = UsuarioSerializer()

    class Meta:
        model = Estudiante
        fields = "__all__" # serializar todos
        # solo lectura
        read_only_fields = (
            "created_at",
            "uuid",
        )
        extra_kwargs = {
            # write_only => El campo NO se devuelve en las respuestas
            "is_status": {"write_only": True},
            "numero_contacto": {"validators": [custom_number_validator]},
            "matricula_univ": {"validators": [custom_number_validator]},
            "ci": {"validators": [custom_number_validator]},
            "ci_complemento": {"validators": [custom_ci_complemento_validator]},
        }
```

Resultado obtenido:

```
from apps.estudiante.serializers import UsuarioEstudianteSerializer
@pytest.mark.django_db()
def test_unit_estudiante_serializer():
    print("\n")
    datos = {}
    estudiante_serializer = UsuarioEstudianteSerializer(data = datos)

    if estudiante_serializer.is_valid():
        print("Prueba unitaria fallida")
        assert estudiante_serializer.data
    else:
        print("Prueba unitaria exitosa")
        assert estudiante_serializer.errors
```

```
[DYL.CASA] > pytest testing/test_unit_app_estudiante.py -s
===== test session starts =====
platform win32 -- Python 3.12.2, pytest-8.3.3, pluggy-1.5.0
django: version: 5.1.2, settings: core.settings (from ini)
rootdir: C:\Users\DYL\Documents\proyecto-medicina\backend-sti-medicina
configfile: pytest.ini
plugins: anyio-4.6.2.post1, django-4.9.0
collected 1 item

testing\test_unit_app_estudiante.py
Prueba unitaria exitosa
.
===== 1 passed in 0.17s =====
```

Prueba Unitaria: App estudiante, serializador

Estado de prueba: OK

3.6.10. Pruebas Tercera Iteración

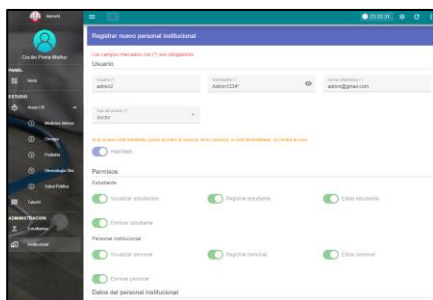
Tabla 24

Prueba funcional registrar personal institucional

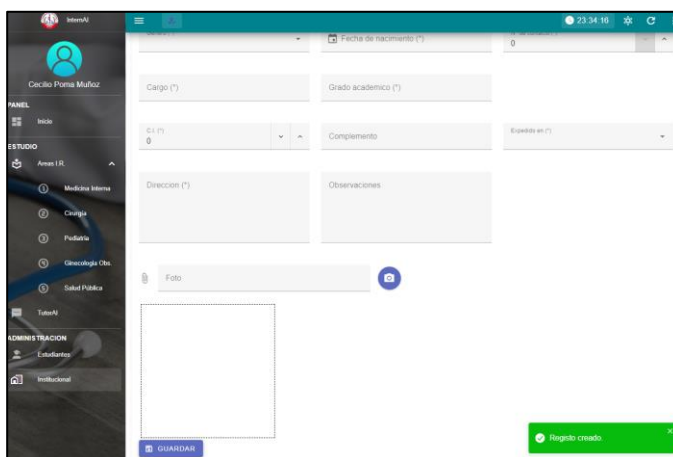
Prueba Funcional: Registrar Personal Institucional

Id único:	CP-004
Caso de prueba:	Registrar nuevo personal con datos válidos.
Prioridad:	Alta
Precondiciones:	<ol style="list-style-type: none"> 1. Navegador web. 2. Acceso a internet. 3. Debe estar registrado en el sistema.
Datos de entrada:	<p>Datos del personal institucional.</p> <ol style="list-style-type: none"> 1. Ingresar al sistema mediante la dirección web. 2. Iniciar sesión. 3. Ir al menú, opción "Institucional".
Pasos:	<ol style="list-style-type: none"> 4. Click en "Nuevo personal". 5. Llenar todos los campos. 6. Asignar todos los roles. 7. Click en "Guardar".
Resultado esperado:	El sistema debe registrar.

Prueba Funcional: Registrar Personal Institucional



Resultado obtenido:



Estado de prueba: OK

Tabla 25

Prueba de Integración, app personal, endpoint n-personal-list-create

Prueba de Integración: App personal, endpoint n-personal-list-create, método post

Id único: CP-003

Caso de prueba: Verificar que el método endpoint n-personal-list-create, método post devuelve un estado respuesta 422

Prioridad: Alta

Precondiciones:

1. Instalar pytest-django.
2. Instalar Visual Studio Code.

Datos de entrada:

Token de acceso

Pasos:

1. Crear una función con pytest-django
2. Crear una petición HTTP usando APIClient

Resultado esperado:

El endpoint debe retornar un estado de respuesta 422.

Resultado obtenido:

```
from django.urls import reverse
from rest_framework.test import APIClient
from apps.usuario.models import Usuario
from apps.authentication.utils import Auth
from rest_framework_simplejwt.tokens import RefreshToken
@pytest.mark.django_db()
def test_personal_endpoint_post():
    print("\n")
    print("*****" * 5)

    # Obtener el token de acceso
    user = Usuario.objects.get(user = "admin")
    if Auth.check_password("1234", user.password):
        refresh = RefreshToken.for_user(user)
        access_token = str(refresh.access_token)
    else:
        access_token = "no-token"

    # Obtener la URL mediante el nombre de la ruta
    url = reverse('n-personal-list-create')
    client = APIClient()
    client.credentials(HTTP_AUTHORIZATION=f'Bearer {access_token}')
    # Hacemos una solicitud POST
    response = client.post(url)

    print("=> App personal_institucional, endpoint n-personal-list-create, metodo post")
    print(response.status_code)
    print("Prueba de integracion exitosa")
    assert response.status_code == status.HTTP_422_UNPROCESSABLE_ENTITY
```

```
[DYL.CASA] > pytest testing/test_integ_app_personal_institucional.py -s
===== test session starts =====
platform win32 -- Python 3.12.2, pytest-8.3.3, pluggy-1.5.0
django: version: 5.1.2, settings: core.settings (from ini)
rootdir: C:\Users\DYL\Documents\proyecto-medicina\backend-sti-medicina
configfile: pytest.ini
plugins: anyio-4.6.2.post1, django-4.9.0
collected 1 item

testing\test_integ_app_personal_institucional.py

*****
=> App personal_institucional, endpoint list
422
Prueba de integracion exitosa
.
===== 1 passed in 2.68s =====
```

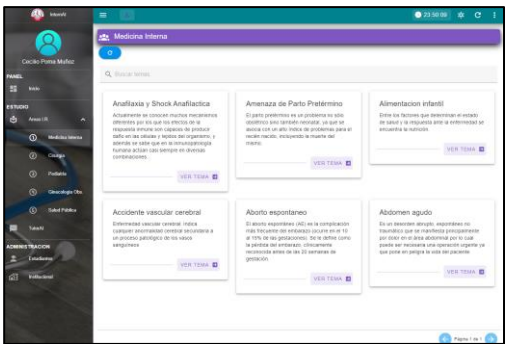
Estado de prueba:

OK

3.6.11. Pruebas Cuarta Iteración

Tabla 26

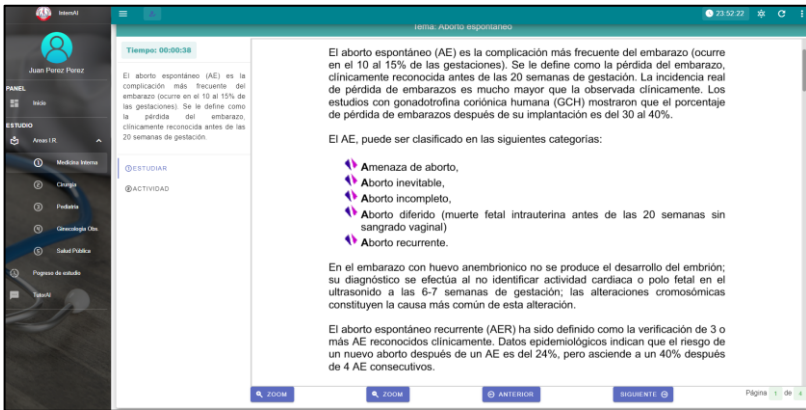
Prueba funcional Internado Rotatorio área de medicina interna

Prueba Funcional: Internado Rotatorio de Área Medicina Interna	
Id único:	CP-005
Caso de prueba:	Verificar si se muestran los temas del Internado Rotatorio del área “medicina interna”.
Prioridad:	Alta
Precondiciones:	<ol style="list-style-type: none"> 1. Navegador web. 2. Acceso a internet. 3. Debe estar registrado en el sistema.
Datos de entrada:	Ninguno
Pasos:	<ol style="list-style-type: none"> 1. Ingresar al sistema mediante la dirección web. 2. Iniciar sesión. 3. Ir al menú, opción “medicina interna”.
Resultado esperado:	Visualizar todos los temas del área de medicina interna.
Resultado obtenido:	
Estado de prueba:	OK

3.6.12. Pruebas Quinta Iteración

Tabla 27

Prueba Funcional cargar material de estudio

Prueba Funcional: Cargar Material de Estudio	
Id único:	CP-006
Caso de prueba:	Verificar si se visualiza/carga el material de estudio.
Prioridad:	Alta
Precondiciones:	<ol style="list-style-type: none"> 1. Navegador web. 2. Acceso a internet. 3. Debe estar registrado en el sistema.
Datos de entrada:	Ninguno
Pasos:	<ol style="list-style-type: none"> 1. Ingresar al sistema mediante la dirección web. 2. Iniciar sesión. 3. Ir al menú, opción “medicina interna”. 4. Seleccionar el tema “Aborto espontaneo”. 5. Click en “Estudiar tema”.
Resultado esperado:	Cargar el material de estudio.
Resultado obtenido:	
Estado de prueba:	OK

3.6.13. Pruebas Sexta Iteración

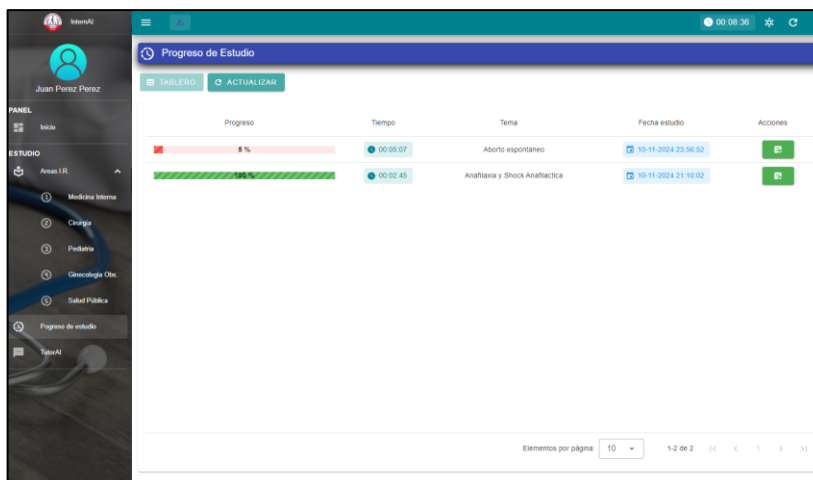
Tabla 28

Prueba Funcional, cargar progreso de estudio

Prueba Funcional: Cargar Progreso de Estudio

Id único:	CP-007
Caso de prueba:	Verificar si cargan los datos del progreso de estudio del estudiante.
Prioridad:	Alta
Precondiciones:	<ol style="list-style-type: none"> 1. Navegador web. 2. Acceso a internet. 3. Debe estar registrado en el sistema.
Datos de entrada:	Ninguno
Pasos:	<ol style="list-style-type: none"> 1. Ingresar al sistema mediante la dirección web. 2. Iniciar sesión. 3. Ir al menú, opción "Progreso de estudio".
Resultado esperado:	Cargar los datos del progreso de estudio del estudiante.

Resultado obtenido:



Estado de prueba: OK

Tabla 29

Prueba Unitaria, App estudiante, listar progreso de estudio

Prueba Unitaria: App estudiante, listar progreso de estudio	
Id único:	CP-003
Caso de prueba:	Verificar el progreso de estudio de todos los estudiantes.
Prioridad:	Media
Precondiciones:	<ol style="list-style-type: none">1. Instalar pytest-django.2. Instalar Visual Studio Code.
Datos de entrada:	Diccionario
Pasos:	<ol style="list-style-type: none">1. Crear una función con pytest-django.2. Listar progreso de estudio utilizando el modelo ProgresoEstudio.
Resultado esperado:	El modelo debe listar todos los progresos de estudio de todos los estudiantes.
Resultado obtenido:	<pre>from apps.estudiante.models import ProgresoEstudio @pytest.mark.django_db() def test_unit_estudiante_progreso_estudio(): print("\n") datos = {} progreso_estudio = ProgresoEstudio.objects.all() print("ProgresoEstudio => Prueba unitaria exitosa") assert progreso_estudio</pre>

Prueba Unitaria: App estudiante, listar progreso de estudio

```
[DYL.CASA] > pytest testing/test_unit_app_estudiante.py -s
===== test session starts =====
platform win32 -- Python 3.12.2, pytest-8.3.3, pluggy-1.5.0
django: version: 5.1.2, settings: core.settings (from ini)
rootdir: C:\Users\DYL\Documents\proyecto-medicina\backend-sti-medicina
configfile: pytest.ini
plugins: anyio-4.6.2.post1, django-4.9.0
collected 2 items

testing\test_unit_app_estudiante.py

UsuarioEstudianteSerializer => Prueba unitaria exitosa
.

ProgresoEstudio => Prueba unitaria exitosa
.

===== 2 passed in 0.13s =====
```

Estado de prueba: OK

3.6.14. Pruebas Séptima Iteración

Tabla 30

Prueba funcional interactuar con TutorAI (Red neuronal)

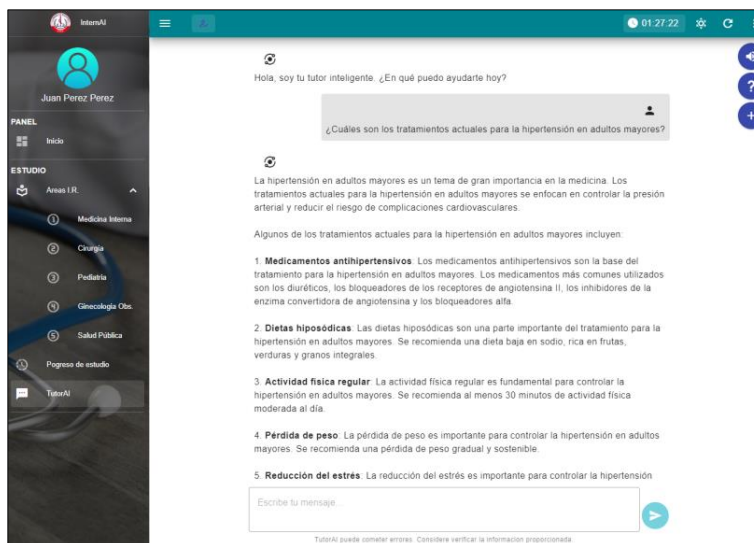
Prueba Funcional: Interactuar con TutorAI (Red Neuronal)

Id único:	CP-008
Caso de prueba:	Interactuar con TutorAI (red neuronal), enviar un prompt y verificar si TutorAI logra responder.
Prioridad:	Alta
Precondiciones:	<ol style="list-style-type: none"> 1. Navegador web. 2. Acceso a internet. 3. Debe estar registrado en el sistema.
Datos de entrada:	Prompt (texto)
Pasos:	<ol style="list-style-type: none"> 1. Ingresar al sistema mediante la dirección web. 2. Iniciar sesión. 3. Ir al menú, opción "Progreso de estudio".

Prueba Funcional: Interactuar con TutorAI (Red Neuronal)

Resultado esperado: TutorAI (red neuronal) debe generar texto en base al prompt.

Resultado obtenido:



Estado de prueba: OK

Tabla 31

Prueba Unitaria, Red Neuronal

Prueba Unitaria: Red Neuronal

Id único:	CP-003
Caso de prueba:	Verificar si la red neurona realiza la predicción de forma correcta
Prioridad:	Alta
Precondiciones:	<ol style="list-style-type: none"> 1. Instalar pytest-django. 2. Instalar Visual Studio Code.
Datos de entrada:	Token de acceso
Pasos:	<ol style="list-style-type: none"> 3. Crear una función con pytest-django 4. Instanciar la red neuronal
Resultado esperado:	La red neuronal debe realizar predicciones según, el texto de entrada

Prueba Unitaria: Red Neuronal

**Resultado
obtenido:**

```
@pytest.mark.django_db()
def test_red_neuronal():
    print("\n")
    print("*****" * 5)

    is_red_neuronal = red_neuronal(
        input="hola",
        top_p=0.9,
        temperature=0.6,
        max_tokens=2000,
        stream=True,
        seed=None,
    )
    print("Prueba unitaria exitosa")
    assert is_red_neuronal
```

Estado de prueba: OK

3.6.15. Evaluación de Los Resultados Del Modelo

Se examinan los resultados del modelo en relación con los objetivos establecidos en la fase de comprensión del negocio. Se considera la precisión, relevancia y coherencia de las preguntas generadas en temas del Internado Rotatorio.

3.6.16. Proceso de Revisión

Se revisa el proceso del entrenamiento en su totalidad para identificar áreas que puedan mejorar la precisión del modelo. Esto podría implicar regresar a fases previas como la preparación de datos o el modelado y realizar ajustes en los parámetros o en la selección de datos.

3.7. DESPLIEGUE

Finalmente, el sistema completo se despliega para los estudiantes y el personal institucional. Se entrega con una documentación detallada que describe su funcionalidad y proporciona un manual de usuario.

3.7.1. Evaluación de Satisfacción

Se realizan pruebas con los usuarios finales para evaluar la efectividad, facilidad de uso y satisfacción del sistema.

3.7.2. Planificación de Nuevas Iteraciones

En base en los resultados de las pruebas, se evaluó que no es necesario adicionar iteraciones.

3.7.3. Implementación de la red neuronal

Se implementa el LLM entrenado en el sistema educativo, integrándolo en el módulo TutorAI y en el Sub módulo Estudio del módulo áreas del Internado Rotatorio. Esta implementación permitirá que el LLM genere preguntas y brinde retroalimentación en tiempo real dentro de las áreas temáticas del internado rotatorio.

El cual el backend es el encargado de interactuar con la red neuronal, el cual este envía y recibe los datos para realizar la predicción del modelo.

CAPÍTULO IV

CALIDAD, COSTO Y

SEGURIDAD

**INGENIERÍA
DE SISTEMAS**
UNIVERSIDAD PÚBLICA DE EL ALTO



CAPÍTULO IV

4. CALIDAD, COSTO Y SEGURIDAD

4.1. INTRODUCCIÓN

En términos de costos, el desarrollo de este sistema supone una inversión significativa tanto en recursos humanos como tecnológicos. El uso de modelos de procesamiento de lenguaje natural que se entrena y optimiza mediante técnicas como QLoRA, requiere recursos computacionales de alta gama, por lo cual para el entrenamiento se optó por Google Colab Pro.

Respecto a la seguridad, es un aspecto crítico dado que el sistema maneja información sensible, tanto médica como educativa, de los estudiantes.

4.2. MÉTRICA DE CALIDAD 25010

Cada métrica tales como rendimiento, usabilidad, confiabilidad y funcionalidad, y están definidas con los principios de la norma ISO 25010.

A continuación de se detallan las métricas de calidad:

4.2.1. *Rendimiento*

Eficiencia en el desempeño, donde la métrica de tiempo de respuesta mide el intervalo desde que el estudiante envía una solicitud al backend hasta que el sistema InternAI responde.

Tabla 32

Métricas de Calidad, Rendimiento

Rendimiento: Tiempo de Respuesta	
Objetivo:	Mantener el tiempo de respuesta por debajo de 2 segundos en consultas generales, y no superior a 6 segundos en consultas que involucran cálculos intensivos de la red neuronal.
Método de evaluación:	Se realizan preguntas los usuarios finales en relación a los tiempos de respuesta del sistema.

Tabla 33*Encuesta, Métrica de Calidad de Rendimiento*

N.º	Preguntas para validar el rendimiento	Respuestas (1-16)		Porcentaje (Positivo)
		Si	No	
1	¿El sistema responde en menos de 2 segundos al realizar consultas generales?	13	3	81.3%
2	¿El tiempo de respuesta al generar cuestionarios es menor a 6 segundos?	14	2	86.5%
3	¿El TutorAI responde de forma eficiente en tareas complejas?	14	2	87.5%
4	¿Experimentó tiempos de espera superiores a 7 segundos?	2	14	87.5%
5	¿El sistema tarda menos de 2 segundos en cargar las áreas de estudio?	15	1	93.3%
Resultado 87%				

4.2.2. Usabilidad

Esta métrica evalúa la facilidad con que los usuarios (estudiantes) completan tareas sin asistencia externa.

Tabla 34*Métrica de Calidad, Usabilidad*

Usabilidad: Tasa de Éxito de Tareas	
Objetivo:	Lograr una tasa de éxito en tareas comunes del 90% o más en pruebas de usuarios, como acceder a áreas de estudio, realizar consultas al TutorAI y revisar el progreso de estudio.
Método de evaluación:	Se realizarán sesiones de pruebas con usuarios.

Tabla 35*Encuesta, Métrica de Calidad de Usabilidad*

N.º	Preguntas para validar la usabilidad	Respuestas (1-16)		Tareas completadas con éxito	Total, de tareas
		Si	No		
1	¿Pudo completar las tareas asignadas (estudio, cuestionarios, TutorAI) sin ayuda externa?	13	3	13	16
2	¿Encontró el sistema intuitivo para navegar y realizar actividades?	15	1	15	16
3	¿Pudo acceder fácilmente a las áreas de estudio?	14	2	14	16
4	¿La interfaz del sistema facilitó la realización de las tareas propuestas?	14	2	14	16
5	¿Se sintió cómodo usando las funcionalidades del sistema sin errores?	14	2	14	16

N.º	Preguntas para validar la usabilidad	Respuestas (1-16)		Tareas completadas con éxito	Total, de tareas
		Si	No		
Totales				70	80

Ecuación 1

Ecuación de usabilidad

$$tasa = \left(\frac{N}{total} \right) \times 100\%$$

Donde:

tasa = Tasa de éxito de tareas

N = Número de tareas completados con éxito

total = Número total de tareas

Calculando:

Los cálculos se llevan a cabo con base en los datos obtenidos de la tabla de encuestas.

$$tasa = \left(\frac{70}{80} \right) \times 100\% = 88\%$$

Resultado: 87 %

4.2.3. Confiabilidad

La tasa de fallos mide la frecuencia con la que el sistema experimenta errores o interrupciones durante su uso regular.

Tabla 36

Métrica de Calidad Confiabilidad

Confiabilidad: Tasa de Fallos	
Objetivo:	Mantener la tasa de fallos en el sistema tutor inteligente, incluida la red neuronal, por debajo del 3% mensual. Esto

Confiabilidad: Tasa de Fallos

implica que tanto el sistema en su conjunto como la red neuronal deben funcionar correctamente sin generar errores en las respuestas o producir fallos operativos debido a malas predicciones, datos mal entrenados o fallos en el procesamiento.

Método de Evaluación: Se identificarán todos los incidentes de fallos, como respuestas incorrectas, inestabilidad del modelo o fallos del sistema relacionados con la red neuronal.

Tabla 37
Encuesta, Métrica de Calidad Confiabilidad

N.º	Preguntas para validar la usabilidad	Respuestas (1-6)		Numero de fallos	Total, de solicitudes
		Si	No		
1	¿El sistema presentó fallos al interactuar con las áreas de estudio?	0	16	0	16
2	¿Las respuestas del TutorAI fueron precisas y sin errores evidentes?	16	0	0	16
3	¿Experimentó interrupciones inesperadas al generar o evaluar cuestionarios?	0	16	0	16
4	¿El sistema funcionó correctamente durante su uso?	16	0	0	16
5	¿Encontró algún problema de inestabilidad al realizar tareas frecuentes?	2	14	2	16

N.º	Preguntas para validar la usabilidad	Respuestas (1-6)		Numero de fallos	Total, de solicitudes
		Si	No		
Totales				2	80

Ecuación 2

Ecuación de confiabilidad

$$tasa = \left(\frac{N}{total} \right) \times 100$$

Donde:

tasa = Tasa de fallos

N = Número de fallos en el mes

total = Número total de solicitudes en el mes

Calculando:

Los cálculos se llevan a cabo con base en los datos obtenidos de la tabla de encuestas.

$$tasa = \left(\frac{13}{80} \right) \times 100 = 2\%$$

4.2.4. Funcionalidad

Involucra a usuarios en el sistema para que verifiquen que el software cumple con los requisitos funcionales definidos.

Tabla 38

Métrica de Calidad Funcionalidad

Funcionalidad: Cobertura de Funcionalidades	
Objetivo:	Lograr al menos un 95% de cobertura de las funcionalidades clave, incluyendo TutorAI, gestión de áreas y temas, y registro de progreso.

Funcionalidad: Cobertura de Funcionalidades

Método de Evaluación: Crear encuestas que soliciten a los usuarios valorar cada funcionalidad.

Tabla 39

Encuesta, Métrica de Calidad Funcionalidad

N.º	Preguntas para validar la usabilidad	Respuestas (1-16)		Porcentaje (Si)
		Si	No	
1	¿Todas las funcionalidades clave (TutorAI, cuestionarios, progreso) cumplieron con sus expectativas?	14	2	87.5%
2	¿Considera que las funcionalidades del sistema están completas y operativas?	14	2	87%
3	¿La gestión de áreas y temas cumplió con lo requerido para su estudio?	14	2	87%
4	¿El TutorAI proporcionó todas las funciones necesarias para sus consultas?	15	1	93.8%
5	¿Las funcionalidades ofrecidas facilitaron su aprendizaje?	15	1	93.8%

Resultado: 90%

4.3. DETERMINACIÓN DE COSTOS

A continuación, se detallan los costos del sistema.

4.3.1. *Licencias de Software*

La lista de software a continuación es de uso libre, por lo tanto, no tienen ningún costo.

- Python
- HTML
- CSS
- JavaScript

4.3.2. *Herramientas de Desarrollo*

Los costos presentes indican las herramientas que se utilizaron para el desarrollo del sistema.

Tabla 40

Herramientas de desarrollo

Herramienta	Características	Costo (Bs)
Google Colab Pro	<ul style="list-style-type: none"> • Mayor disponibilidad de recursos (GPUs). • Mayor disponibilidad de recursos. • Almacenamiento en caché. • Prioridad en las colas. 	400
Inference Endpoint	Inference Endpoints de Hugging Face es una solución que permite desplegar en producción modelos de machine learning de manera eficiente.	320
Computadora	<ul style="list-style-type: none"> • Procesador Core i3 • Placa base MSI • Memoria Ram de 16 GB • Sistema Operativo Linux • Almacenamiento SSD:490 GB • Teclado 	4000

Herramienta	Características	Costo (Bs)
	<ul style="list-style-type: none"> • Mouse • Cámara 	

4.3.3. Desarrollo de software

A continuación, se presenta la estimación de costos del sistema "Tutor Inteligente para el Internado Rotatorio" utilizando el método COSMIC. Este análisis considera las tres fases de medición establecidas por COSMIC: estrategia de medición, mapeo y medición.

4.3.3.1. Estrategia de medición

En esta etapa, se definieron los parámetros clave de medición de acuerdo con los lineamientos de COSMIC:

- **Propósito de la medición:** Determinar el tamaño funcional del sistema y estimar el esfuerzo y costo necesarios para su desarrollo.
- **Alcance del sistema:** Todas las funcionalidades especificadas en los requerimientos funcionales, incluyendo interacción con usuarios, procesamiento de datos, gestión de temas y cuestionarios, y comunicación con la red neuronal.
- **Nivel de descomposición:** Nivel 0, considerando el sistema completo como una unidad funcional.
- **Usuarios funcionales:** Estudiantes, personal institucional, y subsistemas (visor de PDFs, red neuronal, y bases de datos).
- **Capa de software:** Aplicación, abarcando la interacción usuario-sistema y las operaciones internas.

4.3.3.2. Mapeo

Durante esta fase, se identificaron los procesos funcionales del sistema y se analizaron los movimientos de datos correspondientes, clasificados en entradas (E), lecturas (R),

escrituras (W) y salidas (X). Cada movimiento se midió utilizando los Cosmic Function Points (CFP).

Tabla 41

Mapeo de Requerimientos Funcionales

Id	Requerimiento	Entrada (E)	Lectura (R)	Escritura (W)	Salida (X)	CFP Total
RF01	Iniciar sesión	1	1	0	1	3
RF02	Perfil de usuario	1	1	1	1	4
RF03	Registrar estudiantes	1	0	1	1	3
RF04	Editar estudiantes	1	1	1	1	4
RF05	Eliminar estudiantes	1	1	1	1	4
RF06	Visualizar datos del estudiante	1	1	0	1	3
RF07	Registrar personal institucional	1	0	1	1	3
RF08	Editar personal institucional	1	1	1	1	4
RF09	Eliminar personal institucional	1	1	1	1	4
RF10	Visualizar datos del personal institucional	1	1	0	1	3
RF11	Permisos de acceso	1	1	1	1	4
RF12	Entrenar la red neuronal	1	1	1	1	4
RF13	Visualizar áreas disponibles del Internado Rotatorio	1	1	0	1	3

RF14	Visualizar tema de estudio	1	1	0	1	3
RF15	Generar cuestionarios	1	1	1	1	4
RF16	Evaluar cuestionarios	1	1	1	1	4
RF17	Interactuar con el modelo	1	1	0	1	3
RF18	Generación de audio	1	0	0	1	2
RF19	Visualizar el progreso del estudiante	1	1	0	1	3
RF20	Almacenamiento de progresos	1	0	1	0	2
Total						67

4.3.3.3. Medición

La suma total de Cosmic Function Points (CFP) para todos los requerimientos funcionales es:

- CFP Total = 67

Ecuación 3

Ecuación de esfuerzo

$$E = t * te$$

Esfuerzo (H) = Tamaño (CFP) x Tasa de Entrega (H/CFP)

Ecuación 4

Ecuación de costos

$$C = t * cu$$

Costo Total (\$) = Tamaño (CFP) x Costo por unidades (\$ /CFP)

Parámetros utilizados:

- Tasa de entrega: 10 horas/CFP (cantidad de horas necesarias para desarrollar cada punto de función).
- Costo por unidad: USD 50/CFP (costo asociado a cada Cosmic Function Point).

Cálculos:

- Cálculo del esfuerzo total (H):

$$\text{Esfuerzo (H)} = 67 \text{ (CFP)} \times 4 \text{ (H/CFP)} = 670 \text{ H.}$$

- Cálculo del costo total (\$):

$$\text{Costo Total (\$)} = 67 \text{ (CFP)} \times 50 \text{ (USD/CFP)} = 3350 \text{ USD}$$

Resultados finales:

Esfuerzo total: 670 horas.

Costo total en Bs: 23316 Bs (Tipo de cambio oficial 6.96 Bs x USD)

4.4. SEGURIDAD DEL SOFTWARE

La implementación de la ISO/IEC 27002 en seguridad de software se enfoca en proteger la información relacionada con el desarrollo y operación del sistema tutor inteligente.

4.4.1. Control de Acceso

Proteger los recursos del sistema tutor inteligente contra accesos no autorizados.

- Autenticación estricta: Todo usuario (estudiantes y personal institucional) deberá acceder al sistema mediante credenciales únicas y seguras, utilizando autenticación mediante tokens.
- Gestión de permisos: Se aplicarán permisos según el tipo de usuario, asegurando que cada usuario solo pueda acceder a las áreas y funciones necesarias.

4.4.2. Seguridad en Procesos de Desarrollo

Garantizar que el tutor inteligente esté libre de vulnerabilidades explotables antes de su implementación.

- Pruebas: Antes del despliegue, se realizarán pruebas para identificar posibles vulnerabilidades en las funciones críticas del sistema, como la autenticación, la gestión de datos sensibles y la generación de contenido.
- Seguridad en el entrenamiento: Validar que los datos recolectados sean libres de elementos corruptos o maliciosos.

CAPÍTULO V

CONCLUSIONES Y

RECOMENDACIONES

**INGENIERÍA
DE SISTEMAS**
UNIVERSIDAD PÚBLICA DE EL ALTO



CAPÍTULO V

5. CONCLUSIONES Y RECOMENDACIONES

5.1. CONCLUSIONES

A continuación se describen las conclusiones de los objetivos específicos:

- Se realizaron reuniones con profesionales del área de la salud que colaboraron en la identificación y selección de textos y libros relevantes para cada área del Internado Rotatorio, como Cirugía, Pediatría, Medicina Interna, Ginecología Obstetricia y Salud Pública. Estos textos fueron clasificados por áreas temáticas y se almacenaron en la base de datos.
- Se preprocesaron los textos seleccionados, eliminando caracteres especiales, referencias no relevantes y secciones irrelevantes. Además, se estandarizó un formato adecuado posteriormente se realizó el entrenamiento de la red neuronal.
- Se diseñó una estructura de navegación intuitiva con un menú jerárquico para acceder fácilmente a las diferentes áreas y temas del Internado Rotatorio, por lo cual se llegó a visualizar los temas de forma organizada. Se implementaron filtros para facilitar la búsqueda de contenido específico, y la interfaz de visualización de contenido fue diseñada, el cual permitió al estudiante a mejorar la retención de conceptos de medicina.
- Se creó una interfaz donde se puede evaluar al estudiante mediante cuestionarios basados en el contenido estudiado por el estudiante. Además, se proporcionó retroalimentación automatizada para explicar las respuestas correctas e incorrectas, lo cual permitió fortalecer los conocimientos del estudiante, así como también se registró el progreso del estudiante en una base de datos.
- Se desarrolló una interfaz de chat para permitir la interacción entre tutor y estudiante, y se generó audio por cada mensaje enviando por el tutor.

A lo largo del proyecto se logró cumplir con los objetivos específicos, entonces como resultado se logró desarrollar el Sistema Tutor Inteligente Basado en Redes Neuronales Artificiales el cual ha permitido mejorar el logro de competencias en el Internado Rotatorio.

5.2. RECOMENDACIONES

- A medida que el sistema se consolide, sería beneficioso expandir el contenido a más áreas del conocimiento médico, incluyendo especialidades emergentes o de alta demanda, para que el tutor sea aún más completo y útil para los estudiantes en diversas ramas de la medicina.
- Se recomienda seguir perfeccionando las técnicas de preprocesamiento de datos para minimizar el ruido y garantizar que el modelo de red neuronal se entrene con información lo más precisa y clara posible. Esto puede incluir el uso de métodos avanzados de filtrado y análisis de textos médicos.
- Aunque el tutor inteligente ya ofrece interacciones con el estudiante, sería valioso agregar funcionalidades adicionales como simulaciones de escenarios médicos o la capacidad de realizar diagnósticos interactivos para un aprendizaje más práctico. Esta expansión podría mejorar aún más la experiencia del estudiante.

REFERENCIAS

**INGENIERÍA
DE SISTEMAS**
UNIVERSIDAD PÚBLICA DE EL ALTO



REFERENCIAS

- Abud, A. (2009). MeISE: Metodología de Ingeniería de Software Educativo. Revista Internacional de Educación en Ingeniería, 2.
https://www.academia.edu/6834077/P_%C3%A1_g_i_n_a_MeISE_Metodolog%C3%ADa_de_Ingenier%C3%ADa_de_Software_Educativo
- Alexandra, E., Leonel, I., Janellys, I., & Alberto, J. (199d. C.). Inteligencia Artificial Aplicada con técnicas de Procesamiento de Lenguaje Natural y Machine Learning en el campo de la salud Autor/es.
- Amaratunga, T. (2023). Understanding Large Language Models. En Understanding Large Language Models. Apress. <https://doi.org/10.1007/979-8-8688-0017-7>
- Anderson, D., & Mcneill, G. (1992). Artificial Neural Networks.
- Antonieta Abud Figueroa, M. (2003). Calidad en la Industria del Software. La Norma ISO-9126. <http://www.monografias.com/trabajos5/>
- Ayyadevara, V. Kishore. (2019). Neural networks with Keras cookbook : over 70 recipes leveraging deep learning. Packt Publishing.
- Baum, D. (2023). Generative AI and LLMs.
- Bordóns, C., Ruiz, A. M., Daniel, A., Marruedo, L., Manuel, A., Arahál, R., & Limón Marruedo, D. (2001). Teoría de Sistemas.
- Callejas-Cuervo, M., Alarcón-Aldana, A. C., & Álvarez-Carreño, A. M. (2017). Modelos de calidad del software, un estado del arte. ENTRAMADO, 13(1), 236–250.
<https://doi.org/10.18041/entramado.2017v13n1.25125>
- Carlos Javier Mosquera Rector Roberto Ferro Escobar, D. S., Enrique Montenegro Marín, C., Aristides Gonzalez Crespo, R., Sanjuán, O., Mauricio Tarazona Bermude Universidad Distrital FJC, G., Sandra Yanet Velasco, C., Manuel Cueva Lovelle, J., García Díaz, V., Ramos, R., Moreno, W., Mercedes Larrondo, M., Ting, I.-H., Gaona, P., Bello, R.,

- Burgos, D., Pascual, J., Pelayo, C., Pérez, M., Joyanes, L., ... Solano Suárez Msc, A. (2015). Redes de Ingeniería.
- Carrizo, D., & Alfaro, A. (2018). Methodology: a practice approach. En Revista chilena de ingeniería (Vol. 26, Número 1).
- Cataldi, Z., Calvo, P., Salgueiro, F. A., & Lage, F. J. (2007). Diseño de Sistemas Tutores Inteligentes con Tecnología de Agentes: Los Agentes Docentes en el Módulo Tutor.
- Cataldi, Z., & Lage, F. (1997). El problema del modelado del estudiante en Sistemas Tutores Inteligentes.
- Cataldi, Z., Salgueiro, F., & Lage, F. J. (1998). Fundamentos para el Submódulo Evaluador en Sistemas Tutores Inteligentes: Diagnóstico, predicción y autoevaluación.
- César, J., & González, S. (2000). Métodos de Evaluación de Usabilidad para Sistemas de Información Web: Una revisión.
- Cordero Vargasm, I. (2004). Control de calidad en software - Material complementario.
- Cruz, S., González, T., & Cañizal, A. (2007). Gestión de la calidad : conceptos, enfoques, modelos y sistemas. Pearson Educación : Prentice Hall.
- Cuomo, V., & Castares, M. (1997). ISO 9126-ISO 14598 Calidad de Software. eBook. (2019). Django Rest Framework. <https://github.com/YeasirArafatRatul/DjangoBooks/blob/master/django-rest-framework.pdf>
- Eisenstein, J. (2018). Natural Language Processing.
- Elizabeth, P., María, A. B., Polesello, F., Macarena, M., Mendoza, S., Troila, J. A., Alejandra, M., & De Guerrero, M. (2001). Coso I y Coso II.
- En, A., Educación, L. A., Durango Hernández, J. A., Smith, Y., & Rengifo, P. (2007). Los sistemas tutores inteligentes.
- Euclides, G., Fernando, B., Gissel, K., & Enrique, B. (2006). La Revolución de la Inteligencia.

- Flores Mamani, E. R. (2022). Tutor Inteligente Para El Apoyo Del Aprendizaje En La Lectura De 1° De Primaria Comunitaria Vocacional De La Unidad Educativa Tecnico Humanistico "Don Bosco".
- Gómez, A., del CLópez, M., Migani, S., & Otazú, A. (2001). Cocomo.
- González-González, C. S. (2023). El impacto de la inteligencia artificial en la educación: transformación de la forma de enseñar y de aprender. *Qurriculum. Revista de Teoría, Investigación y Práctica educativa*, 36, 51–60.
<https://doi.org/10.25145/j.qurricul.2023.36.03>
- Grau, X. F., Isabel, M., & Segura, S. (s. f.). Desarrollo Orientado a Objetos con UML.
- Haykin, S. S. . (2009). *Neural networks and learning machines*. Prentice Hall/Pearson.
- Heikkinen, D., Yiu, J., & Lam, A. (2016). *Deep Learning with Python, Second Edition*.
- itmentor. (2000). Pruebas de software.
- Johansen Bertoglio, O. (2010). *Introducción A La Teoria General De Sistemas*.
- Kendall, K. E. ., & Kendall, J. E. . (2011). *Systems analysis and design*. Pearson Prentice Hall.
- López Guillén, J. (2020). Diseño e Implementación de una funcionalidad de Transfer Learning en un modelo de Aprendizaje por Refuerzo Aplicado a la navegación con Drones. Universidad Politécnica de Madrid.
- María P. Bonomini, Eduardo Fernández, & José M. Ferrández. (2011). *Artificial Neural Networks and Retinal Ganglion Cell Responses*. INTECH Open Access Publisher.
- Mariuxi, Lady, Javier, R., Leonel, I., & Alexandra, E. (1998). *Inteligencia Artificial Aplicada a Procesamiento de Lenguaje Natural (NLP) con Python y Machine Learning*. Autor/es.
- Mendoza Apaza, J. (2020). Tutor Inteligente Móvil Para La Enseñanza De La Estática Y Dinámica En Estudiantes De Quinto De Secundaria. UNIVERSIDAD PÚBLICA DE EL ALTO.

- Mera Paz, J. A., Miranda Gómez, M. Y., & Cuaran Rosas, S. (2017). Análisis sistemático de información de la Norma ISO 25010.
https://www.researchgate.net/publication/360017691_Systematic_analysis_of_ISO_25010_information_as_a_basis_for_implementation_in_a_software_testing_laboratory_at_the_Cooperative_University_of_Colombia_Headquarters_Popayan
- Molina Ríos, J., Loja Mora, M., & Ordóñez, Z. (2016). Evaluación de los Frameworks en el Desarrollo de Aplicaciones Web con Python. En Revista Latinoamericana de Ingeniería de Software (Vol. 4, Número 4). <https://revistas.unla.edu.ar/software/article/view/1149>
- Mora Contreras, C. E. (2011). La Calidad del Servicio y la Satisfacción del Consumidor. Revista Brasileira de Marketing, 10(2), 146–162.
<https://doi.org/10.5585/remark.v10i2.2212>
- Moreno Capuchino, A. (2005). Estimación de Proyectos Software.
- Mulder, V., Mermoud, A., & Lenders, V. (2024). Large Language Models in Cybersecurity.
- Pai, Suhas. (2025). Large Language Model Applications. O'REILLY MEDIA.
- Peño Sánchez, J. (2015). Pruebas de Software Fundamentos y Técnicas.
https://oa.upm.es/40012/1/PFC_JOSE_MANUEL_SANCHEZ_PENO_3.pdf
- Polo Usaola, M. (2008). Pruebas de sistemas de información.
- Pradel, J., Jose, M., & Martos, R. (1995). Introducción a la ingeniería del software.
- Pressman, R. S. (2010). Ingeniería del software. Un enfoque practico (Séptima Edición).
- Raschka, S. (2016). Machine Learning Q and AI Expand Your Machine Learning & AI Knowledge With 30 In-Depth Questions and Answers. <http://leanpub.com/machine-learning-q-and-ai>
- Raschka, S. (2024). Build a Large Language Model (From Scratch).
- Raymond S. T., L. (2024). Natural Language Processing A Textbook with Python Implementation. <https://doi.org/10.1007/978-981-99-1999-4>

- Rodríguez, O. H. (2000). Descripción breve El documento incluye una descripción de la
Arquitectura de Software para el sistema OpenCNC DOCUMENTO DE
ARQUITECTURA DE SOFTWARE.
- Rolando, J., & Ríos, M. (2019). Swirl.
- Romero, J., Dafonte, C., Gómez, Á., & Penousal, F. J. (2007). Inteligencia Artificial y
Computación Avanzada.
- Ruiz. (2006). Modelo de evaluación.
- Russell, S., & Norvig, P. (2004). Inteligencia artificial. El Cid Editor.
- Sanchez, J. A. (2018). Desarrolla aplicaciones con Vuejs.
https://www.academia.edu/36309943/Desarrolla_aplicaciones_con_vuejs
- Sanz Torres, I. (2023). Detección de Contenido Sexual mediante Aprendizaje Profundo y
Aprendizaje por Transferencia. Universidad Complutense de Madrid.
- Sarkar, D. (2019). Text Analytics with Python. En Text Analytics with Python. Apress.
<https://doi.org/10.1007/978-1-4842-4354-1>
- Sommerville, Ian., & Domínguez Torres, J. Alejandro. (2011). Ingeniería de software.
Addison-Wesley, Pearson Educación.
- Stevens, E., Antiga, L., & Viehmann, T. (2000). Deep Learning with PyTorch.
https://isip.piconepress.com/courses/temple/ece_4822/resources/books/Deep-Learning-with-PyTorch.pdf
- Suzuki, K. (2011). Artificial Neural Networks - Methodological Advances and Biomedical
Applications. INTECH Open Access Publisher.
- Tello, E. A. R. (2007). Estrategias y técnicas de prueba del software.
- Toro, M. I. S. (2000). Calidad de Software (ISO/IEC 9126-25000).
- Tuya, J. (2003). Pruebas del Software: Descubrir Errores y Más.
<http://www.di.uniovi.es/~tuya/>

Wirth, R., & Hipp. (2000). Metodología CRISP-DM.

<https://es.scribd.com/document/698683829/Metodologia-CRISP-DM>

Yang, Q. (2020). Transfer Learning.

Yourdon. (1993). Análisis y Diseño Estructurado Moderno.

Zopp, M. (1999). Método Zoop.

globalsuitesolutions. (2024). globalsuitesolutions. Obtenido de globalsuitesolutions:

<https://www.globalsuitesolutions.com/es/que-es-la-norma-iso-27002-y-para-que-sirve/>

Invensity. (2024). Invensity. Obtenido de Invensity:

<https://www.invensity.com/consultoria/ingenieria-de-software/garantia-de-calidad-del-software/?lang=es>

ANEXOS

**INGENIERÍA
DE SISTEMAS**
UNIVERSIDAD PÚBLICA DE EL ALTO



ANEXO A

**INGENIERÍA
DE SISTEMAS**
UNIVERSIDAD PÚBLICA DE EL ALTO



ANEXO A: MATRIZ DE PLANIFICACIÓN DE PROYECTO

Estrategias del proyecto	Indicadores	Fuentes de verificación	Supuestos
Objetivo general			
<p>Desarrollar un Tutor Inteligente Basado en Redes Neuronales Artificiales para mejorar el logro Competencias en el Internado Rotatorio, en la carrera de Medicina de la Universidad Pública de El Alto.</p>	<p>Se logró el desarrollar el sistema tutor inteligente, alcanzando el 65% de los objetivos de funcionalidad.</p>	<p>Manual de usuario del sistema.</p>	<p>Disponibilidad de recursos humanos capacitados en medicina.</p>
Objetivos específicos			
<p>Seleccionar textos y libros de los temas relevantes del Internado Rotatorio según sus áreas: Cirugía, Pediatría, Medicina Interna, Ginecología Obstetricia y Salud Publica en colaboración con profesionales de medicina para crear una base de datos con material a disposición para el estudiante.</p>	<p>Se logró la selección y organización de textos en un 60%</p>	<p>Listado de documentos en la base de datos.</p>	<p>Apoyo del Dr. Luis flores profesional del área de medicina.</p>
<p>Eliminar el ruido de los textos seleccionados para realizar el entrenamiento de la red neuronal.</p>	<p>Se logró una reducción del ruido textual del 60%.</p>	<p>Los textos procesados se encuentran en un DataSet.</p>	<p>Herramientas adecuadas de procesamiento de texto, como Google Colab Pro.</p>

Visualizar distintos temas de manera organizada para facilitar al estudiante el acceso rápido, facilitando así la retención y comprensión de los conceptos clave.	Se organizo los temas, permitiendo un acceso rápido en un 68%.	Pruebas de acceso a la plataforma por estudiantes.	Infraestructura de soporte.
Evaluar al estudiante mediante un cuestionario al finalizar la lectura o estudio de un determinado tema para fortalecer los conocimientos del mismo.	Se logró evaluar al estudiante, alcanzando un 60%.	Registros de cuestionarios y resultados de evaluación.	Participación activa del estudiante gracias a la interacción de la red neuronal.
Disponer de un tutor mediante un modelo de procesamiento de lenguaje natural para interactuar con el estudiante en tiempo real.	Se puso a disposición un modelo de procesamiento de lenguaje natural para la interacción, cubriendo el 70% las necesidades del estudiante.	Satisfacción del estudiante.	Acceso al modelo de procesamiento de lenguaje natural.
Resultados			
Base de datos estructurada con material de estudio en medicina.	55%	Manual técnico.	Selección de textos y libros de medicina.
Textos sin ruido para entrenamiento de la red neuronal.	61%	Manual técnico.	Uso de Google Colab Pro.
Interfaz de usuario accesible y funcional para facilitar la consulta.	68%	Pruebas de funcionalidad.	Disponibilidad del programador.

Cuestionarios disponibles para evaluar el conocimiento en cada tema.	70%	Progreso del estudiante registrados en la base de datos.	Disponibilidad de la red neuronal (modelo)
Módulo TutorAI que permite la interacción en tiempo real con estudiantes.	70%	Manual de usuario.	Uso de Google Colab Pro
Actividades			
Organizar reuniones con profesionales para la selección de material de estudio.	Presupuestos de hardware: 1. Memoria: 0Bs 2. Cámara: 0Bs	Presupuestos de software: Aplicaciones de play store sin costos.	Horas-hombre: Aproximadamente 1 hora por sesión
Limpiar y preprocesar textos para eliminar ruido.	Presupuestos de hardware: 1. GPU: 400 Bs 2. CPU: 0 Bs	Presupuestos de software: Python: 0 Bs JavaScript: 0 Bs Google Colab Pro: 400 Bs Vue: 0 Bs Django: 0 Bs CSS: 0 Bs	Horas-hombre: Tiempo del desarrollador
Organizar temas y estructurar el acceso a contenidos en el sistema.	Presupuestos de hardware: 1. CPU: 0 Bs	Presupuestos de software: Python: 0 Bs JavaScript: 0 Bs Google Colab Pro: 400 Bs Vue: 0 Bs Django: 0 Bs CSS: 0 Bs	Horas-hombre: Tiempo del desarrollador

<p>Crear cuestionarios de evaluación para cada tema.</p>	<p>Presupuestos de hardware:</p> <ol style="list-style-type: none"> 1. CPU: 0 Bs 2. GPU: 400 Bs 	<p>Presupuestos de software:</p> <p>Python: 0 Bs JavaScript: 0 Bs Google Colab Pro: 400 Bs Vue: 0 Bs Django:0 Bs CSS: 0 Bs</p>	<p>Horas-hombre:</p> <p>Tiempo del desarrollador</p>
<p>Entrenar el modelo de tutoría para interacción en tiempo real con estudiantes.</p>	<p>Presupuestos de hardware:</p> <p>CPU: 0 Bs GPU: 400 Bs</p>	<p>Presupuestos de software:</p> <p>Python: 0 Bs JavaScript: 0 Bs Google Colab Pro: 400 Bs Vue: 0 Bs Django:0 Bs CSS: 0 Bs</p>	<p>Horas-hombre:</p> <p>Tiempo del desarrollador</p>

ANEXO B

**INGENIERÍA
DE SISTEMAS**
UNIVERSIDAD PÚBLICA DE EL ALTO



ANEXO B: DIAGRAMA DE FLUJO DEL PROBLEMA PRINCIPAL

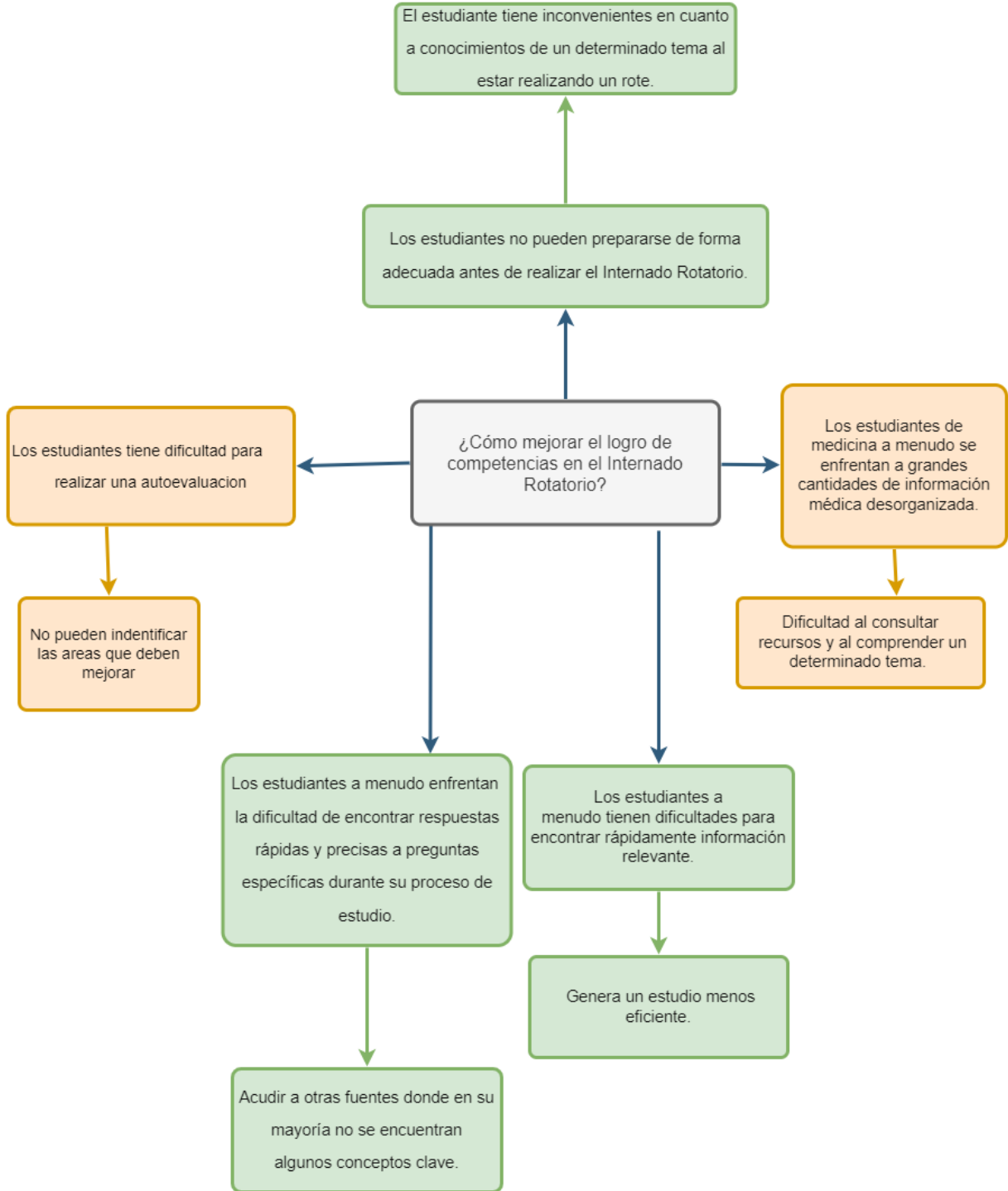


ANEXO C

**INGENIERÍA
DE SISTEMAS**
UNIVERSIDAD PÚBLICA DE EL ALTO



ANEXO C: ARBOL DE PROBLEMAS

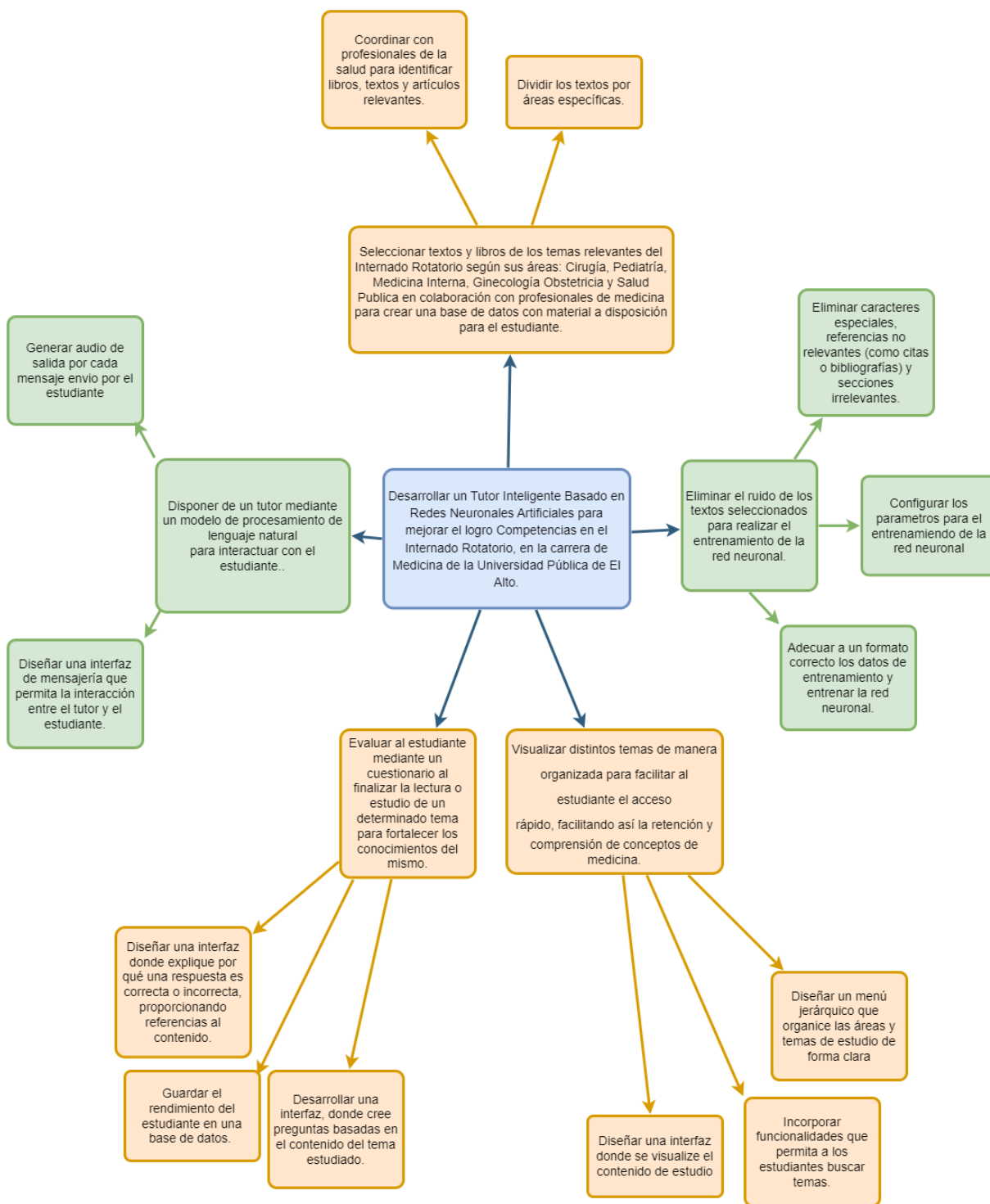


ANEXO D

**INGENIERÍA
DE SISTEMAS**
UNIVERSIDAD PÚBLICA DE EL ALTO



ANEXO D: ÁRBOL DE OBJETIVOS

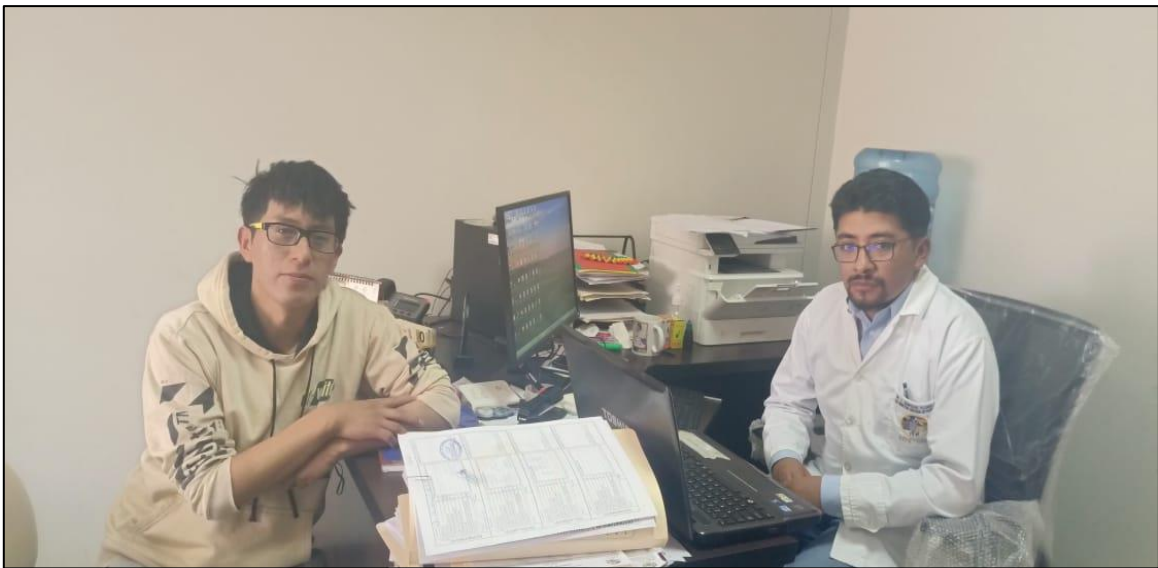


ANEXO E

**INGENIERÍA
DE SISTEMAS**
UNIVERSIDAD PÚBLICA DE EL ALTO



ANEXO E: REUNIONES

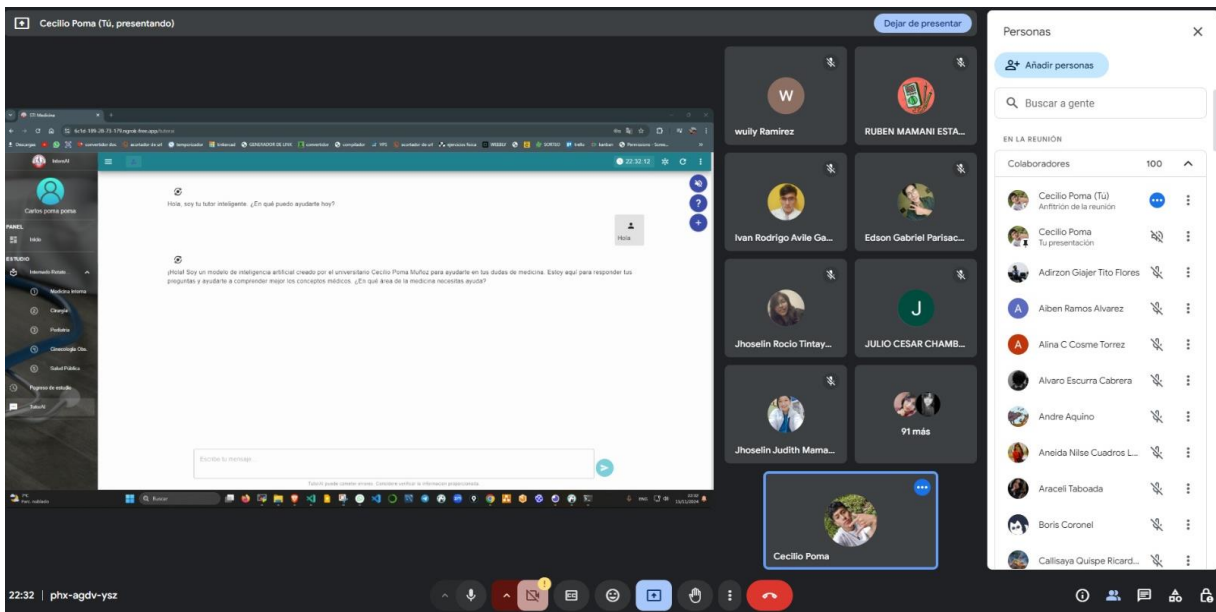


ANEXO F

**INGENIERÍA
DE SISTEMAS**
UNIVERSIDAD PÚBLICA DE EL ALTO



ANEXO F: USO DEL SISTEMA



ANEXO G

**INGENIERÍA
DE SISTEMAS**
UNIVERSIDAD PÚBLICA DE EL ALTO



ANEXO G: RESULTADO DE LAS ENCUESTAS



Usabilidad del sistema InternAI

Sistema Tutor Inteligente basado en redes neuronales artificiales para mejorar el logro de competencias en el Internado Rotatorio

Debe ingresar al sistema y verificar cada funcionalidad (opciones).

[Iniciar sesión en Google](#) para guardar lo que llevas hecho. [Más información](#)

* Indica que la pregunta es obligatoria

¿El sistema responde en menos de 2 segundos al realizar consultas generales? *

- Sí
- No

¿El tiempo de respuesta al generar cuestionarios es menor a 6 segundos? *

- Sí
- No

¿El TutorAI responde de forma eficiente en tareas complejas? *

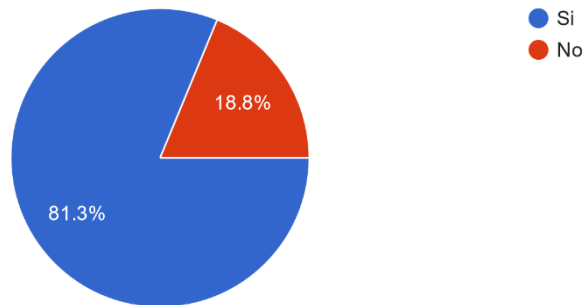
- Sí
- No

¿Experimentó tiempos de espera superiores a 7 segundos? *

- Sí
- No

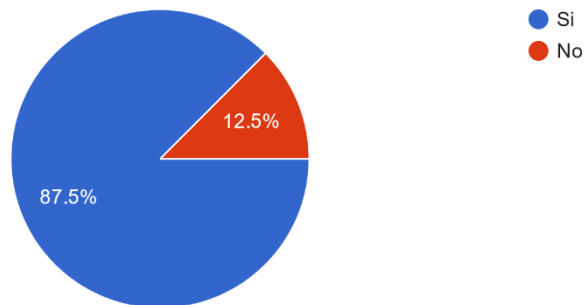
¿El sistema responde en menos de 2 segundos al realizar consultas generales?

16 respuestas



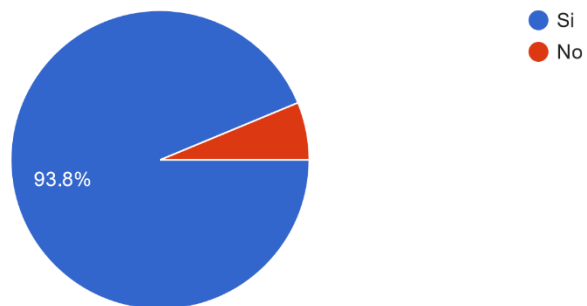
¿El tiempo de respuesta al generar cuestionarios es menor a 6 segundos?

16 respuestas



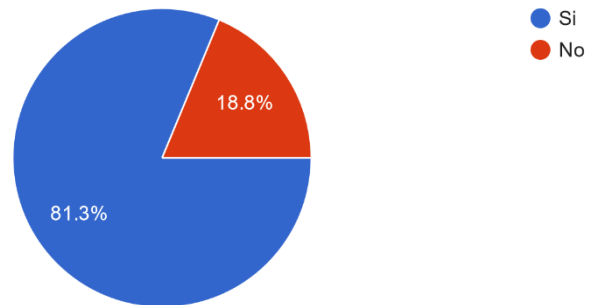
¿El sistema tarda menos de 2 segundos en cargar las áreas de estudio?

16 respuestas



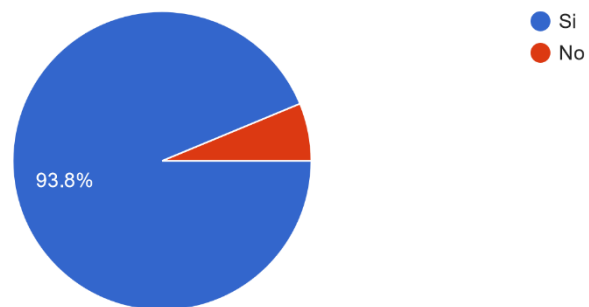
¿Pudo completar las tareas asignadas (estudio, cuestionarios, TutorAI) sin ayuda externa?

16 respuestas



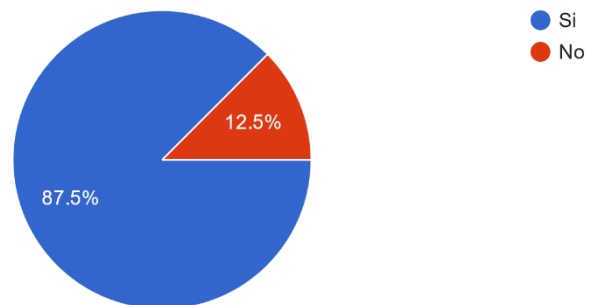
¿Encontró el sistema intuitivo para navegar y realizar actividades?

16 respuestas



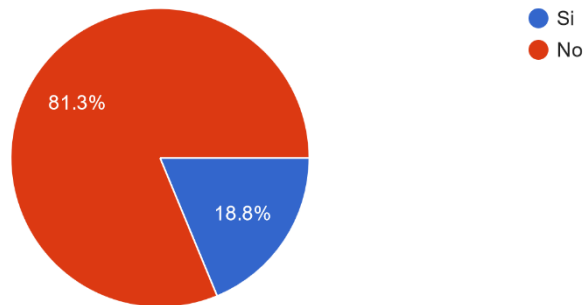
¿Pudo acceder fácilmente a las áreas de estudio?

16 respuestas



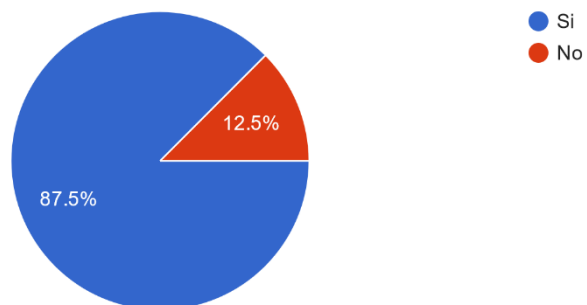
¿El sistema presentó fallos al interactuar con las áreas de estudio?

16 respuestas



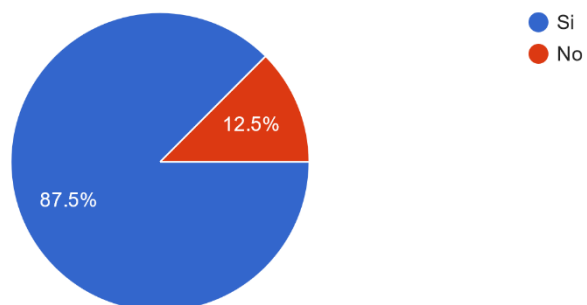
¿Las respuestas del TutorAI fueron precisas y sin errores evidentes?

16 respuestas



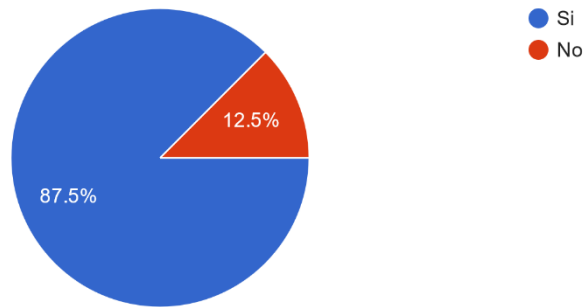
¿Considera que las funcionalidades del sistema están completas y operativas?

16 respuestas



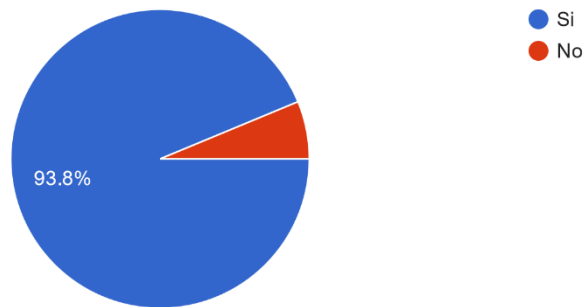
¿La gestión de áreas y temas cumplió con lo requerido para su estudio?

16 respuestas



¿Las funcionalidades ofrecidas facilitaron su aprendizaje?

16 respuestas



ANEXO H

**INGENIERÍA
DE SISTEMAS**
UNIVERSIDAD PÚBLICA DE EL ALTO



**ANEXO H: CUESTIONARIO PARA EVALUAR EL MEJORAMIENTO DEL ESTUDIANTE
EN EL INTERNADO TRAS USAR EL SISTEMA**

Internado Rotatorio

Debe llenar el formulario, en base a la forma como le ayudo el sistema, en el Internado Rotatorio

[Acceder a Google](#) para guardar el progreso. [Más información](#)

* Indica que la pregunta es obligatoria

Nombre completo

Tu respuesta _____

¿En qué área del internado estás actualmente? *

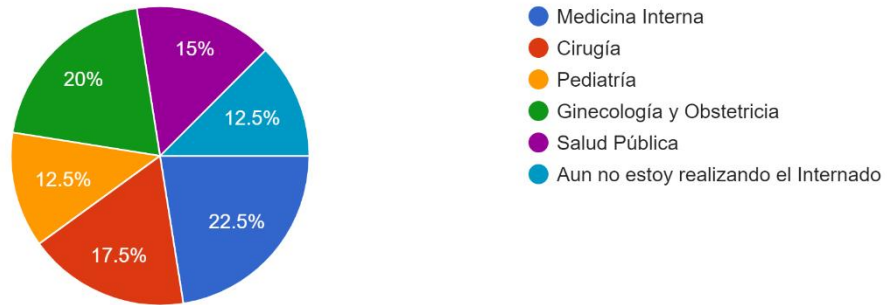
- Medicina Interna
- Cirugía
- Pediatría
- Ginecología y Obstetricia
- Salud Pública

¿Has utilizado anteriormente herramientas de tutoría inteligente para tu aprendizaje? *

- Si
- No

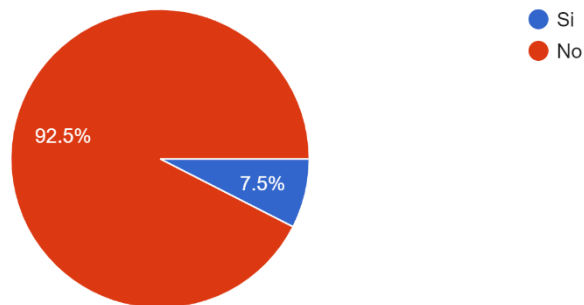
¿En qué área del internado estás actualmente?

40 respuestas



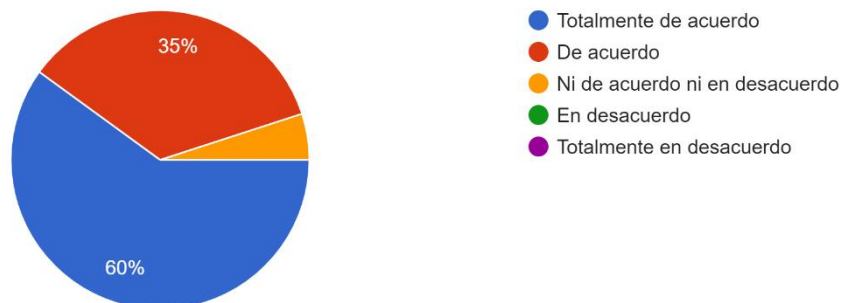
¿Has utilizado anteriormente herramientas de tutoría inteligente para tu aprendizaje?

40 respuestas



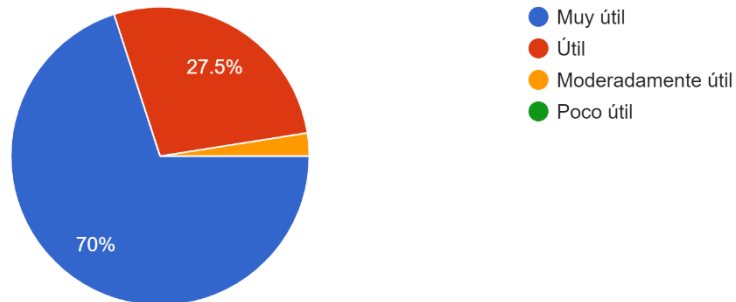
¿El sistema tutor inteligente te ha ayudado a comprender mejor los temas del internado?

40 respuestas



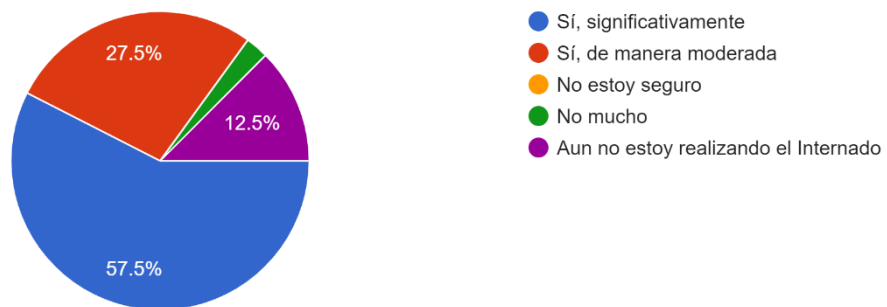
¿Qué tan útil consideras que el sistema tutor inteligente es para tu aprendizaje en las distintas áreas del internado rotatorio?

40 respuestas



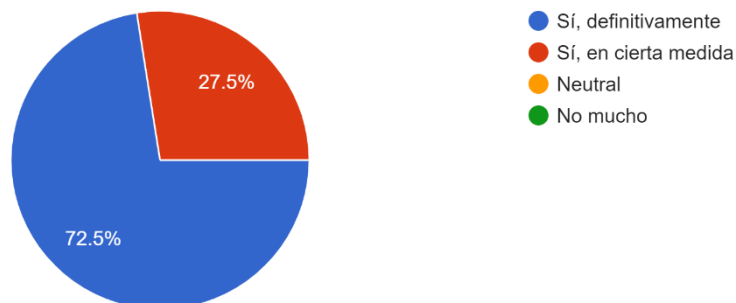
¿Consideras que el sistema ha mejorado tus habilidades prácticas en el internado?

40 respuestas



¿Crees que el tutor inteligente contribuye a tu preparación para los exámenes y evaluaciones en las áreas del internado?

40 respuestas



AVALES

**INGENIERÍA
DE SISTEMAS**
UNIVERSIDAD PÚBLICA DE EL ALTO



AVAL DE CONFORMIDAD

(TUTOR METODOLÓGICO)

El Alto, noviembre de 2024

Señor:
Ing. William Roque Roque
DIRECTOR CARRERA INGENIERÍA DE SISTEMAS
UNIVERSIDAD PÚBLICA DE EL ALTO
Presente. –

REF. AVAL DE CONFORMIDAD

Distinguido director:

Mediante la presente tengo a bien comunicarle mi conformidad del Trabajo de Grado:

TITULO: TUTOR INTELIGENTE BASADO EN REDES NEURONALES ARTIFICIALES PARA MEJORAR EL LOGRO DE COMPETENCIAS EN EL INTERNADO ROTATORIO

CASO: CARRERA DE MEDICINA DE LA UNIVERSIDAD PÚBLICA DE EL ALTO

MODALIDAD: PROYECTO DE GRADO

UNIVERSITARIO: Cecilio Poma Muñoz

REGISTRO UNIVERSITARIO: 200029033

CEDULA DE IDENTIDAD: 7339585 OR

Para que se le otorgue fecha y hora de su defensa pública y evaluación correspondiente a la materia de Taller de Grado II, de acuerdo al Reglamento Vigente de la Carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

Atentamente,


.....
M.Sc. Lic. Ing. Mario Arguedas Balladares
TUTOR METODOLÓGICO

AVAL DE CONFORMIDAD

(TUTOR REVISOR)

El Alto, noviembre de 2024

Señora:
M.Sc. Marisol Arguedas Balladares
TUTOR METODOLÓGICO
TALLER DE GRADO II
Presente. –

REF. AVAL DE CONFORMIDAD

Distinguido tutor metodológico:
Mediante la presente tengo a bien comunicarle mi conformidad del Trabajo de Grado:

**TITULO: TUTOR INTELIGENTE BASADO EN REDES NEURONALES
ARTIFICIALES PARA MEJORAR EL LOGRO DE COMPETENCIAS EN EL
INTERNADO ROTATORIO**

CASO: CARRERA DE MEDICINA DE LA UNIVERSIDAD PÚBLICA DE EL ALTO

MODALIDAD: PROYECTO DE GRADO


Universitario: Cecilio Poma Muñoz

Registro Universitario: 200029033

Cedula de Identidad: 7339585 OR

Para su defensa pública y evaluación correspondiente a la materia de Taller de Grado II, de acuerdo al Reglamento Vigente de la Carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

Atentamente,


.....
Lic. Ing. Miguel Ángel Pérez Bustillos
TUTOR REVISOR

AVAL DE CONFORMIDAD

(TUTOR ESPECIALISTA)

El Alto, noviembre de 2024

Señora:
M.Sc. Marisol Arguedas Balladares
TUTOR METODOLÓGICO
TALLER DE GRADO II
Presente. –

REF. AVAL DE CONFORMIDAD

Distinguido tutor metodológico:
Mediante la presente tengo a bien comunicarle mi conformidad del Trabajo de Grado:

**TITULO: TUTOR INTELIGENTE BASADO EN REDES NEURONALES
ARTIFICIALES PARA MEJORAR EL LOGRO DE COMPETENCIAS EN EL
INTERNADO ROTATORIO**
CASO: CARRERA DE MEDICINA DE LA UNIVERSIDAD PÚBLICA DE EL ALTO
MODALIDAD: PROYECTO DE GRADO
Universitario: **Cecilio Poma Muñoz**
Registro Universitario: **200029033**
Cedula de Identidad: **7339585 OR**

Para su defensa pública y evaluación correspondiente a la materia de Taller de Grado II, de acuerdo al Reglamento Vigente de la Carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

Atentamente.


.....
Lic. Margarita Bernarda Lopez Mariaca
TUTOR ESPECIALISTA



Universidad Pública de El Alto

Creada por Ley 2115 del 5 de septiembre de 2000 y Autónoma por Ley 2556 del 12 de noviembre de 2003



DECANATO DEL ÁREA CIENCIAS DE LA SALUD

NOTA INTERNA
UPEA-DCS-N° 215/2024
El Alto 08 de noviembre del 2024

Señora:

M. Sc. Lic. Ing. Marisol Arguedas Balladares
TUTOR METODOLÓGICO
TALLER DE GRADO II
CARRERA DE INGENIERIA DE SISTEMAS
UNIVERSIDAD PÚBLICA DE EL ALTO

PRESENTE

REF. AVAL DE CONFORMIDAD

Distinguido tutor metodológico:

Mediante la presente tengo a bien de comunicarle mi conformidad del Trabajo de Grado.

**TITULO: TUTOR INTELIGENTE BASADO EN REDES NEURONALES ARTIFICIALES
PARA MEJORAR EL LOGRO DE COMPETENCIAS EN EL INTERNADO ROTATORIO**

CASO: CARRERA DE MEDICINA DE LA UNIVERSIDAD PÚBLICA DE EL ALTO

MODALIDAD: PROYECTO DE GRADO

UNIVERSITARIO: CECILIO POMA MUÑOZ

REGISTRO UNIVERSITARIO: 200029033

CEDULA DE IDENTIDAD: 7339585 OR

De tal forma cabe recalcar que el proyecto satisface los requerimientos de la institución de esta forma se dio cumplimiento de los objetivos del presente.

El presente proyecto fue **IMPLEMENTADO** satisfactoriamente y se realizó las capacitaciones necesarias en la institución.

En cuanto certifico, en honor a la verdad, para fines consiguientes del interesado para su defensa pública y evaluación correspondiente a la materia de Taller de Grado II, de acuerdo con el reglamento vigente de la Carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

Agradecido de antemano su gentil atención, me despido muy atentamente.

c.c. Arch. Decanato
WGRCH/mepv

Dr. Wully Genaro Ramirez Chambi
DECANO
ÁREA CIENCIAS DE LA SALUD
UPEA



MANUAL DE USUARIO

**INGENIERÍA
DE SISTEMAS**
UNIVERSIDAD PÚBLICA DE EL ALTO





MANUAL DE USUARIO

SISTEMA TUTOR INTELIGENTE BASADO EN REDES NEURONALES ARTIFICIALES “INTERNAI”

Versión 1.0

2024



INDICE

1. INTRODUCCIÓN.....	3
2. REQUISITOS DE ACCESO.....	3
3. GUÍA DE USO GENERAL	3
3.1. NAVEGACIÓN POR EL SISTEMA:.....	3
4. MÓDULOS DEL SISTEMA	4
4.1. MÓDULO DE INICIO DE SESIÓN	4
4.2. MÓDULO DE PERFIL DE USUARIO	4
4.3. MÓDULO INTERNADO ROTATORIO.....	5
4.4. SUBMÓDULO DE ESTUDIO	6
4.5. MÓDULO DE PROGRESO DEL ESTUDIANTE	7
4.6. MÓDULO DE TUTORAI	9
4.7. MÓDULO DE GESTIÓN DE ESTUDIANTES	10
4.8. MÓDULO DE GESTIÓN DE PERSONAL INSTITUCIONAL.....	13
5. PREGUNTAS FRECUENTES	16



1. INTRODUCCIÓN

Este manual guía al usuario en el uso del sistema "Tutor Inteligente Basado en Redes Neuronales Artificiales" InternAI, que apoya el aprendizaje del internado rotatorio mediante módulos especializados para gestionar usuarios, temas de estudio, cuestionarios y la interacción con un tutor virtual basado en inteligencia artificial.

2. REQUISITOS DE ACCESO

- **Dispositivo:** Celular, Tablet o PC.
- **Navegador recomendado:** Google Chrome, Mozilla Firefox, Opera o Brave.
- **Credenciales:** Un usuario y contraseña válidos proporcionados por el administrador del sistema.
- **Dirección URL:**

3. GUÍA DE USO GENERAL

3.1. NAVEGACIÓN POR EL SISTEMA:

1. Al ingresar, el usuario verá un menú de navegación principal con las siguientes opciones:
 - Inicio
 - Internado Rotatorio (Ginecología, Cirugía, Pediatría, Medicina Interna y Salud Pública)
 - TutorAI
 - Progreso del Estudiante
 - Estudiante (Opción disponible solo para administradores del sistema)
 - Institucional (Opción disponible solo para administradores del sistema)
2. Cada sección tiene funciones específicas según los permisos del usuario (estudiante o personal institucional).
 - Estudiantes: Acceso a áreas de estudio, cuestionarios y TutorAI.



- Personal Institucional: Acceso a áreas de estudio, cuestionarios, TutorAI, Gestión de estudiantes y personal.

4. MÓDULOS DEL SISTEMA

4.1. MÓDULO DE INICIO DE SESIÓN

Objetivo: Permitir a los usuarios acceder al sistema de manera segura.

1. Instrucciones para Iniciar Sesión:

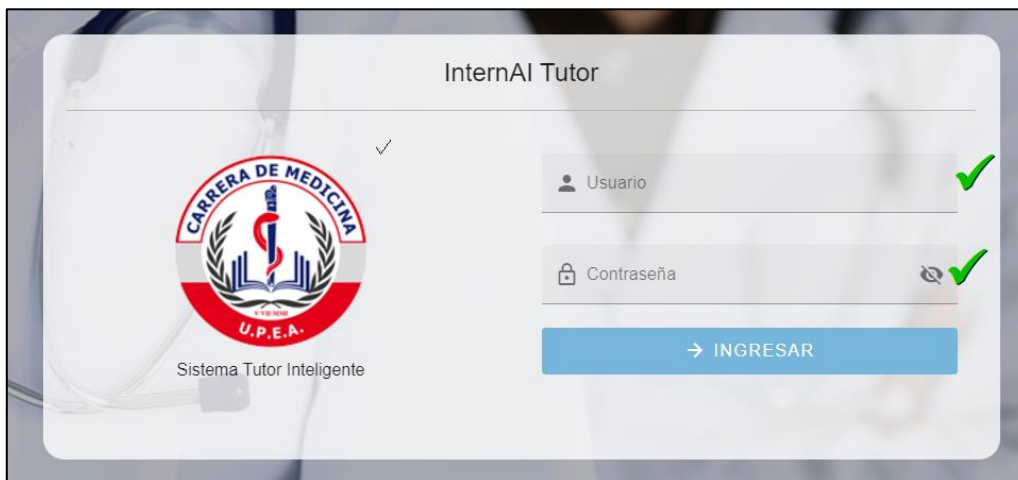
- Ingrese el nombre de usuario y contraseña en los campos indicados.
- Haga clic en el botón "Ingresar".

2. Olvido de Contraseña:

- Debe dirigirse o comunicarse con los administradores del sistema, para el restablecimiento de la contraseña.

3. Errores Comunes y Soluciones:

- Error de credenciales: Asegúrese de que su usuario y contraseña sean correctos.
- Cuenta deshabilitada: Contacte al administrador del sistema.



4.2. MÓDULO DE PERFIL DE USUARIO

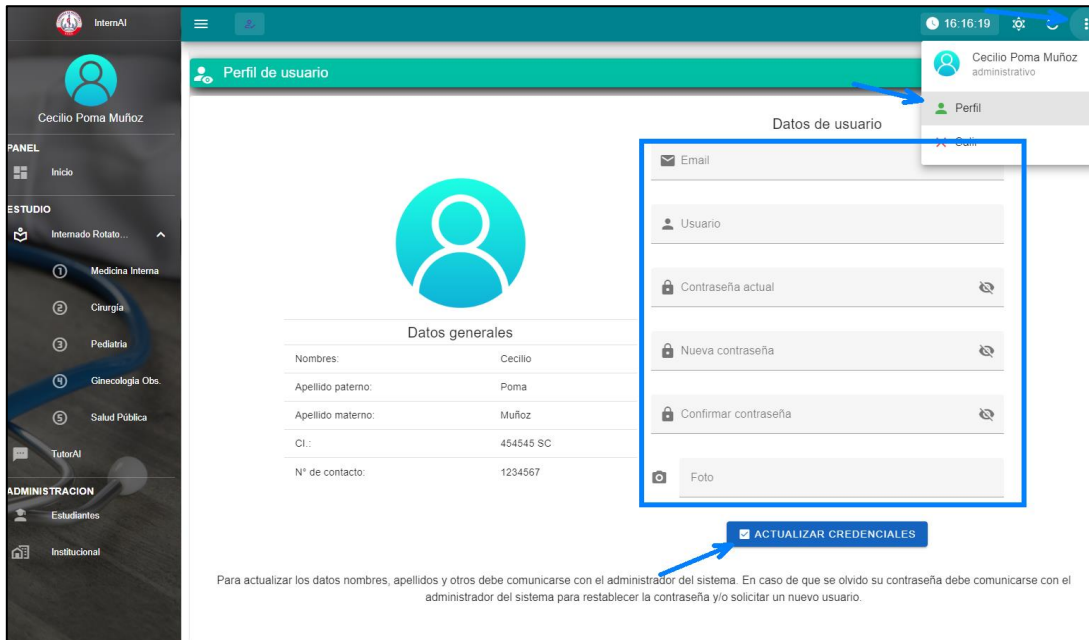
Objetivo: Permitir a los usuarios visualizar y actualizar sus datos personales.

1. Acceso al Perfil:

- Desde la barra superior, seleccione "Perfil".

2. Funciones Disponibles:

- Ver información personal como nombre, correo electrónico y otros datos.
- Puede actualizar solo las credenciales de acceso.
- Complete los campos disponibles.
- Haga clic en "Actualizar credenciales".



4.3. MÓDULO INTERNADO ROTATORIO

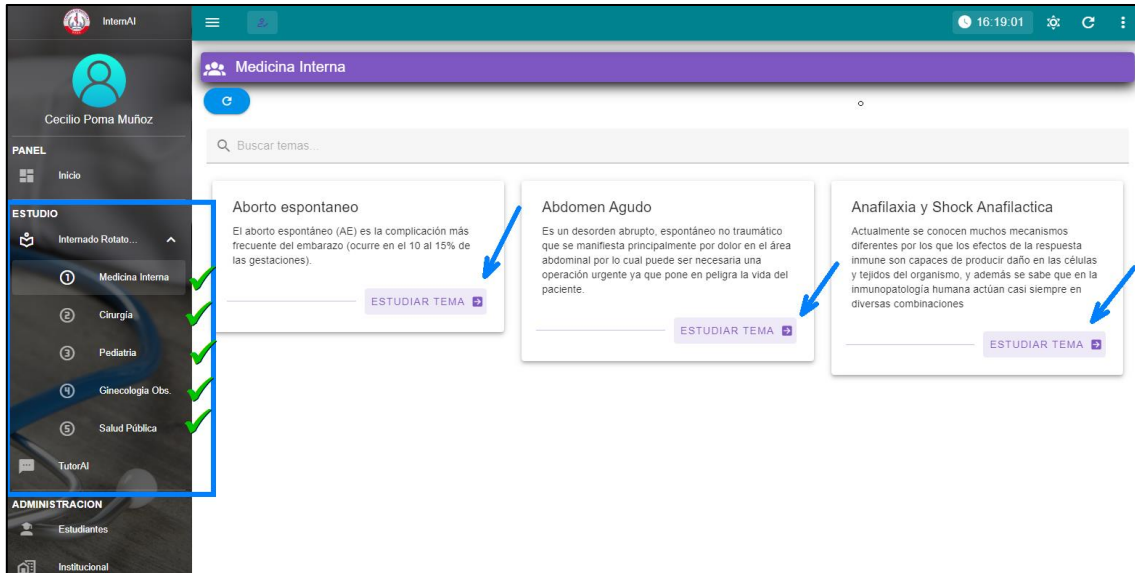
Objetivo: Organizar los materiales de estudio por áreas de especialización.

1. Seleccionar un Área:

- Desde el menú, elija entre Medicina Interna, Cirugía, Ginecología Obstetricia, Pediatría o Salud Pública.

2. Acceso a Temas de Estudio:

- Haga clic en un tema para visualizar su contenido.
- Visualice el contenido de estudio.



4.4. SUBMÓDULO DE ESTUDIO

Objetivo: Proporcionar materiales y cuestionarios dinámicos.

1. Visualización del contenido:
 - Use el visor para navegar por el material.
 - Opciones de búsqueda y zoom están disponibles.
2. Generación y Evaluación de Cuestionarios:
 - Haga clic en "Actividad".
 - Complete las preguntas
 - Clic en "Evaluarme" y reciba retroalimentación inmediata.

4.5. MÓDULO DE PROGRESO DEL ESTUDIANTE

Objetivo: Visualizar y registrar los avances en el estudio.

1. Acceso al Progreso:

- Navegue a "Progreso del Estudiante".
- Consulte áreas completadas, cuestionarios realizados y las evaluaciones obtenidas.

2. Registro de Intentos:

- Visualice el historial de intentos de cada tema y sus resultados.



Progreso de Estudio

TABLERO ACTUALIZAR

Progreso	Tiempo	Tema	Fecha estudio	Acciones
100%	00:03:42	Anafilaxia y Shock Anafilactica	16-11-2024 18:50:45	
5%	00:02:01	Abdomen Agudo	16-11-2024 18:40:27	
5%	00:01:45	Aborto espontaneo	15-11-2024 22:18:35	

Elementos por página: 10 1-3 de 3

Progreso de Estudio

TABLERO ACTUALIZAR

Tema :Anafilaxia y Shock Anafilactica

100% Tiempo de Estudio: 00:03:42 Fecha: 16/11/2024 18:50:45

Intento 1
Cuestionario evaluado
16-11-2024 18:50:45

[VER CUESTIONARIO](#)

Cuestionario evaluado

1 Cuestionario 2 Evaluacion

1. Una persona que ha consumido alimentos a los que es alérgica experimenta una reacción anafiláctica. ¿Cuál es el tipo de reacción inmunológica que se está produciendo en este caso, según la clasificación de Coombs y Gell?
Respuesta: Noste

2. Un paciente que ha sido sometido a un trasplante de órganos desarrolla una reacción inmunológica que causa daño a los tejidos del órgano transplantado. ¿A qué tipo de reacción inmunológica se refiere esta descripción, según la clasificación de Coombs y Gell?
Respuesta: Noste

3. Una persona que ha sido infectada con un parásito experimenta una reacción inmunológica que causa daño a los tejidos del parásito. ¿Cuál es el tipo de reacción inmunológica que se está produciendo en este caso, según la clasificación de Coombs y Gell?
Respuesta: Noste

4. Un paciente que ha sido sometido a un tratamiento con un medicamento alérgico experimenta una reacción anafiláctica. ¿Cuál es el tipo de reacción inmunológica que se está produciendo en este caso, según la clasificación de Coombs y Gell?
Respuesta: Noste

5. Una persona que ha sido infectada con un virus experimenta una reacción inmunológica que causa daño a los tejidos del virus. ¿Cuál es el tipo de reacción inmunológica que se está produciendo en este caso, según la clasificación de Coombs y Gell?
Respuesta: Noste

6. Un paciente que ha sido sometido a un trasplante de médula ósea desarrolla una reacción inmunológica que causa daño a los tejidos del cuerpo. ¿A qué tipo de reacción inmunológica se refiere esta descripción, según la clasificación de Coombs y Gell?
Respuesta: Noste

7. Una persona que ha sido infectada con un hongo experimenta una reacción inmunológica que causa daño a los tejidos del hongo. ¿Cuál es el tipo de reacción inmunológica que se está produciendo en este caso, según la clasificación de Coombs y Gell?
Respuesta: Noste

8. Un paciente que ha sido sometido a un tratamiento con un medicamento que causa una reacción inmunológica experimenta una reacción anafiláctica. ¿Cuál es el tipo de reacción inmunológica que se está produciendo en este caso, según la clasificación de Coombs y Gell?

X CERRAR

4.6. MÓDULO DE TUTORAI

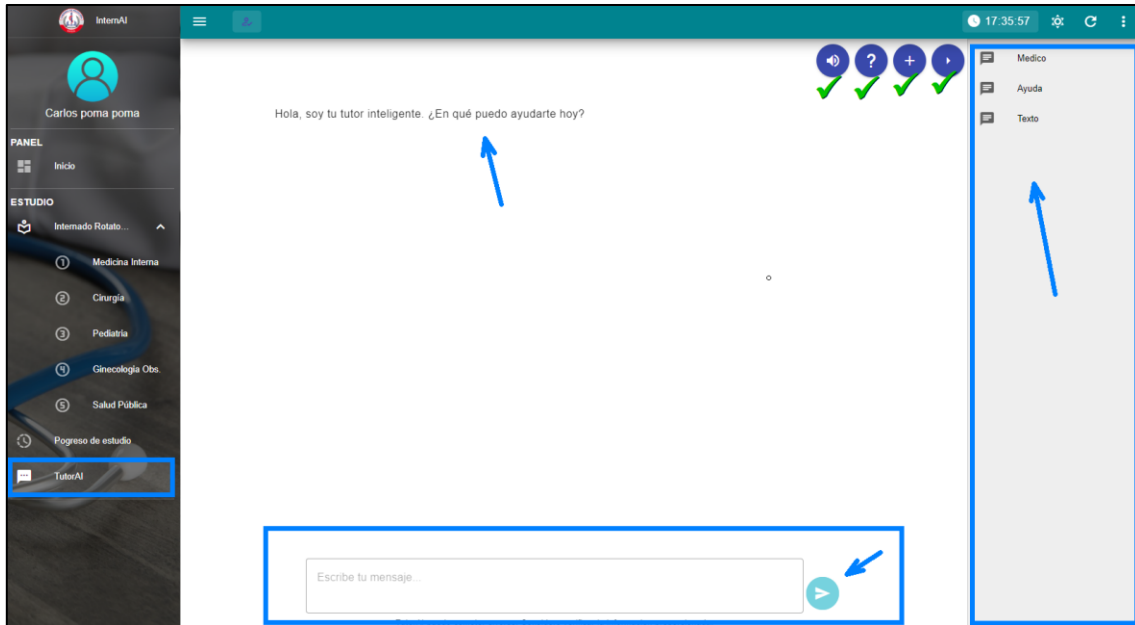
Objetivo: Ofrecer una interacción avanzada con un tutor virtual basado en IA.

1. Realizar Consultas:

- Escriba preguntas sobre los temas estudiados.
- Reciba respuestas detalladas generadas por la red neuronal.

2. Generación de Audio:

- Active esta opción para escuchar las respuestas generadas.



4.7. MÓDULO DE GESTIÓN DE ESTUDIANTES

Objetivo: Permitir al personal institucional gestionar la información de los estudiantes.

1. Registrar Nuevo Estudiantes:

- Navegue a "Estudiantes".
- Haga clic en "Nuevo Estudiante".
- Complete el formulario con los datos básicos (usuario, nombre, apellidos y otros datos).
- Haga clic en "Guardar".

The screenshot shows the 'Estudiantes' management interface. On the left is a sidebar with navigation options: 'Inicio', 'ESTUDIO' (with sub-items: Internado Rotato..., Medicina Interna, Cirugía, Pediatría, Ginecología Obs., Salud Pública, TutorAI), and 'ADMINISTRACION' (with sub-items: Estudiantes, Institucional). The main area has a header 'Estudiantes' and a '+ NUEVO ESTUDIANTE' button highlighted with a blue box and arrow. Below is a search bar 'Buscar Registros...' and a table of students.

Nombres	Apellido paterno	Apellido materno	Numero de contacto	Correo electronico	Activo	Acciones
Carlos	poma	poma	4351213	user@gmail.com	Habilitado	[Edit] [Delete]

The screenshot shows the 'Registrar nuevo estudiante' form. It includes a 'Usuario' section with fields for 'Usuario (*)', 'Contraseña (*)', and 'Correo electronico (*)', each with a green checkmark. A 'Habilitado' toggle is present. The 'Datos del estudiante' section includes fields for 'Nombres (*)', 'Apellido paterno (*)', 'Apellido materno (*)', 'Genero (*)', 'Fecha de nacimiento (*)', 'N° de contacto (*)', 'Matricula universitaria (*)', and 'Internado Rotario (*)', all with green checkmarks.

2. Editar Estudiantes:

- Seleccione un estudiante en el tablero y haga clic en "Editar".
- Realice los cambios necesarios y guarde.



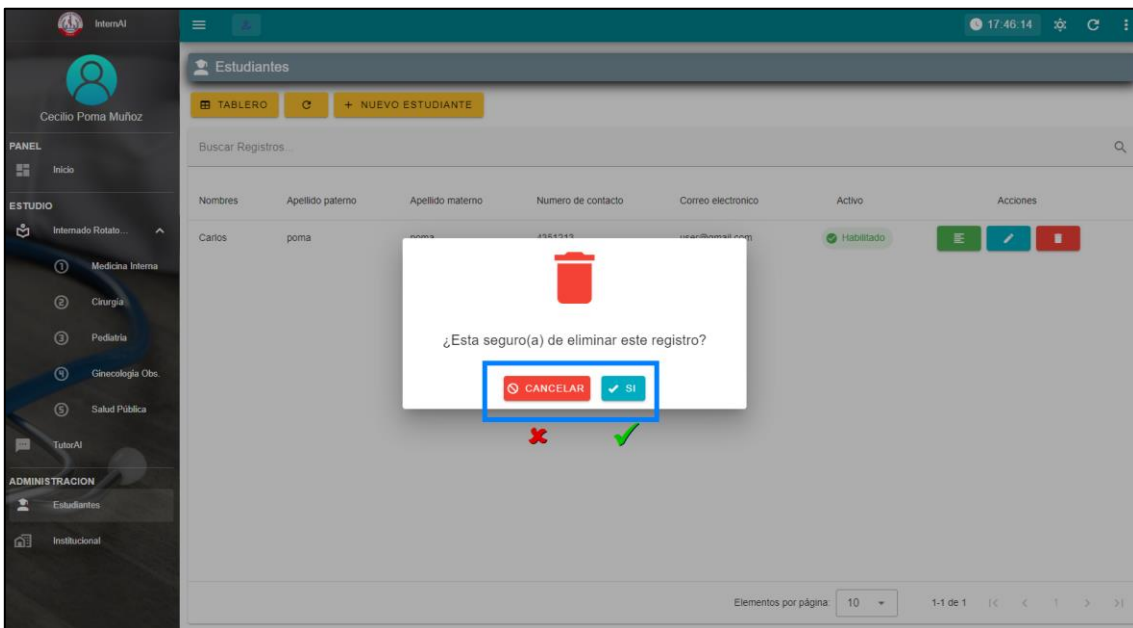
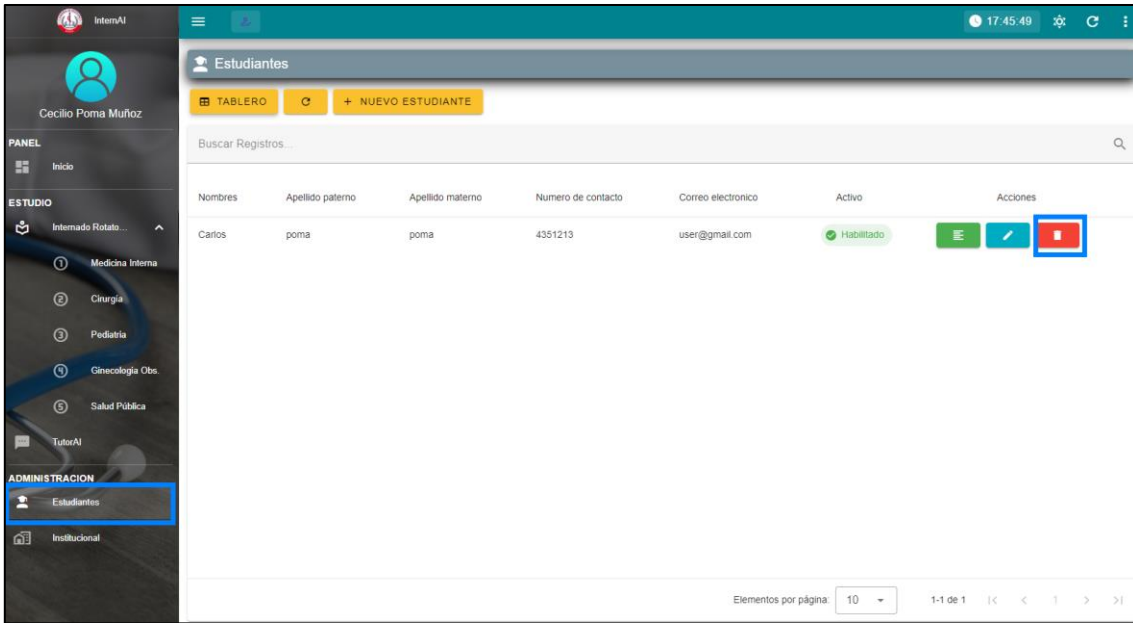
The screenshot shows the 'Estudiantes' management interface. The left sidebar contains the user profile 'Cecilio Poma Muñoz' and navigation menus for 'ESTUDIO' (Medicina Interna, Cirugía, Pediatría, Ginecología Obs., Salud Pública, TutorAI) and 'ADMINISTRACION' (Estudiantes, Institucional). The main area displays a table with the following data:

Nombres	Apellido paterno	Apellido materno	Numero de contacto	Correo electronico	Activo	Acciones
Carlos	poma	poma	4351213	user@gmail.com	Habilitado	[Edit] [Delete]

The screenshot shows the 'Actualizar datos del estudiante' form. It includes a 'Usuario' section with fields for 'Usuario (*)' (value: user), 'Contraseña', and 'Correo electronico (*)' (value: user@gmail.com). A 'Habilitado' toggle is checked. The 'Datos del estudiante' section includes fields for 'Nombres (*)' (Carlos), 'Apellido paterno (*)' (poma), 'Apellido materno (*)' (poma), 'Genero (*)' (masculino), 'Fecha de nacimiento (*)' (15/11/2024), 'N° de contacto (*)' (4351213), 'Matrícula universitaria (*)' (1535121), and 'Internado Rotario (*)' (hospital).

3. Eliminar Estudiantes:

- Haga clic en el botón "Eliminar" junto al estudiante correspondiente.
- Confirme la acción.

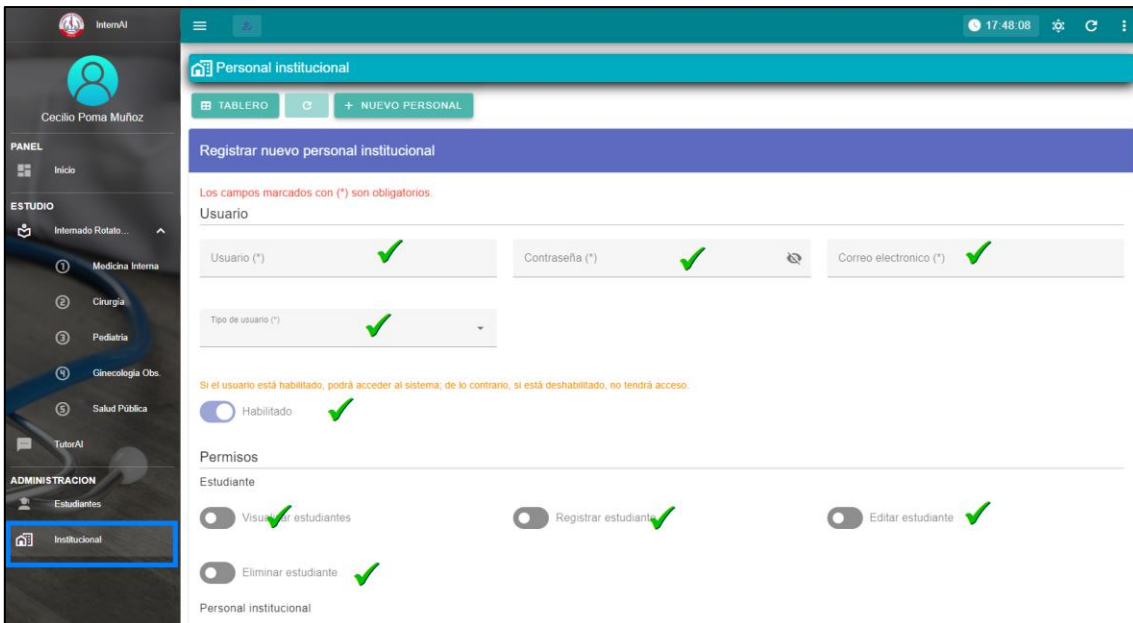
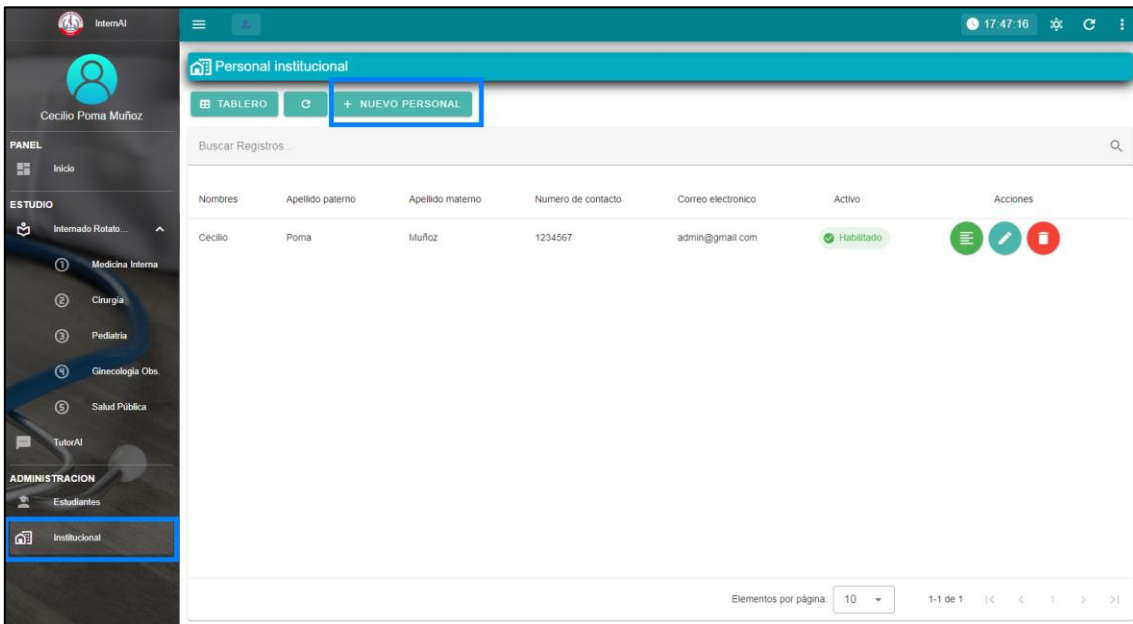


4.8. MÓDULO DE GESTIÓN DE PERSONAL INSTITUCIONAL

Objetivo: Administrar cuentas del personal institucional.

1. Registrar Nuevo Personal:
 - Navegue a "Institucional".
 - Haga clic en "Nuevo Personal".

- Complete el formulario con los datos básicos (usuario, nombre, apellidos, asigne permisos necesarios y otros datos).
- Haga clic en "Guardar".



2. Editar Personal:

- Seleccione un personal registrado en el tablero y haga clic en "Editar".
- Realice los cambios necesarios y guarde.

Nombres	Apellido paterno	Apellido materno	Numero de contacto	Correo electronico	Activo	Acciones
Cecilio	Poma	Muñoz	1234567	admin@gmail.com	Habilitado	[Eliminar]

Actualizar datos del personal institucional

Los campos marcados con (*) son obligatorios.

Usuario

Usuario (*) admin Contraseña Correo electronico (*) admin@gmail.com

Tipo de usuario (*) administrativo

Si el usuario está habilitado, podrá acceder al sistema, de lo contrario, si está deshabilitado, no tendrá acceso.

Habilitado

Permisos

Estudiante

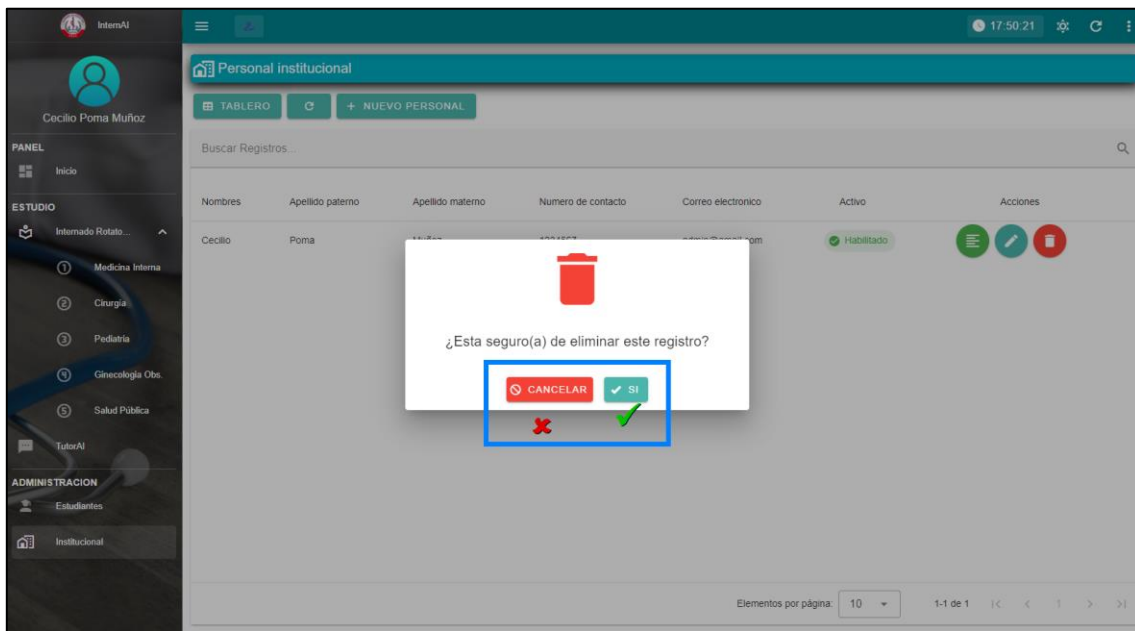
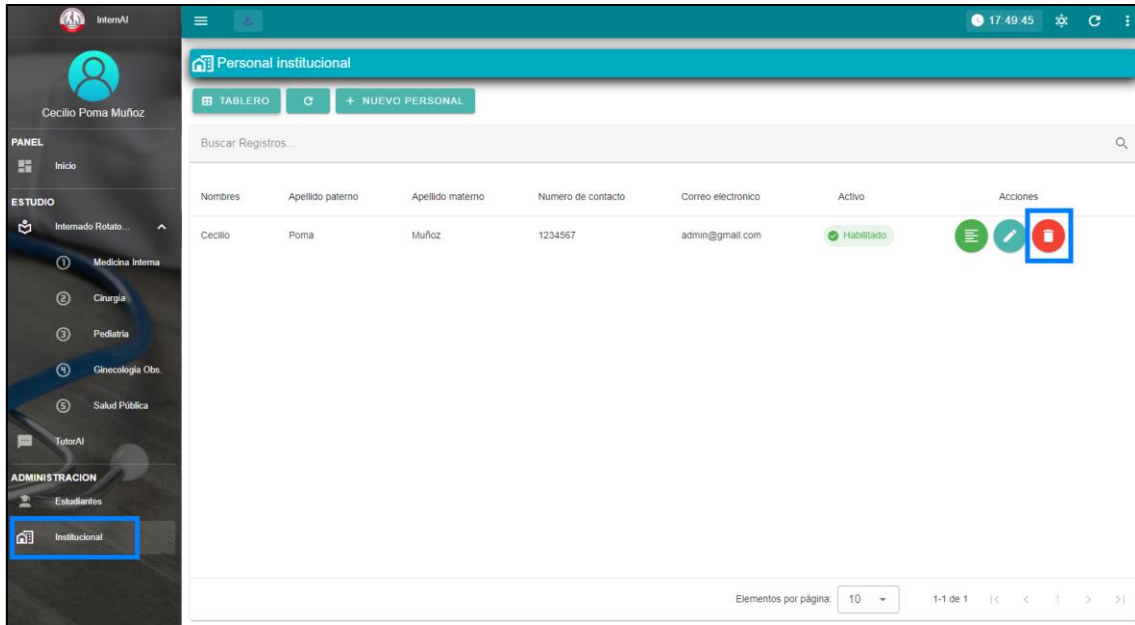
Visualizar estudiantes Registrar estudiante Editar estudiante

Eliminar estudiante

Personal institucional

3. Eliminar Personal:

- Haga clic en el botón "Eliminar" junto al personal registrado correspondiente.
- Confirme la acción.



5. PREGUNTAS FRECUENTES

1. ¿Cómo puedo cambiar mi contraseña?
 - Desde el perfil de usuario, seleccione "Editar" y siga las instrucciones.
2. ¿Qué hago si no puedo acceder al sistema?
 - Verifique su conexión a internet y sus credenciales.

MANUAL TÉCNICO

**INGENIERÍA
DE SISTEMAS**
UNIVERSIDAD PÚBLICA DE EL ALTO





MANUAL TÉCNICO

SISTEMA TUTOR INTELIGENTE BASADO EN REDES NEURONALES ARTIFICIALES “INTERNAI”

Versión 1.0

2024



INDICE

1. INTRODUCCIÓN.....	3
2. REQUISITOS DEL SISTEMA	3
2.1. REQUISITOS DE HARDWARE.....	3
2.2. REQUISITOS DE SOFTWARE	3
3. ARQUITECTURA DEL SISTEMA	4
3.1. DESCRIPCIÓN GENERAL	4
4. INSTALACIÓN Y CONFIGURACIÓN	4
4.1. BACKEND.....	4
4.2. FRONTEND	5
5. MÓDULOS DEL SISTEMA.....	5
5.1. MÓDULO DE INICIO DE SESIÓN	5
5.2. MÓDULO DE PERFIL DE USUARIO	6
5.3. MÓDULO INTERNADO ROTATORIO.....	6
5.4. SUBMÓDULO DE ESTUDIO	7
5.5. MÓDULO DE PROGRESO DEL ESTUDIANTE.....	7
5.6. MÓDULO DE TUTORIA.....	7
5.7. MÓDULO ESTUDIANTES.....	8
5.8. MÓDULO DE GESTIÓN DE PERSONAL INSTITUCIONAL.....	8



1. INTRODUCCIÓN

El Manual Técnico de InternAI describe los aspectos tecnológicos del sistema para garantizar una correcta instalación, mantenimiento y actualización. Este documento está dirigido a desarrolladores, administradores de sistemas y técnicos responsables del soporte.

2. REQUISITOS DEL SISTEMA

2.1. REQUISITOS DE HARDWARE

Backend y frontend:

- Hosting: 100 GB SSD, python app.

Red neuronal:

- Inference Endpoints: Nvidia L4, 1 GPU (RAM 24GB)

Ciente:

- Dispositivos compatibles: Computadora, tableta o teléfono inteligente.
- Resolución mínima: 1280 x 720 píxeles.

2.2. REQUISITOS DE SOFTWARE

Backend:

- django 5.1.2
- django-cors-headers 4.4.0
- django-enviro 0.11.2
- django-seeding 1.3.5
- djangorestframework 3.15.2
- djangorestframework-simplejwt 5.3.1
- numpy 2.1.2
- pandas 2.2.3
- pillow 10.4.0
- psycopg2-binary 2.9.9



- django_csp 4.0b1
- gttts 2.5.3
- dj-database-url 2.3.0
- Python 3.12.x

Frontend:

- Node.js 20.11.1
- Vue.js 3.4.29
- Vuetify 3.7.1.
- vue-i18n 10.0.4

Otros:

- Docker (opcional para despliegue).
- Navegadores compatibles: Google Chrome, Firefox, Edge.

3. ARQUITECTURA DEL SISTEMA

3.1. DESCRIPCIÓN GENERAL

El sistema InternAI sigue una arquitectura cliente-servidor con separación de responsabilidades:

- Frontend: Vue.js y Vuetify para interfaces de usuario.
- Backend: Django REST Framework para lógica del negocio y gestión de APIs.
- Base de Datos: PostgreSQL para almacenamiento relacional.
- Modelo IA: Modelo LLM implementado con APIs REST.

4. INSTALACIÓN Y CONFIGURACIÓN

4.1. BACKEND

- Clonar el Repositorio:

`git clone https://github.com/poma61/backend-sti-medicina.git`

- Instalar Dependencias:



```
pip install -r requirements.txt
```

- Configurar Base de Datos

Crear una base de datos en PostgreSQL:

- *CREATE DATABASE db_intern_ai;*

- Aplicar Migraciones

```
python manage.py migrate
```

- Aplicar Seeders:

```
python manage.py seed
```

4.2. FRONTEND

- Clonar el Repositorio:

```
git clone https://github.com/poma61/frontend-sti-medicina.git
```

- Instalar Dependencias:

```
npm install
```

5. MÓDULOS DEL SISTEMA

5.1. MÓDULO DE INICIO DE SESIÓN

Permite a los usuarios acceder al sistema de manera segura mediante credenciales de usuario y contraseña.

- El usuario debe ingresar un nombre de usuario y contraseña válidos en la pantalla principal.
- El sistema valida las credenciales contra la base de datos.
- Si las credenciales son correctas, el usuario será redirigido al panel principal según su permiso (estudiante o personal institucional).
- En caso de olvido de contraseña, el usuario deberá contactar al administrador para restablecerla.



5.2. MÓDULO DE PERFIL DE USUARIO

Permite a los usuarios visualizar y actualizar sus datos personales.

- Los usuarios pueden acceder a esta sección desde la barra superior del sistema.
- Visualizan información básica como su nombre, correo electrónico y rol dentro del sistema.
- Solo pueden modificar credenciales de acceso (nombre de usuario y contraseña) mediante un formulario protegido.
- La actualización de credenciales requiere que el usuario proporcione su contraseña actual como medida de seguridad.

5.3. MÓDULO INTERNADO ROTATORIO

Organiza y presenta los materiales de estudio divididos por áreas de especialización médica.

1. Las áreas disponibles incluyen:
 - Medicina Interna
 - Cirugía
 - Ginecología y Obstetricia
 - Pediatría
 - Salud Pública
2. Los estudiantes pueden seleccionar un área para explorar temas específicos.
3. Al hacer clic en un tema, acceden al contenido de estudio en formato PDF u otros recursos relacionados.
4. Este módulo sirve como base para las actividades de estudio, como generar cuestionarios o consultar al TutorAI.



5.4. SUBMÓDULO DE ESTUDIO

Facilita el acceso al material de estudio y la realización de cuestionarios dinámicos basados en los temas seleccionados.

- Los estudiantes pueden navegar por el material proporcionado utilizando un visor integrado con opciones de zoom, búsqueda y desplazamiento.
- Este submódulo incluye una funcionalidad para generar cuestionarios dinámicos relacionados con el tema actual.
- Los cuestionarios son evaluados automáticamente por el modelo de red neuronal, y los estudiantes reciben retroalimentación inmediata sobre sus respuestas.
- Registra cada intento de cuestionario y almacena los resultados en el módulo de progreso del estudiante.

5.5. MÓDULO DE PROGRESO DEL ESTUDIANTE

Registra y muestra el avance del estudiante en cada área y tema de estudio.

1. Los estudiantes pueden visualizar estadísticas detalladas sobre:
 - Áreas completadas.
 - Número de cuestionarios realizados.
 - Resultados obtenidos en cada actividad.
2. El historial de intentos muestra el rendimiento del estudiante por tema, lo que permite evaluar su evolución a lo largo del tiempo.
3. La interfaz presenta gráficos o tablas para facilitar la interpretación del progreso.

5.6. MÓDULO DE TUTORAI

Ofrece una interacción avanzada con el modelo de inteligencia artificial para resolver dudas y generar respuestas relacionadas con los temas de estudio.

- Los estudiantes pueden realizar preguntas abiertas al modelo escribiéndolas en un campo de texto.



- El modelo de red neuronal genera una respuesta detallada basada en el contexto del tema seleccionado.
- Este módulo incluye una funcionalidad adicional para convertir las respuestas generadas en audio, facilitando el aprendizaje para estudiantes con necesidades específicas.

5.7. MÓDULO ESTUDIANTES

Permite al personal institucional administrar la información de los estudiantes registrados en el sistema.

1. Registrar estudiantes:
 - Completar un formulario con datos básicos como nombre, correo electrónico y otros datos.
 - Crear credenciales de acceso únicas para cada estudiante.
2. Editar estudiantes:
 - Actualizar información existente en caso de errores o cambios necesarios.
3. Eliminar estudiantes:
 - Eliminar cuentas obsoletas o duplicadas del sistema.

5.8. MÓDULO DE GESTIÓN DE PERSONAL INSTITUCIONAL

Gestiona las cuentas del personal administrativo encargado del sistema.

1. Registrar personal institucional:
 - Crear nuevas cuentas para administradores o personal docente.
 - Asignar permisos específicos según las responsabilidades de cada usuario.
2. Editar personal:
 - Modificar datos existentes, como nombre, correo electrónico o permisos asignados.
3. Eliminar personal:
 - Retirar cuentas de personal institucional.