

**Desarrollo de un Prototipo de chat inteligente para la Optimización y Mejora de Procesos  
Administrativos y Operativos en una Institución Educativa**

**AUTORES:**

**Luis Alejandro González González**

**Brian Alexander Patiño Santander**

**UNIVERSIDAD DE CUNDINAMARCA**

**Facultad de ingeniería**

**Programa de ingeniería de sistemas**

**Facatativá, abril 2025**

**Desarrollo de un Prototipo de chat inteligente para la Optimización y Mejora de Procesos  
Administrativos y Operativos en una Institución Educativa**

**AUTORES:**

**Luis Alejandro González González**

**Brian Alexander Patiño Santander**

**Director: Carlos Alfonso Vargas Morales**

**UNIVERSIDAD DE CUNDINAMARCA**

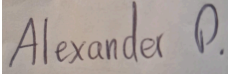
**Facultad de ingeniería**

**Programa de ingeniería de sistemas**

**Facatativá, abril 2025**

### COMPROMISO DE AUTOR

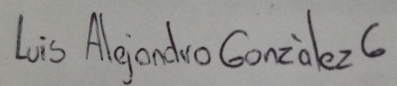
Yo, Brian Alexander Patiño Santander con cédula de identidad No. 1007509943 y con cód. 461219156 estudiante del programa de Ingeniería de Sistemas de la Universidad de Cundinamarca, declaró que: el contenido del presente documento es un reflejo de mi trabajo personal y manifiesto que, ante cualquier notificación de plagio, copia o falta a la fuente original, soy responsable directo legal, económico y administrativo sin afectar al director del trabajo, a la Universidad y a cuantas instituciones hayan colaborado en dicho trabajo, asumiendo las consecuencias derivadas de tales prácticas.

Firma: 

### COMPROMISO DE AUTOR

Yo, Luis Alejandro Gonzalez Gonzalez con cédula de identidad No. 1003777589 y con cód. 461219132 estudiante del programa de Ingeniería de Sistemas de la Universidad de Cundinamarca, declaró que: el contenido del presente documento es un reflejo de mi trabajo personal y manifiesto que, ante cualquier notificación de plagio, copia o falta a la fuente original, soy responsable directo legal, económico y administrativo sin afectar al director del trabajo, a la Universidad y a cuantas instituciones hayan colaborado en dicho trabajo, asumiendo las consecuencias derivadas de tales prácticas.

Firma:

Handwritten signature of Luis Alejandro Gonzalez G in black ink on a light gray background.

## RESUMEN

Este proyecto se presenta como prototipo de Chat Inteligente con el objetivo de optimizar la eficacia de la administración institucional en ambientes educativos. El sistema, desarrollado a través de FastAPI para el backend y Flutter para el frontend, está diseñado para distinguir funciones dependiendo del papel del usuario (estudiante, profesor, administrativo y administrador), garantizando respuestas automatizadas y personalizadas.

Mediante métodos de Procesamiento de Lenguaje Natural (PLN), inteligencia artificial y modelos de intentos personalizados, el chatbot procesa peticiones vinculadas a procesos internos, tales como peticiones de horarios, anotaciones, registros o autorizaciones. Además, se incorporaron sistemas sólidos de protección de datos, que incluyen encriptación de contraseñas, confirmación de identidad a través del correo institucional y cuestiones de seguridad.

En el proceso de desarrollo, se llevaron a cabo evaluaciones internas utilizando datos simulados, evaluando la eficacia operacional, los tiempos de respuesta y la solución de consultas. Esta valoración facilitó la mejora del rendimiento del sistema antes de su implementación en un contexto real.

La solución sugerida tiene como objetivo automatizar los procesos de interacción en la institución, disminuyendo la carga administrativa, potenciando la administración de información en línea acorde a las demandas actuales de digitalización educativa.

## ABSTRACT

This project implements an Intelligent Chat system with the aim of optimizing the efficiency of institutional administration in educational environments. The system, developed using FastAPI for the backend and Flutter for the frontend, is designed to distinguish functions depending on the user's role (student, teacher, administrative staff, and administrator), ensuring automated and personalized responses.

Through Natural Language Processing (NLP) methods and customized intent models, the chatbot processes requests related to internal processes, such as requests for schedules, grades, registrations, or authorizations. In addition, robust data protection systems were incorporated, including password encryption, identity confirmation through institutional email, and security questions.

During the development process, internal evaluations were carried out using simulated data, assessing operational efficiency, response times, and query resolution. This evaluation facilitated performance improvement of the system before its implementation in a real-world context.

The proposed solution aims to automate and enhance interaction processes within the institution, reducing administrative workload and strengthening information management online in line with current demands for educational digitalization.

**Palabras clave:** Optimización de procesos, Gestión de entidades, Procesamiento de lenguaje natural, Identificación de roles, Inteligencia artificial aplicada, Seguridad de datos.

## TABLA DE CONTENIDO

I. INFORME DE INVESTIGACIÓN.....	9
1.1. ESTADO DEL ARTE.....	9
1.2. LÍNEA DE INVESTIGACIÓN.....	27
1.3. PLANTEAMIENTO DEL PROBLEMA Y PREGUNTA DE INVESTIGACIÓN.....	28
1.4. OBJETIVO GENERAL Y OBJETIVOS ESPECÍFICOS.....	29
1.5. ALCANCE E IMPACTO DEL PROYECTO.....	30
1.6. METODOLOGÍA.....	33
1.6.1 METODOLOGÍA DE INVESTIGACIÓN MIXTA.....	33
1.6.2. Design Thinking.....	34
1.7. MARCOS DE REFERENCIA.....	37
1.7.1. MARCO TEÓRICO.....	37
1.7.1.1. Inteligencia Artificial (IA):.....	38
1.7.1.2. Procesamiento del Lenguaje y Representación del Conocimiento (incluyendo PLN):	38
1.7.1.3. Modelo de Investigación Mixto:.....	38
1.7.1.4. Marcos a Nivel de Aplicación (FastAPI, Flutter, TensorFlow):.....	38
1.7.2. MARCO LEGAL.....	39
1.7.2.1. Ley de Protección de Información Personal en Colombia (Ley 1581 de 2012).....	39
1.7.2.2. Decreto 1377 de 2013.....	39
1.7.2.3. Licencia de Uso de FastAPI.....	39
1.7.2.4. Licencia de Uso de Flutter.....	39
1.7.2.5. Licencia de Uso de TensorFlow.....	39
1.7.2.6. LEY N.º 23 DE 1982 DERECHOS DE AUTOR EN COLOMBIA.....	39
II. DOCUMENTACIÓN DEL SOFTWARE.....	40
2.1. PLAN DE PROYECTO.....	40
2.2. DETERMINACIÓN DE REQUERIMIENTOS.....	40
2.2.1.1. Propósito.....	40
2.2.1.2. Ámbito del sistema.....	41
2.2.1.3. Definiciones, Acrónimos y Abreviaturas.....	42

2.2.1.4. Visión General del Documento.....	43
2.2.2 DESCRIPCIÓN GENERAL.....	44
2.2.2.1. Perspectiva del Producto.....	45
2.2.2.2. Funciones del Producto.....	46
2.2.2.3. Características de los usuarios.....	47
2.2.2.3.1. Funciones Específicas y Validaciones por Rol.....	49
2.2.2.4. Restricciones.....	59
2.2.2.5. Suposiciones y Dependencias.....	60
2.2.2.6. Requisitos Futuros.....	62
2.2.3 REQUISITOS ESPECÍFICOS.....	63
2.3. ESPECIFICACIÓN DEL DISEÑO.....	67
2.3.1. MODELO DE ENTIDAD RELACIÓN (MER) -TABLAS PARA LA LÓGICA DE NEGOCIO DEL PROYECTO.....	67
2.3.1.2. Descripción de las tablas de la base de datos del proyecto.....	67
2.3.2. DIAGRAMAS DE CASOS DE USO.....	69
2.3.3. DIAGRAMAS DE SECUENCIAS.....	71
2.3.4. DIAGRAMAS DE ACTIVIDADES.....	71
2.3.5. DIAGRAMA DE CLASES.....	72
2.3.6. DESCRIPCIÓN DE LAS CLASES DEL PROYECTO.....	73
2.4. DISEÑO DE LOS CASOS DE PRUEBA.....	74
2.4.1 Pruebas de Calidad del Código.....	74
2.4.2. DEFINICIÓN DE LOS CASOS DE PRUEBA FUNCIONALES DE LA APLICACIÓN.....	75
2.4.3. EVALUACIÓN TÉCNICA DEL CÓDIGO.....	76
2.5. ESTIMACIÓN DE RECURSOS.....	77
2.6. VALIDACIÓN INTERNA DEL PROYECTO.....	78
2.7. CONCLUSIONES Y RECOMENDACIONES.....	78
2.8. LIMITACIONES DEL PROTOTIPO.....	79
2.9. PROYECCIONES FUTURAS.....	79
3.1. BIBLIOGRAFÍA.....	80
4. ANEXOS.....	83
4.1. MANUALES.....	83
4.1.1. Manual técnico.....	83
4.1.2. Manual de usuario.....	88
4.2 SOLICITUD DE DERECHOS DE AUTOR.....	104
4.2 PRUEBAS TÉCNICAS.....	104

## LISTA DE TABLAS

Tabla 1. Alcance del Proyecto.....	34
Tabla 2 Funcionalidades fuera del alcance .....	34
Tabla 3. Chat inteligente institucional.....	38
Tabla 4. Usuario.....	39
Tabla 5. Casos de prueba.....	77

Tabla 6 estimación de recursos.....	79
-------------------------------------	----

## LISTA DE FIGURAS

Diagrama 1 MER. Fuente propia.....	67
Diagrama 2.Casos de uso participantes. fuente propia.....	70
Diagrama 3. Casos de uso administrador. fuente propia.....	70
Diagrama 4 secuencia 1. fuente propia.....	71
Diagrama 5 secuencia 2. fuente propia.....	71
Diagrama 6 actividades. fuente propia.....	72
Diagrama 7 Clases. fuente propia.....	7
Figura 1. inicio de sesión. fuente propia.....	91
Figura 2. registro. fuente propia.....	92
Figura 3 Verificación. fuente propia.....	93
Figura 4 en espera. fuente propia.....	93
Figura 5 términos y condiciones. fuente propia.....	93
Figura 6 Asignación de rol. fuente propia.....	94
Figura 7 inicio de sesión. fuente propia.....	95
Figura 8 pantalla de inicio.fuente propia.....	96
Figura 9. recuperar. fuente propia.....	97
Figura 10. recuperar contraseña. fuente propia.....	97
Figura 11. pregunta. fuente propia.....	98
Figura 12.Código correo.fuente propia.....	99
Figura 13. código de recuperación. fuente propia.....	99
Figura 14 nueva contraseña. fuente propia.....	99
Figura 15 interfaz. fuente propia.....	100
Figura 16 fin de sesión. fuente propia.....	101
Figura 17 eliminar. fuente propia.....	101
Figura 18. confirmación. fuente propia.....	102
Figura 19 asignación. fuente propia.....	103
Figura 20. Prueba técnica #1.....	105
Figura 21. Prueba técnica #2.....	106
Figura 22. Prueba técnica #3.....	106
Figura 23. Prueba técnica #4.....	107
Figura 24. Prueba técnica #5.....	107
Figura 25.Prueba técnica #6.....	107
Figura 26. Prueba técnica #7.....	108
Figura 27.Prueba técnica #8.....	109
Figura 28.Prueba técnica #9.....	109
Figura 29.Prueba técnica #10.....	110
Figura 30. Prueba técnica #11.....	110
Figura 31. Prueba técnica #12.....	111
Figura 32.Prueba técnica #13.....	111
Figura 33. Prueba técnica #14.....	112
Figura 34.Prueba técnica #15.....	112
Figura 35.Prueba técnica #16.....	113

Figura 36.Prueba técnica #17..... 113

## I. INFORME DE INVESTIGACIÓN

### 1.1. ESTADO DEL ARTE

<b>TÍTULO</b>
Implementación un chat bot basado en analítica de datos para la disminución de tiempos de respuesta y mejorar la calidad del servicio de la Corporación Unificada Nacional De Educación Superior
<b>AUTOR(ES)</b>
Espinosa Peña, Ricardo Andres, Castillo Díaz, Michael Andres
<b>PUBLICACIÓN</b>
2024
<b>DESCRIPCIÓN</b>
<p>Según Espinosa Peña y Castillo Díaz (2024), el proyecto "CUNDIGITAL" busca mejorar la calidad del servicio en la Universidad CUN mediante un chatbot que responde consultas personalizadas y organiza trámites académicos. Incluye analítica de datos para evaluar la interacción y satisfacción de los estudiantes.</p> <p>Nuestro proyecto comparte el objetivo de optimizar la experiencia del usuario mediante la automatización, pero se enfoca en resolver problemas administrativos y de permisos, con un alcance más amplio que solo la interacción estudiantil. Además, buscamos integrar roles diferenciados (administradores, docentes, estudiantes) y personalizar las respuestas según las necesidades de cada usuario.</p>

<b>TÍTULO</b>
chatbot educativo para un curso de ingeniería de software
<b>AUTOR(ES)</b>
RAFAEL CAMILO ARCHILA CUY
<b>PUBLICACIÓN</b>
2022

<b>DESCRIPCIÓN</b>
<p>Según Archila Cuy (2022), este proyecto aborda los desafíos de la enseñanza de ingeniería de software mediante un asistente educativo conversacional que brinda apoyo personalizado las 24 horas del día. Su enfoque está en dirigir a los estudiantes hacia fuentes confiables y adaptarse a diferentes estilos de aprendizaje, mejorando la experiencia educativa y ampliando las capacidades de asesoramiento de los profesores.</p> <p>Comparado con nuestro proyecto, compartimos el objetivo de proporcionar una solución accesible y eficiente, pero nuestro chatbot se enfoca en la gestión de procesos administrativos, como permisos y consultas institucionales, además de integrar roles diferenciados. Aunque ambos proyectos buscan mejorar la interacción educativa, nuestro enfoque amplía el alcance más allá de la orientación académica directa hacia la optimización de operaciones dentro de las instituciones educativas.</p>

<b>TÍTULO</b>
Aplicación móvil para interconexión con chatbot en el proceso de enseñanza – aprendizaje de competencias ciudadanas.
<b>AUTOR(ES)</b>
Rincón Rico, Erika Marcela Molina González, Maria Fernanda
<b>PUBLICACIÓN</b>
2022
<b>DESCRIPCIÓN</b>
<p>Según Rincón Rico y Molina González (2022), este proyecto propone una aplicación móvil con un chatbot de fácil manejo para mitigar la brecha académica en la educación pública de Tunja. Su enfoque está en la enseñanza de competencias cívicas, ofreciendo a los estudiantes la posibilidad de registrarse, iniciar sesión, dialogar con el asistente virtual, revisar su perfil y realizar pruebas sobre habilidades cívicas desde sus teléfonos móviles. La iniciativa busca igualar las oportunidades entre instituciones públicas y privadas, promoviendo el acceso a herramientas tecnológicas educativas.</p>

<b>TÍTULO</b>
Asistente educativo inteligente para la utilización de tecnologías IOT aplicadas en las aulas de clase de las instituciones educativas de Colombia
<b>AUTOR(ES)</b>
Vergara Herrera, Rafael David
<b>PUBLICACIÓN</b>
2021
<b>DESCRIPCIÓN</b>
<p>Según Vergara Herrera (2021), este proyecto explora el impacto de un asistente educativo virtual en el desarrollo de capacidades de razonamiento lógico y resolución de problemas en estudiantes secundarios. Mediante la interacción con el asistente, se observaron mejoras significativas en habilidades cognitivas, lo que respalda el uso de tecnologías digitales en el ámbito educativo para enriquecer el aprendizaje y promover el razonamiento lógico.</p> <p>Comparado con nuestro proyecto, aunque ambos utilizan tecnologías avanzadas para mejorar el sistema educativo, nuestro enfoque se centra en la automatización de procesos administrativos y operativos dentro de instituciones educativas. Mientras el proyecto de</p>

Vergara Herrera busca directamente desarrollar habilidades cognitivas en estudiantes, nuestro chatbot se orienta a optimizar la gestión interna, asegurando que los recursos administrativos sean más accesibles y eficaces.

<b>TÍTULO</b>
Asistente virtual de tipo ChatBot
<b>AUTOR(ES)</b>
García-Reina, Luis Felipe
<b>PUBLICACIÓN</b>
2018
<b>DESCRIPCIÓN</b>
<p>Según García-Reina (2018), este trabajo de grado se enfocó en diseñar un asistente virtual para gestionar peticiones, quejas y reclamos en la Corporación Educativa San Isidro. El sistema utiliza técnicas de inteligencia artificial para clasificar mensajes y semi-automatizar procesos administrativos, integrando tecnologías como web services, aprendizaje profundo, bases de datos relacionales (MySQL y RedisDB), y procesamiento de lenguaje natural, implementado en Python con una arquitectura orientada a modelos.</p> <p>Comparado con nuestro proyecto, compartimos el objetivo de optimizar procesos administrativos mediante la automatización y el uso de inteligencia artificial. Sin embargo, nuestro chatbot amplía su enfoque al incluir la gestión de permisos y consultas institucionales específicas dentro de un contexto más generalizado de operaciones educativas. Además, buscamos personalizar las respuestas según roles diferenciados, adaptando la funcionalidad a las necesidades específicas de estudiantes, docentes y administrativos.</p>

<b>TÍTULO</b>
Chatbot conversacional de apoyo al uso del aula extendida de la Universidad Simón Bolívar
<b>AUTOR(ES)</b>
Castro Quijano, Lucy Andrea Henríquez Herazo, Pedro Manuel Ortiz Torres, Luis Carlos Pacheco Noguera, Milton José
<b>PUBLICACIÓN</b>
2021
<b>DESCRIPCIÓN</b>
<p>Según Castro Quijano et al. (2021), este proyecto desarrolló un chatbot conversacional llamado TELMA, diseñado para resolver preguntas de los estudiantes sobre el uso de la plataforma de gestión de aprendizaje Moodle en la Universidad Simón Bolívar. Su propósito principal fue mejorar la comprensión y uso de esta plataforma, apoyando a los estudiantes en actividades académicas diarias y mitigando problemas de rendimiento relacionados con el manejo incorrecto de la herramienta.</p> <p>Comparado con nuestro proyecto, compartimos el enfoque en la mejora de la experiencia del usuario mediante un chatbot, pero nuestro desarrollo se centra en la gestión de procesos administrativos, como permisos y consultas operativas, dentro de instituciones educativas.</p> <p>Además, nuestro chatbot busca integrar roles diferenciados (estudiantes, docentes, administrativos) para personalizar la interacción, mientras que TELMA está orientado exclusivamente al soporte académico sobre el uso de plataformas tecnológicas.</p>

--

<b>TÍTULO</b>
Gestión de información académica de solicitudes estudiantiles y docentes en la Jefatura de Software de la UNICESMAG mediante Chatbot aplicando PLN
<b>AUTOR(ES)</b>
Jamanoy Bacca, Brayan Camilo Montenegro Rosero, Jeisson Fernando
<b>PUBLICACIÓN</b>
2023
<b>DESCRIPCIÓN</b>
Según Jamanoy Bacca y Montenegro Rosero (2023), este proyecto utiliza un chatbot con procesamiento de lenguaje natural (PLN) para gestionar solicitudes académicas en la Jefatura de Software de la UNICESMAG. Su objetivo es automatizar procesos manuales susceptibles a errores, asegurando una resolución oportuna de solicitudes mediante herramientas tecnológicas, como plataformas de mensajería. El chatbot contribuye al procesamiento eficiente de información y mejora la reputación institucional al optimizar el manejo de solicitudes.

Comparado con nuestro proyecto, ambos comparten la intención de automatizar procesos administrativos para mejorar la eficiencia en las instituciones educativas. Sin embargo, nuestro proyecto amplía el alcance al integrar funciones específicas, como la gestión de permisos y consultas institucionales, con personalización según los roles diferenciados de usuarios. Además, nuestro chatbot busca ser escalable y replicable en distintas instituciones, lo que lo diferencia del enfoque puntual aplicado en la Jefatura de Software de la UNICESMAG.

<b>TÍTULO</b>
Chatbot que facilita la información en la Facultad de Ingenierías de la Universidad Simón Bolívar
<b>AUTOR(ES)</b>
G. Casseres, J. Cuao, M. Londoño, L. Obredor, S. Orozco & P. Sánchez
<b>PUBLICACIÓN</b>
2018
<b>DESCRIPCIÓN</b>
<p>Según Casseres et al. (2018), este proyecto se centra en la implementación de un chatbot para facilitar las consultas más frecuentes realizadas por estudiantes en la Facultad de Ingeniería de la Universidad Simón Bolívar. El sistema utiliza inteligencia artificial para procesar preguntas en lenguaje natural y proporcionar respuestas en tiempo real, optimizando el tiempo de los funcionarios y mejorando la precisión de la información.</p> <p>Comparado con nuestro proyecto, compartimos el objetivo de automatizar procesos administrativos y mejorar la interacción entre los usuarios y la institución. Sin embargo, nuestro chatbot amplía su alcance al incluir funcionalidades específicas como la gestión de permisos y consultas avanzadas, personalizando las respuestas según el rol del usuario</p>

(administrador, docente o estudiante). Además, nuestro diseño busca ser adaptable a diferentes instituciones educativas, promoviendo una mayor escalabilidad.

<b>TÍTULO</b>
Cliobot, un chatbot de convicción en las aulas: Un abordaje de la enseñanza de la historia a través del diseño de herramientas digitales interactivas
<b>AUTOR(ES)</b>
Castillo Neira, Liliana
<b>PUBLICACIÓN</b>
2024
<b>DESCRIPCIÓN</b>
<p>Según Castillo Neira (2024), Cliobot es un chatbot diseñado para abordar la enseñanza de la historia mediante herramientas digitales interactivas. Este proyecto explora el uso de inteligencia artificial y aprendizaje automático para crear una experiencia educativa que fomente la conexión emocional y la comunicación profunda, buscando nuevas formas de interacción educativa en entornos culturales y académicos. Aunque los chatbots en este ámbito están en una etapa temprana de desarrollo, el proyecto destaca por su enfoque innovador en el uso de la tecnología en la educación histórica.</p> <p>Comparado con nuestro proyecto, ambos comparten el uso de inteligencia artificial para mejorar los procesos educativos. Sin embargo, mientras Cliobot se enfoca en la enseñanza específica de la historia mediante interacciones emocionales, nuestro chatbot se orienta a la optimización de procesos administrativos y operativos en instituciones educativas. Nuestro</p>

diseño busca integrar funcionalidades como la gestión de permisos y la personalización según los roles de los usuarios, ampliando su aplicabilidad más allá de un área académica específica.

<b>TÍTULO</b>
Diseño de Prototipo de Chat Bot para la Orientación en la Selección de una Carrera de Pregrado de la Facultad de Ingeniería de la Universidad Distrital Francisco José De Caldas
<b>AUTOR(ES)</b>
Ballesteros Vargas, Elkin Humberto Buitrago Restrepo, Camilo Esteban
<b>PUBLICACIÓN</b>
2018
<b>DESCRIPCIÓN</b>
<p>Según Ballesteros Vargas y Buitrago Restrepo (2018), este proyecto propone un chatbot para reducir la deserción de estudiantes en los primeros semestres de la Facultad de Ingeniería. El chatbot utiliza procesamiento de lenguaje natural (PLN) para responder preguntas frecuentes y orientar a los estudiantes en la selección de sus carreras, ofreciendo información clara y precisa mediante una interfaz interactiva.</p> <p>Comparado con nuestro proyecto, ambos comparten el uso de chatbots para mejorar procesos educativos, aunque el enfoque de este trabajo es exclusivamente académico, centrado en la orientación vocacional. Nuestro chatbot, en cambio, se orienta a la automatización de procesos administrativos, como la gestión de permisos y consultas operativas dentro de las instituciones educativas. Además, nuestro diseño busca ser escalable</p>

y adaptable para diversas instituciones, personalizando las interacciones según el rol del usuario.

<b>TÍTULO</b>
Uso de chatbots como apoyo para la comunicación en el aula
<b>AUTOR(ES)</b>
Batista, Alejandro Roberto
<b>PUBLICACIÓN</b>
2017
<b>DESCRIPCIÓN</b>
<p>Según Batista (2017), este trabajo analiza el uso de chatbots como herramientas de apoyo en la comunicación educativa. Los chatbots permiten a los docentes interactuar con los estudiantes de manera autónoma mediante redes sociales, facilitando la organización de respuestas a consultas frecuentes y promoviendo discusiones sobre conceptos clave de los cursos. Este enfoque mejora la eficiencia comunicativa en comparación con métodos tradicionales como el correo electrónico.</p> <p>Comparado con nuestro proyecto, este trabajo se centra exclusivamente en la comunicación docente-estudiante en el aula, mientras que nuestro chatbot amplía su alcance a procesos administrativos dentro de las instituciones educativas. Además, nuestro diseño busca integrar roles diferenciados y automatizar la gestión de permisos y consultas institucionales, ofreciendo una solución más integral y personalizada.</p>

<b>TÍTULO</b>
Implementación de la Inteligencia Artificial (IA) como Recurso Educativo
<b>AUTOR(ES)</b>
Peñaherrera Acurio, W. P., Cunuhay Cuchiye, W. C., Nata Castro, D. J., & Moreira Zamora, L. E.
<b>PUBLICACIÓN</b>
2022
<b>DESCRIPCIÓN</b>
<p>Según Peñaherrera Acurio et al. (2022), este artículo analiza cómo disciplinas como el Machine Learning, el Deep Learning y el Procesamiento de Lenguaje Natural (PLN) permiten automatizar y personalizar procesos educativos. Estas tecnologías facilitan el manejo y organización de grandes volúmenes de datos, ayudando a adaptar las trayectorias educativas</p>

a las características individuales de los estudiantes. Además, se exploran los beneficios de la IA en la mejora de los procesos de enseñanza, aprendizaje y gestión educativa, destacando su papel fundamental en la renovación y modernización del sector educativo.

Comparado con nuestro proyecto, compartimos el uso de inteligencia artificial para automatizar procesos y mejorar la administración en el contexto educativo. Sin embargo, nuestro enfoque está específicamente en la optimización de procesos administrativos, como la gestión de permisos y consultas avanzadas, integrando roles diferenciados para una personalización más amplia. Nuestro diseño busca ser una solución específica y escalable para instituciones educativas.

<b>TÍTULO</b>
Propuesta de mejora en el proceso de atención estudiantil, mediante un chatbot en la Escuela Superior Politécnica de Chimborazo sede Morona Santiago, 2022
<b>AUTOR(ES)</b>
Flores Orozco, Angel Patricio, Gómez García, Diana Elizabeth
<b>PUBLICACIÓN</b>
2022
<b>DESCRIPCIÓN</b>
Según Flores Orozco y Gómez García (2022), este proyecto busca mejorar la atención estudiantil mediante un chatbot que automatiza respuestas a solicitudes individuales. Se identificó una tardanza significativa en la respuesta a las consultas de los estudiantes, lo que generaba insatisfacción. La implementación del chatbot permitió establecer procesos más

eficientes, reducir los tiempos de respuesta y proporcionar un mecanismo de control para garantizar el cumplimiento en la atención de solicitudes.

Comparado con nuestro proyecto, ambos comparten el objetivo de optimizar procesos administrativos en el ámbito educativo. Sin embargo, mientras este proyecto se enfoca en resolver problemas de atención estudiantil, nuestro diseño incluye una gestión integral de procesos administrativos, como permisos y consultas institucionales, adaptándose a roles diferenciados (estudiantes, docentes y administrativos). Además, nuestro chatbot busca ser escalable y aplicable a distintas instituciones educativas.

<b>TÍTULO</b>
Propuesta de un CHATBOT para el manejo de Información Académica recurrente de la Universidad Estatal de Milagro
<b>AUTOR(ES)</b>
León Granizo, Oscar Darío Pazmiño Muguersa, Dennisse Alejandra
<b>PUBLICACIÓN</b>
2021
<b>DESCRIPCIÓN</b>
<p>Según León Granizo y Pazmiño Muguersa (2021), este proyecto propone un chatbot para facilitar la comunicación entre estudiantes y la Universidad Estatal de Milagro, respondiendo dudas sobre las carreras ofrecidas y simplificando el flujo comunicativo entre docentes, estudiantes y la institución. Con apoyo de inteligencia artificial, el chatbot responde de manera personalizada, promueve la interacción efectiva y ahorra recursos, especialmente útil en tiempos de pandemia para la educación semipresencial.</p> <p>Comparado con nuestro proyecto, ambos comparten la intención de optimizar procesos</p>

educativos mediante la tecnología de chatbots. Sin embargo, nuestro diseño se enfoca en procesos administrativos como la gestión de permisos y consultas operativas, adaptándose a roles diferenciados y ampliando el alcance más allá de la comunicación académica. Además, nuestro chatbot está pensado para ser escalable y replicable en diferentes instituciones educativas.

<b>TÍTULO</b>
Chatbot para mejorar la interacción académico-virtual de los estudiantes en el area de arte y cultura, I. E. Túpac Amaru II, 2022
<b>AUTOR(ES)</b>
Gil Campoverde, Luis Angel
<b>PUBLICACIÓN</b>
2022
<b>DESCRIPCIÓN</b>
<p>Según Gil Campoverde (2022), este proyecto empleó un enfoque cuantitativo y preexperimental para evaluar el impacto de un chatbot en la interacción académico-virtual en el área de arte y cultura. Los resultados mostraron un incremento del 146% en la cantidad de consultas realizadas, un aumento del 54.74% en la satisfacción de los estudiantes con el acceso a su material de estudio y un 16.24% en la entrega de evidencias de aprendizaje. Este estudio concluyó que el uso del chatbot tiene un efecto positivo en la educación, especialmente en el área de arte y cultura.</p> <p>Comparado con nuestro proyecto, ambos comparten el objetivo de mejorar procesos educativos mediante chatbots, aunque el trabajo de Campoverde se centra en el ámbito</p>

académico-cultural. Nuestro diseño se orienta hacia la gestión administrativa, incluyendo la automatización de permisos y consultas institucionales, con personalización según los roles de los usuarios y una escalabilidad más amplia para diferentes instituciones educativas.

<b>TÍTULO</b>
Desarrollar una solución chatbot para comunicar de manera eficiente la información de la Universidad Ricardo Palma al sector universitario utilizando Facebook Messenger
<b>AUTOR(ES)</b>
Sánchez Hurtado, Renzo André Uriarte Flores, Johan Alexander
<b>PUBLICACIÓN</b>
2021
<b>DESCRIPCIÓN</b>
Según Sánchez Hurtado y Uriarte Flores (2021), este proyecto desarrolla un chatbot para abordar las deficiencias en la comunicación de la Universidad Ricardo Palma con los estudiantes. Utilizando Facebook Messenger, el chatbot responde consultas sobre servicios educativos, apoyo al estudiante, admisión y otros datos relevantes, mejorando la eficiencia en la distribución de información. Además, el sistema genera reportes para analizar las necesidades de los usuarios interesados en los servicios universitarios.
Comparado con nuestro proyecto, ambos comparten el uso de chatbots para optimizar la

interacción entre instituciones educativas y sus usuarios. Sin embargo, mientras este proyecto se centra en la comunicación y generación de reportes, nuestro diseño incluye funcionalidades administrativas más amplias, como la gestión de permisos y consultas complejas, adaptándose a roles diferenciados en la institución. Además, nuestro chatbot está diseñado para ser escalable y aplicable en diferentes contextos educativos.

<b>TÍTULO</b>
Proyecto de mejora del proceso de gestión de documentos y consultas escolares mediante el uso de asistente virtual (ChatBot) con inteligencia artificial
<b>AUTOR(ES)</b>
Lino Noboa, Jaime Fabian
<b>PUBLICACIÓN</b>
2023
<b>DESCRIPCIÓN</b>
<p>Según Lino Noboa (2023), este proyecto busca resolver las limitaciones de la administración de documentos y consultas escolares mediante un chatbot con inteligencia artificial. El sistema automatiza tareas como la consulta de notas, horarios de clases y certificados, agilizando procesos y reduciendo tiempos de respuesta. Al implementar el chatbot en plataformas familiares como WhatsApp, se asegura una mayor accesibilidad y facilidad de uso, mejorando la experiencia para estudiantes y docentes por igual.</p> <p>Comparado con nuestro proyecto, ambos comparten el objetivo de optimizar procesos administrativos mediante chatbots, aunque este trabajo se centra en la consulta de documentos específicos. Nuestro proyecto, además, amplía su enfoque al incluir permisos y consultas operativas, personalizando las respuestas según roles diferenciados. También</p>

consideramos la escalabilidad para implementar el chatbot en múltiples instituciones educativas.

<b>TÍTULO</b>
Implementación de un chatbot integrado a un LMS, como asistente para la gestión educativa
<b>AUTOR(ES)</b>
Pazmiño Ricaurte, Julio David
<b>PUBLICACIÓN</b>
2020
<b>DESCRIPCIÓN</b>
<p>Según Pazmiño Ricaurte (2020), este proyecto desarrolló un chatbot integrado en una plataforma de gestión educativa (LMS) para apoyar la administración y enseñanza dentro del aula virtual de la Universidad de las Américas (Quito, Ecuador). Usando herramientas como Visual Studio 2019, Microsoft Bot Framework SDK y el servicio cognitivo LUIS (Language Understanding Intelligent Service), el chatbot proporcionó inteligencia artificial para gestionar la interacción con estudiantes y docentes. El sistema fue evaluado en un entorno real, mostrando mejoras en la experiencia educativa mediante la automatización de tareas.</p> <p>Comparado con nuestro proyecto, ambos utilizan chatbots para optimizar procesos educativos. Sin embargo, mientras este proyecto se centra en la integración con plataformas</p>

LMS para apoyar el aprendizaje, nuestro chatbot amplía su alcance a la automatización de procesos administrativos, como permisos y consultas institucionales, además de personalizar las respuestas según roles diferenciados. Nuestro enfoque también prioriza la escalabilidad para diferentes instituciones educativas.

<b>TÍTULO</b>
Percepción del modelo CRM y la gestión educativa en la comunicación con estudiantes de educación secundaria pública, 2021
<b>AUTOR(ES)</b>
Bullon Miguel, Yeni Carmen
<b>PUBLICACIÓN</b>
2022
<b>DESCRIPCIÓN</b>
Según Bullón Miguel (2022), este estudio analiza la relación entre el modelo CRM (Customer Relationship Management) y la gestión educativa en instituciones públicas. Con una muestra de 194 estudiantes de educación secundaria, se evaluó cómo la implementación de un chatbot basado en CRM puede mejorar la comunicación entre estudiantes y la institución educativa. El estudio empleó un enfoque cuantitativo y correlacional-causal, utilizando cuestionarios con escalas Likert para evaluar la percepción de los estudiantes sobre el impacto del CRM. Los resultados muestran una correlación positiva significativa entre el uso del CRM y la eficacia en la comunicación educativa.
Comparado con nuestro proyecto, este trabajo también aborda la mejora de la comunicación

institucional mediante tecnología innovadora. Sin embargo, mientras este estudio se enfoca en la percepción y evaluación del modelo CRM, nuestro diseño se orienta a la automatización de procesos administrativos específicos, como la gestión de permisos y consultas avanzadas, adaptándose a roles diferenciados dentro de las instituciones educativas. Además, nuestro chatbot busca ser una herramienta escalable y replicable en distintos contextos educativos.

<b>TÍTULO</b>
La Inteligencia Artificial al servicio de la gestión y la implementación en la educación
<b>AUTOR(ES)</b>
Isusqui, J. C. P., Villavicencio, I. E. S., Inga, C. V., Gutiérrez, H. O. C., Díaz, B. L. G., & Amaya, K. L. A.
<b>PUBLICACIÓN</b>
2023
<b>DESCRIPCIÓN</b>
Según Isusqui et al. (2023), la inteligencia artificial (IA) ha transformado la administración educativa al automatizar labores repetitivas, como la clasificación de documentos, control de asistencia, calificaciones y programación de tareas. Estas herramientas liberan a los educadores para enfocarse en la enseñanza y permiten personalizar la educación al adaptar materiales a las necesidades de los estudiantes, mejorando así el rendimiento académico y el aprendizaje. Sin embargo, el estudio también advierte sobre preocupaciones en la privacidad y seguridad de los datos, destacando la necesidad de medidas robustas para proteger información sensible.

Comparado con nuestro proyecto, ambos comparten el uso de la inteligencia artificial para mejorar procesos en el ámbito educativo. No obstante, nuestro enfoque está dirigido a la automatización de procesos administrativos específicos, como la gestión de permisos y consultas institucionales, con un diseño que integra roles diferenciados (docentes, estudiantes y administrativos). Además, buscamos garantizar la escalabilidad y la adaptabilidad del chatbot en diversas instituciones educativas, alineándonos también con las preocupaciones de seguridad y privacidad de los datos.

## 1.2. LÍNEA DE INVESTIGACIÓN

**Inteligencia Artificial y Procesamiento del Lenguaje Natural (PLN):** Este proyecto se ubica en esta línea, ya que su diseño, desarrollo e implementación final están vinculados a técnicas, herramientas y metodologías características de estas áreas de la computación avanzada, entre las que sobresalen:

- A. **Uso de métodos de procesamiento del lenguaje natural (PLN):** Se establecieron modelos personalizados de PLN formados a partir de intentos institucionales, lo que posibilita que el chatbot entienda, atienda y produzca respuestas automáticas de forma contextualizada, optimizando la interacción natural con los usuarios de acuerdo a su función (estudiante, profesor, administrativo o administrador).
- B. **Incorporación de algoritmos artificiales:** El sistema emplea métodos de aprendizaje automático para capacitar su motor para identificar intenciones. Esta integración posibilita que el chatbot no solo atienda preguntas habituales, sino que también ejecuta tareas automatizadas como revisar horarios, registrar peticiones o administrar información de usuarios, mejorando así la eficiencia en las operaciones.
- C. **Capacidad de modulación y estructura escalable:** El proyecto se desarrolló siguiendo los fundamentos de la modularidad, segmentando el sistema en elementos autónomos: capacitación de modelos NLP, gestión de consultas, comprobación de seguridad y autenticación de usuarios. Esta estructura simplifica su conservación, renovación y potencial desarrollo futuro hacia nuevas funcionalidades o plataformas.
- D. **Técnicas ágiles para el desarrollo y la formación iterativa:** Se utilizó un método de desarrollo ágil (Design Thinking) que facilita la transición rápida entre las etapas de diseño, prueba y corrección del chatbot, modificando constantemente los modelos de Inteligencia Artificial según las necesidades identificadas en los usuarios institucionales.

Este enfoque asegura que el proyecto no solo satisfaga los principios esenciales de la línea de investigación en Inteligencia Artificial y PLN, sino que también brinde una respuesta auténtica, escalable y sostenible a las demandas de automatización y mejora de procesos en el sector educativo.

### 1.3. PLANTEAMIENTO DEL PROBLEMA Y PREGUNTA DE INVESTIGACIÓN

En los diferentes tipos de gestión del campo institucional, se pueden manifestar problemas de comunicación, descoordinación y falta de acceso eficiente a la información, lo que provoca retrasos en la toma de decisiones, una mayor carga administrativa y la insatisfacción de los usuarios. La ausencia de un sistema que gestione de manera eficaz los permisos y accesos, y que automatice la resolución de consultas, agrava esta situación, impactando negativamente la calidad del servicio que se presta a estudiantes, docentes y personal administrativo.

Actualmente, los sistemas de información y comunicación en las instituciones educativas son limitados y no integran las capacidades avanzadas que proporciona la inteligencia artificial (IA) para mejorar los procesos de estas mismas. Esto genera que el personal encargado se enfrente a la duplicación de tareas, procesos manuales y a un uso ineficiente de los recursos disponibles. Como resultado, la resolución de consultas y la gestión de permisos o accesos puede llevar días o incluso semanas, afectando directamente la productividad de la institución y el bienestar de la comunidad educativa.

La ineficiencia en la gestión de permisos, accesos y consultas internas en las instituciones educativas puede ser atribuida a múltiples factores que afectan la operatividad y la calidad del servicio. Según un estudio realizado por García et al. (2021) en donde las instituciones educativas enfrentan desafíos significativos relacionados con la falta de integración tecnológica, que limita la capacidad de los administradores para gestionar eficientemente los flujos de información. Este estudio destaca que muchos sistemas actuales son obsoletos y no están diseñados para facilitar la comunicación interdepartamental, lo que resulta en demoras en la toma de decisiones y una carga administrativa excesiva.

Por lo que el problema que se aborda en este proyecto se relaciona con la ineficiencia en la gestión de información y procesos dentro de las instituciones educativas, donde no se dispone de herramientas automatizadas que agilicen la comunicación y los flujos de trabajo. Esto genera un contexto en el que las decisiones importantes se toman con demoras significativas, los errores humanos son frecuentes en los estudiantes y demás usuarios reciben una atención inadecuada.

De acuerdo al informe de la UNESCO (2020), muchas instituciones educativas carecen de programas adecuados de formación que permitan a los docentes y al personal administrativo utilizar eficientemente las herramientas tecnológicas disponibles. Esta carencia no solo limita la adopción de sistemas avanzados, sino que también perpetúa procesos manuales que son ineficaces y propensos a errores.

Pregunta concreta a responder:

**¿Cuáles son los factores clave que causan ineficiencia en la gestión de permisos, accesos y consultas internas en las instituciones educativas, y cómo se pueden abordar mediante el desarrollo de un sistema inteligente que use técnicas de inteligencia artificial y procesamiento de lenguaje natural?**

Aporte del proyecto:

Este proyecto contribuirá al desarrollo de nuevo conocimiento en la aplicación de inteligencia artificial y procesamiento de lenguaje natural (NLP) dentro del contexto educativo, al proponer un sistema que no solo mejora los flujos de trabajo y la toma de decisiones, sino que también automatiza la identificación de roles y la resolución de consultas frecuentes. El proyecto proporcionará un modelo escalable que podrá ser replicado en otras instituciones educativas, resolviendo problemas similares a nivel nacional y regional.

Además, se generará nuevo conocimiento sobre la integración de IA en los procesos de gestión de instituciones educativas, optimizando la interacción entre los diferentes actores (administrativos, docentes, estudiantes). El sistema propuesto no solo reducirá la carga administrativa y los tiempos de espera, sino que también permitirá a las instituciones educativas adaptarse a entornos dinámicos y exigentes, incrementando la satisfacción de los usuarios y mejorando la productividad de la institución.

#### **1.4. OBJETIVO GENERAL Y OBJETIVOS ESPECÍFICOS**

##### **Objetivo General:**

Desarrollar un sistema inteligente para la optimización de procesos en instituciones educativas que automatice la gestión de permisos, accesos y consultas frecuentes, utilizando técnicas de inteligencia artificial y procesamiento de lenguaje natural (NLP), con el fin de mejorar la eficiencia operativa, reducir los tiempos de respuesta y aumentar la satisfacción de los usuarios.

##### **Objetivos Específicos:**

1. Analizar los requerimientos, formas de comunicación y medidas de seguridad en instituciones educativas para identificar ineficiencias importantes, definir módulos funcionales del sistema y optimizar la gestión de permisos y resolución de consultas frecuentes, garantizando la protección de datos sensibles.
2. Desarrollar un asistente virtual inteligente integrado dentro del aplicativo, utilizando técnicas de procesamiento de lenguaje natural (NLP) para automatizar las respuestas a consultas

frecuentes. El aplicativo centralizará el acceso al chatbot, reduciendo tiempos de respuesta y asegurando el manejo seguro de datos personales.

3. Incorporar en el sistema un mecanismo de redirección de consultas complejas a las áreas o departamentos correspondientes, garantizando una atención eficiente y segura para aquellas solicitudes que necesitan intervención humana.
4. Realizar pruebas en el prototipo de **chat inteligente** para verificar el funcionamiento del sistema simulando los procesos esperados dentro de los parámetros establecidos.

## 1.5. ALCANCE E IMPACTO DEL PROYECTO

El desarrollo e implementación del **Chat Inteligente** en la institución educativa, moderniza los servicios académicos y administrativos brindados. Este proyecto no solo transforma la manera en que estudiantes, docentes y administrativos interactúan entre sí, sino que también introduce un modelo de gestión automatizada más eficiente, segura y escalable.

### Impactos principales:

- **Mejora de la comunicación institucional:**  
El chat inteligente permite una atención inmediata, personalizada y disponible para resolver consultas frecuentes, realizar trámites internos y gestionar información académica y administrativa de manera ágil.
- **Optimización de procesos:**  
Al automatizar acciones como consulta de horarios, calificaciones, datos personales, carga de notas, y gestión de usuarios, se reduce considerablemente la carga operativa del personal administrativo y se minimizan los tiempos de espera de los usuarios.
- **Seguridad y control de acceso:**  
La implementación de mecanismos como la verificación de identidad en el registro, autenticación reforzada (con pregunta de seguridad y verificación por correo), cifrado de contraseñas y roles de usuario controlados garantiza que la información sensible esté protegida y solo accesible para usuarios autorizados.
- **Inclusión tecnológica:**  
El proyecto facilita la interacción de la comunidad académica seleccionada (administrativos, docentes y estudiantes) con herramientas tecnológicas intuitivas, promoviendo habilidades

digitales y reduciendo barreras de acceso a la información.

- **Escalabilidad y sostenibilidad:**

Gracias a su arquitectura basada en tecnologías modernas (FastAPI, Flutter, almacenamiento estructurado en JSON), el sistema está diseñado para crecer fácilmente, adaptarse a nuevas necesidades y ser mantenido a largo plazo con bajo costo.

- **Experiencia de usuario mejorada:**

El enfoque en una interfaz amigable, tiempos de respuesta rápidos y flujo conversacional natural mejora la satisfacción de los usuarios y fomenta su uso continuo.

- **Impacto académico-administrativo positivo:**

Al agilizar procesos críticos como el registro, la recuperación de cuentas y la gestión de notas o materias, se contribuye a un entorno educativo más organizado, eficiente y adaptado a las necesidades actuales.

### **Alcance del Proyecto:**

Este proyecto busca desarrollar una herramienta tecnológica que facilite el día a día en las instituciones educativas. A través de un sistema de chat inteligente, se pretende optimizar procesos internos como la solicitud de permisos, el acceso a información institucional y la resolución de dudas frecuentes, que hoy en día suelen tomar tiempo y recursos humanos valiosos. La solución está pensada para ser útil tanto para estudiantes como para docentes y personal administrativo, brindando atención inmediata.

El chat inteligente será capaz de responder preguntas comunes y de guiar a los usuarios según su rol dentro del aplicativo. Además, incluirá un panel para los administradores, desde donde podrán gestionar los permisos solicitados, aprobar usuarios y mantener actualizado el sistema. Todo esto funcionará a través de una aplicación móvil, adaptada a las necesidades reales de la comunidad educativa.

Aunque el proyecto se apoya en tecnologías como inteligencia artificial y procesamiento de lenguaje natural, el enfoque no está en la complejidad técnica, sino en resolver problemas cotidianos de forma sencilla, rápida y segura. No se trata de reemplazar a las personas, sino de ofrecer una herramienta que les permita enfocarse en tareas más importantes, dejando que el sistema se encargue de lo repetitivo.

Este desarrollo no busca sustituir plataformas académicas ni encargarse de temas financieros o de gestión de calificaciones. Su propósito es claro: hacer más ágil y eficiente la gestión de solicitudes y consultas en el entorno institucional. Además, está diseñado para ser adaptable, por lo que en el

futuro podría adoptar diferentes tipos de rol y subrol e implementarse en otras instituciones educativas del país que compartan necesidades similares.

### Tabla de Alcance del Proyecto

Tabla 1. Alcance del Proyecto. fuente propia

Categoría	Descripción
Nombre del Proyecto	Sistema Inteligente para la Optimización de Procesos Educativos
Tipo de solución	Chatbot multirol con IA y procesamiento de lenguaje natural (PLN)
Usuarios	Estudiantes, Docentes, Administrativos, Administrador
Plataformas	Aplicación móvil (Flutter) y API backend (FastAPI)
Funcionalidades incluidas	Registro y login seguro, gestión de permisos y roles, chatbot por rol, panel de administración, reportes, verificación por correo y pregunta de seguridad
Tecnologías utilizadas	Flutter, FastAPI, JSON, TensorFlow, JWT, bcrypt, correo SMTP
Ámbitos cubiertos	Automatización de consultas, permisos, redirección de solicitudes, gestión interna educativa
Seguridad	Autenticación con token, cifrado, validación de identidad
Alcance geográfico	Instituciones educativas locales y regionales (escalable a nivel nacional)
Escalabilidad	Alta. Diseño modular preparado para integración con LMS u otros sistemas

### Funcionalidades fuera del alcance

Tabla 2 Funcionalidades fuera del alcance .fuente propia

Exclusiones	Motivo
-------------	--------

Gestión financiera o transacciones	No corresponde al objetivo administrativo del chatbot
Matrículas o inscripción de asignaturas	Son procesos propios de los sistemas académicos existentes
Calificaciones académicas o control de rendimiento escolar	Fuera del foco de automatización administrativa y consultas frecuentes
Funcionalidades offline	Requiere conexión a internet por diseño estructural del sistema

## 1.6. METODOLOGÍA

### Metodología Propuesta

La metodología propuesta para este proyecto se divide en dos enfoques clave: la **metodología de investigación** y la **metodología de desarrollo de software**. Esto garantiza un enfoque riguroso en la recopilación y análisis de datos para comprender los problemas actuales de las instituciones educativas, mientras se utiliza una metodología ágil para el desarrollo iterativo del sistema inteligente.

#### 1.6.1 METODOLOGÍA DE INVESTIGACIÓN MIXTA

Este proyecto emplea una **metodología mixta**, que incluye tanto enfoques cualitativos como cuantitativos, con el fin de obtener una comprensión integral de los flujos de comunicación y gestión dentro de las instituciones educativas. Como lo menciona Hernandez (2013) La metodología mixta, que combina enfoques cualitativos y cuantitativos, es esencial para obtener una comprensión integral de los flujos de comunicación y gestión dentro de las instituciones educativas. Este enfoque permite aprovechar las fortalezas de ambos métodos y mitigar sus debilidades, proporcionando una visión holística y detallada de los procesos y dinámicas involucradas.

- **Cualitativa:**
  - Se llevó a cabo un **estudio** documental enfocado en el estudio de proyectos parecidos llevados a cabo en contextos institucionales y administrativos. Esta revisión permitió detectar requerimientos habituales, obstáculos y aspectos críticos en los procedimientos de administración de permisos, accesos y consultas.
  - El método **cualitativo** permitió recoger las experiencias registradas y las percepciones documentadas en otras investigaciones, obteniendo todo lo requerido para guiar el progreso del proyecto.
- **Cuantitativa:**

- A partir de evaluaciones internas efectuadas durante el desarrollo y validación de la aplicación obtenemos resultados medibles, donde se evaluó el rendimiento del sistema, los tiempos de respuesta, la eficacia en la solución de consultas y el rendimiento operativo global. Esta valoración facilitó la definición de un marco inicial para detectar las mejoras requeridas.
- **Resultados esperados de la investigación:**
  - Un análisis detallado que identifique las **principales ineficiencias** en los procesos de gestión interna.
  - Un **documento de requisitos** que sirva de base para el desarrollo del sistema inteligente, con un enfoque en la mejora de la eficiencia operativa y la satisfacción del usuario.

## 1.6.2. Design Thinking

### Metodología para el Desarrollo del Software

El desarrollo del prototipo seguirá el enfoque de **Design Thinking**, priorizando la experiencia del usuario, la eficiencia operativa y la seguridad. Este enfoque iterativo permite identificar y resolver problemas desde la perspectiva de los usuarios, asegurando que el sistema cumpla con sus necesidades funcionales y respete los estándares básicos de protección de datos.

### Fases de Design Thinking

#### 1. Empatizar:

- **Propósito:** Comprender las necesidades y retos que enfrentan los usuarios finales (estudiantes, docentes, administrativos) mediante observación directa.
- **Actividades:**
  - Identificación de expectativas en términos de funcionalidad y percepción de seguridad.
  - Detección de flujos de información sensible, como datos personales y académicos, para su posterior protección.

#### 2. Definir:

- **Propósito:** Sintetizar los hallazgos obtenidos para establecer un problema central y determinar los requisitos funcionales y no funcionales del sistema.
- **Actividades:**
  - Definición de roles principales (usuario final y administrador) y sus interacciones.

- Redacción de requisitos funcionales (automatización de consultas, roles diferenciados) y no funcionales (usabilidad, tiempos de respuesta).
- Establecimiento de lineamientos básicos de seguridad, como autenticación y protección de datos sensibles, sin comprometer la facilidad de uso.

### 3. Idear:

- **Propósito:** Proponer soluciones creativas, seleccionando tecnologías adecuadas y definiendo funcionalidades clave.
- **Actividades:**
  - Diseño preliminar del chatbot y su integración en prototipo viable.
  - Selección de tecnologías que soportan funcionalidades clave (procesamiento de lenguaje natural, bases de datos en archivos .json) y ofrezcan características básicas de seguridad.
  - Priorización de características según impacto y factibilidad, asegurando un balance entre eficiencia, seguridad y experiencia del usuario.

### 4. Prototipar:

- **Propósito:** Desarrollar un prototipo funcional inicial, con interfaces intuitivas y validando las funcionalidades fundamentales.
- **Actividades:**
  - Desarrollo del chatbot para responder consultas frecuentes y gestionar roles diferenciados.
  - Implementación de medidas básicas de seguridad, como encriptación de datos y autenticación por credenciales.
  - Diseño de una interfaz intuitiva que permita a los usuarios interactuar fácilmente con el sistema.

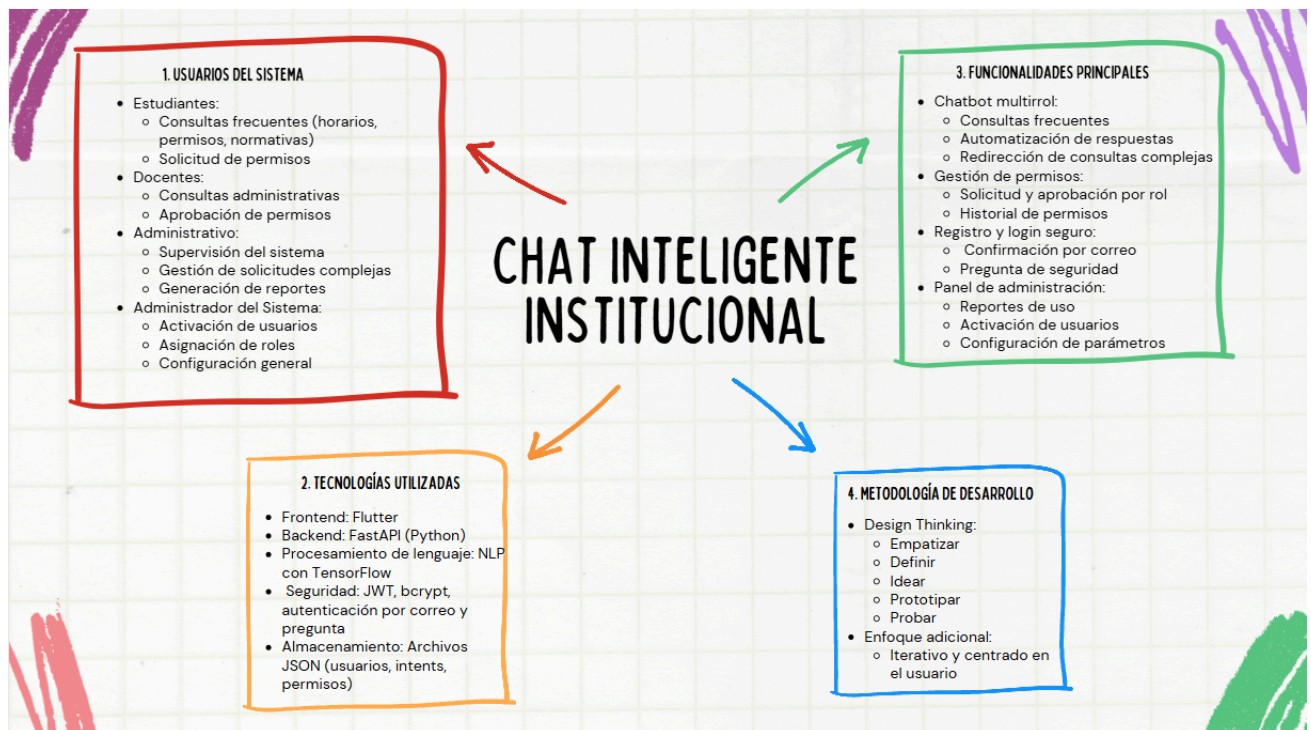
### 5. Probar:

- **Propósito:** Evaluar el prototipo en un entorno controlado, asegurando su funcionalidad, usabilidad y confiabilidad.
- **Actividades:**
  - Pruebas piloto con usuarios de una institución educativa (Universidad De Cundinamarca) para medir tiempos de respuesta, precisión de las respuestas del chatbot y satisfacción general.
  - Revisión de las medidas de seguridad implementadas para confirmar que protegen adecuadamente los datos de los usuarios.
  - Ajustes basados en retroalimentación y resultados de pruebas, refinando tanto la experiencia del usuario como las medidas de seguridad.

### Resultados esperados del Desarrollo del Software:

- Un prototipo funcional que automatice la gestión de permisos y consultas, combinando una experiencia de usuario intuitiva con medidas de seguridad.
- Validación de la solución mediante pruebas piloto, garantizando su alineación con los objetivos del proyecto y las expectativas de los usuarios.
- Documentación clara del proceso y recomendaciones para futuras mejoras o escalabilidad del sistema.

Tabla 3. Chat inteligente institucional.fuente propia



.Recuperado

de: [https://www.canva.com/design/DAGnALcokz4/7KtOu\\_2rJYnQjEbnvx\\_o2w/view?utm\\_content=DAGnALcokz4&utm\\_campaign=designshare&utm\\_medium=link2&utm\\_source=uniquelinks&utlId=h75793d27a8](https://www.canva.com/design/DAGnALcokz4/7KtOu_2rJYnQjEbnvx_o2w/view?utm_content=DAGnALcokz4&utm_campaign=designshare&utm_medium=link2&utm_source=uniquelinks&utlId=h75793d27a8)

El mapa de empatía fue construido a partir del análisis de las experiencias, necesidades y dificultades de los distintos perfiles de usuarios involucrados en el sistema (estudiantes, docentes y administrativos). Esta herramienta permitió entender, desde una perspectiva más humana, qué piensan, sienten, ven, escuchan y hacen los usuarios frente a los procesos institucionales actuales. A partir de este ejercicio, se identificaron puntos críticos como la lentitud en las gestiones, la falta de claridad en los canales de comunicación y la necesidad de contar con respuestas inmediatas. Este

insumo fue clave durante la fase de **empatizar** del modelo Design Thinking, ya que permitió construir un sistema alineado con las expectativas reales del entorno educativo.

Tabla 4. Usuario. fuente propia



. Recuperado

de: [https://www.canva.com/design/DAGnAIIAis/JtDOnCgoh1pysDFpBOEDBA/view?utm\\_content=DAGnAIIAis&utm\\_campaign=designshare&utm\\_medium=link2&utm\\_source=uniquelinks&utm\\_id=hb8b31fcaa1](https://www.canva.com/design/DAGnAIIAis/JtDOnCgoh1pysDFpBOEDBA/view?utm_content=DAGnAIIAis&utm_campaign=designshare&utm_medium=link2&utm_source=uniquelinks&utm_id=hb8b31fcaa1)

El mapa mental resume de forma estructurada y jerárquica los componentes esenciales del sistema propuesto. Esta representación gráfica facilitó la organización de ideas y funcionalidades desde las etapas iniciales del diseño del prototipo. A través de sus ramas principales —usuarios, tecnologías, funcionalidades, metodología y resultados esperados—, se establece una visión general del proyecto, permitiendo visualizar tanto su alcance técnico como su impacto en la comunidad académica. Este esquema fue utilizado como guía durante la fase de **definir e idear** de la metodología Design Thinking para asegurar la coherencia entre las soluciones propuestas y los requerimientos reales.

## 1.7. MARCOS DE REFERENCIA

### 1.7.1. MARCO TEÓRICO

Este capítulo enumera los antecedentes teóricos esenciales sobre los cuales se construyó el proyecto, "Chat Inteligente para Instituciones Educativas", empleando tecnologías de IA y PLN.

#### 1.7.1.1. Inteligencia Artificial (IA):

La Inteligencia Artificial (IA) es un área que se ocupa del diseño de sistemas que puedan realizar tareas que, si fueran realizadas por personas, serían descritas como inteligentes. En este proyecto, la IA se aplica para automatizar todos los procesos administrativos y mejorar la interfaz de la escuela o colegio con sus usuarios.

La IA es "la ciencia e ingeniería de crear máquinas inteligentes", según Russell y Norvig (2021), y es el "diseño de agentes inteligentes", sistemas o robots que "actúan para lograr sus objetivos" (Russell & Norvig, 2021).

#### 1.7.1.2. Procesamiento del Lenguaje y Representación del Conocimiento (incluyendo PLN):

El Procesamiento de Lenguaje Natural es un campo de la informática que se ocupa de la interacción entre computadoras y el lenguaje humano. Su tarea principal es leer, descodificar, comprender y dar sentido a los lenguajes humanos de manera valiosa (Cambria y White, 2014).

En el proyecto, el PLN se aplica para comprender las preguntas de los usuarios y generar respuestas automáticas dirigidas a estudiantes, docentes y administradores.

#### 1.7.1.3. Modelo de Investigación Mixto:

La investigación de métodos mixtos combina métodos cualitativos y cuantitativos, aprovechando sus respectivas fortalezas. Este método permite la interpretación de fenómenos complejos mediante análisis cuantitativo (estadístico) y mediante la exploración de percepciones o experiencias (cualitativo).

La investigación de métodos mixtos es "un enfoque de investigación que incluye o se basa en la mezcla o combinación de métodos de investigación cuantitativos y cualitativos", como usar entrevistas para interpretar los resultados de un cuestionario poblacional (Creswell, 2014, p. 12).

#### 1.7.1.4. Marcos a Nivel de Aplicación (FastAPI, Flutter, TensorFlow):

- **FastAPI:** Un marco web moderno y rápido para construir APIs con Python, con enfoque en herramientas y protocolos estándar, incluido OpenAPI (Documentación de FastAPI, 2024).
- **Flutter:** Un kit de interfaz de usuario de código abierto desarrollado por Google para el desarrollo multiplataforma de aplicaciones compiladas nativamente (Documentación de Flutter, 2024).

- **TensorFlow:** Esta es una biblioteca de código abierto para el cálculo numérico y el aprendizaje automático, que se utiliza extensamente para desarrollar y entrenar modelos de PLN (cf. Documentación de TensorFlow, 2024).

## **1.7.2. MARCO LEGAL**

Este sitio web presentará las leyes y licencias que conciernen al proyecto y al uso de las tecnologías y datos implementados.

### **1.7.2.1. Ley de Protección de Información Personal en Colombia (Ley 1581 de 2012)**

La ley introdujo las normas generales para la protección de datos personales en Colombia, asegurando que se mantenga el derecho constitucional de toda persona a conocer, actualizar y rectificar la información que se ha recopilado sobre ellas en bases de datos o archivos (Ley 1581 de 2012, Congreso de Colombia).

### **1.7.2.2. Decreto 1377 de 2013**

Artículo que regula parcialmente la Ley 1581 de 2012 y establece procedimientos especiales para el tratamiento y manejo de datos personales almacenados en bancos de datos; aplicable tanto para el sector público como privado (Resolución 1377 de 2013).

### **1.7.2.3. Licencia de Uso de FastAPI**

FastAPI está licenciada bajo los términos de la Licencia MIT, lo que permite tanto el uso personal como comercial; la licencia está incluida en el archivo (Licencia MIT de FastAPI).

### **1.7.2.4. Licencia de Uso de Flutter**

Flutter es un marco de código abierto, y es de uso gratuito, pero se publica bajo una licencia permisiva BSD (Berkeley Software Distribution), que contiene muy pocas restricciones, y por lo tanto puede ser usado de manera rentable por proyectos de código abierto y organizaciones propietarias para crear aplicaciones de software (Licencia BSD de Flutter).

### **1.7.2.5. Licencia de Uso de TensorFlow**

Es un software de código abierto inicialmente lanzado al público a través de GitHub, bajo la licencia Apache 2.0, que permite a las personas usar, modificar y distribuir el código fuente para propósitos comerciales o no comerciales bajo los términos de la licencia (Licencia Apache 2.0 de TensorFlow).

### 1.7.2.6. LEY N.º 23 DE 1982 DERECHOS DE AUTOR EN COLOMBIA

La Ley 23 de 1982 reconoce a los desarrolladores de software como autores de obras protegidas por derechos de autor, y desde la creación de esas obras los derechos perduran hasta ochenta años después de la muerte del autor (Ley 23 de 1982, Congreso de Colombia).

## II. DOCUMENTACIÓN DEL SOFTWARE

### 2.1. PLAN DE PROYECTO

La evolución del sistema de soporte inteligente para la optimización institucional en contextos educativos se abordará en varias etapas.

- **Análisis y Planificación:**
  - Recopilación de requisitos funcionales y no funcionales a través de observaciones y análisis de la documentación institucional.
  - Definición del cronograma de actividades y los posibles riesgos.
- **Modelado de Arquitectura:**
  - Se modelará la arquitectura del sistema en diagramas UML.
  - Definición de interfaces, estructura de datos y flujos del chatbot.
- **Desarrollo del Sistema:**
  - Desarrollo del backend utilizando FastAPI para usuarios, permisos, autenticación, seguridad, y PLN.
  - Desarrollo paralelo del frontend en Flutter: pantallas para usuarios, administración, chat.
- **Entrenamiento del Modelo de PLN:**
  - Entrenamiento con archivos personalizados de intención basados en roles.
  - Ajuste del comportamiento del chatbot para búsquedas comunes y redirección de las más complejas.
- **Pruebas y Validaciones:**
  - Realización de pruebas unitarias, de integración y funcionales.
  - Validaciones con usuarios reales (estudiantes, profesores, trabajadores administrativos).
- **Futuras Evoluciones:**
  - Integración con plataformas educativas (como LMS) y desarrollo de informes avanzados.
  - Escalabilidad del sistema para su aplicación en otras instituciones educativas.

## 2.2. DETERMINACIÓN DE REQUERIMIENTOS

### 2.2.1.1. Propósito

El propósito de este documento es definir de manera detallada los requisitos funcionales y no funcionales del **Sistema Inteligente para la Optimización de Procesos Institucionales en Entornos Educativos**. Este documento está dirigido a desarrolladores de software, ingenieros de sistemas, administradores de proyectos y otras partes interesadas que deseen participar en el diseño, implementación, pruebas y mantenimiento del sistema.

El sistema tiene como objetivo principal automatizar la gestión de roles, permisos y la resolución de consultas frecuentes en instituciones educativas, utilizando inteligencia artificial (IA) y procesamiento de lenguaje natural (NLP). Al proporcionar una especificación completa, este documento garantiza que el sistema cumpla con los requisitos operativos y de rendimiento esperados, asegurando una simulación exitosa y una mejora en la eficiencia operativa de las instituciones.

### 2.2.1.2. Ámbito del sistema

El **Sistema Inteligente para la Optimización de Procesos Interinstitucionales en Entornos Educativos** está diseñado para mejorar la eficiencia operativa de las instituciones educativas, automatizando procesos clave como la asignación de permisos, la gestión de roles y la resolución de consultas frecuentes. A través de un **chatbot inteligente** basado en técnicas de **inteligencia artificial (IA)** y **procesamiento de lenguaje natural (NLP)**, el sistema proporcionará una interfaz accesible para estudiantes, docentes y personal administrativo.

#### Lo que el sistema hará:

- **Automatización de la gestión de permisos y roles:** Los usuarios podrán solicitar y gestionar permisos y accesos a través de la plataforma, con una reducción significativa en los tiempos de respuesta.
- **Resolución de consultas frecuentes:** el chat inteligente gestionará automáticamente las consultas de los usuarios relacionadas con horarios u otros temas relevantes.
- **Plataforma:** El sistema estará disponible en dispositivos móviles como descargable y para la realización de pruebas por medio de flutter.

#### Lo que el sistema no hará:

- El sistema no maneja directamente transacciones financieras ni procesos de matriculación de estudiantes.
- No incluirá un sistema de calificaciones académicas ni gestión de programas curriculares.

- No actuará como un sistema de gestión de recursos humanos, más allá de la asignación de permisos.

#### **Objetivos del Sistema:**

- **Mejorar la eficiencia operativa:** Reduciendo los tiempos de respuesta mediante la automatización de procesos de consulta y permisos.
- **Optimización de la atención:** Proporcionar una atención rápida y precisa a los usuarios, mejorando la experiencia del personal administrativo, docentes y de los estudiantes.
- **Escalabilidad:** El sistema será diseñado para ser escalable y replicable en otras instituciones educativas, tanto a nivel regional como nacional.

#### **Beneficios:**

- Reducción de la carga administrativa y tiempos de espera para los usuarios.
- Mejora en la toma de decisiones al disponer de información actualizada y precisa.
- Aumento de la satisfacción de los usuarios, incluyendo estudiantes, personal administrativo y docentes.
- Facilitar la integración de tecnologías modernas como IA y NLP en los procesos educativos.

#### **2.2.1.3. Definiciones, Acrónimos y Abreviaturas**

**Sistema Inteligente:** Software que emplea técnicas avanzadas de inteligencia artificial para realizar tareas que normalmente requieren intervención humana, como la toma de decisiones o la automatización de procesos.

**IA (Inteligencia Artificial):** Rama de la informática que permite a las máquinas imitar funciones cognitivas humanas, como el aprendizaje y la resolución de problemas. En este proyecto, la IA se usa para la automatización de consultas y la gestión de procesos internos.

**NLP (Procesamiento de Lenguaje Natural):** Subcampo de la IA que se enfoca en la interacción entre computadoras y el lenguaje humano. Permite a las máquinas comprender, interpretar y generar texto o lenguaje hablado de manera significativa.

**Chat inteligente:** Software que simula una conversación con los usuarios mediante IA, respondiendo a preguntas o realizando tareas simples. En este proyecto, el chatbot automatizará las consultas de los usuarios.

**Sistema de Información:** Conjunto de herramientas y procedimientos que procesan, almacenan y distribuyen información dentro de una organización, facilitando la toma de decisiones y la gestión de procesos administrativos.

**GISTFA (Grupo de Investigación en Sistemas y Tecnologías de la Información de Facatativá):**

Grupo de investigación enfocado en el desarrollo de soluciones tecnológicas avanzadas, con un enfoque particular en sistemas inteligentes y la automatización de procesos.

**Usuarios:** Personas que interactuaron con el sistema, divididos en tres grupos principales:

- **Estudiantes:** Accederán a información como horarios, manuales de convivencia y solicitarán permisos.
- **Docentes:** Gestionarán permisos y realizarán consultas administrativas.
- **Personal Administrativo:** Administrarán los procesos internos de la institución a través del sistema.

**Permisos:** Autorizaciones que los usuarios pueden solicitar para acceder a ciertas áreas, realizar actividades o disponer de recursos dentro de la institución educativa.

**CRUD (Crear, Leer, Actualizar, Eliminar):** Operaciones básicas que permiten la manipulación de datos en un sistema de gestión. En este proyecto, los roles tendrán acceso a estas operaciones para gestionar la información de los usuarios y el sistema.

**Back-end:** Parte del sistema que gestiona la lógica de negocio, la base de datos y el procesamiento de datos. En este proyecto, el back-end será responsable de gestionar los permisos, accesos y consultas.

**Front-end:** Interfaz gráfica con la que interactúan los usuarios. En este proyecto, el front-end incluye la aplicación móvil y web a través de las cuales los usuarios realizan sus gestiones.

**API (Interfaz de Programación de Aplicaciones):** Conjunto de definiciones y protocolos que permiten que diferentes componentes de software se comuniquen entre sí. Las API permitirán la integración de la plataforma con otros sistemas, si es necesario.

#### 2.2.1.4. Visión General del Documento

Este documento está organizado de manera que cubre todos los aspectos clave necesarios para la correcta especificación del **Sistema Inteligente para la Optimización de Procesos Interinstitucionales en Entornos Educativos**. A continuación se describe la estructura general del documento:

##### **Sección 1: Introducción**

Proporciona una introducción al documento, incluyendo el propósito, el ámbito del sistema, las definiciones clave, y las referencias utilizadas para la elaboración del mismo.

##### **Sección 2: Descripción General del Sistema**

Ofrece una descripción detallada del sistema, proporcionando una perspectiva general del producto, las características principales, los usuarios a los que va dirigido, y las restricciones que deben considerarse durante su desarrollo.

##### **Sección 3: Requisitos Específicos**

Contiene los requisitos técnicos y funcionales del sistema. Se detallan las interfaces del sistema, las funciones que debe realizar, los requisitos de rendimiento y las restricciones de diseño, así como otros atributos del sistema.

##### **Sección 4: Apéndices**

Incluye cualquier información adicional, como diagramas técnicos, detalles de implementación y referencias relevantes que puedan ser útiles para comprender mejor el sistema.

### 2.2.2 DESCRIPCIÓN GENERAL

El **Sistema Inteligente para la Optimización de Procesos Interinstitucionales en Entornos Educativos** es un software diseñado para mejorar la eficiencia en la gestión de procesos internos dentro de las instituciones educativas, automatizando tareas clave como la asignación de permisos, la gestión de roles y la resolución de consultas frecuentes mediante el uso de **inteligencia artificial (IA)** y **procesamiento de lenguaje natural (NLP)**.

El sistema tiene como objetivo principal reducir significativamente los tiempos de respuesta y la carga administrativa mediante la automatización de procesos rutinarios. Proporcionará a los estudiantes, docentes y personal administrativo una plataforma accesible, que les permitirá gestionar sus solicitudes y consultas de manera rápida y eficiente.

El sistema se adapta a las necesidades cambiantes de los usuarios mediante técnicas de aprendizaje automático, permitiendo mejorar continuamente la precisión de las respuestas y la gestión de procesos. Su arquitectura flexible garantiza que sea escalable y replicable en otras instituciones educativas y sectores con necesidades similares de optimización de procesos.

### 2.2.2.1. Perspectiva del Producto

El **Sistema Inteligente para la Optimización de Procesos Interinstitucionales en Entornos Educativos** es un sistema autónomo que interactúa principalmente con los usuarios de las instituciones educativas. El sistema se posiciona como una solución independiente que, sin embargo, puede integrarse con otras herramientas de gestión existentes en la institución, como sistemas de gestión académica.

#### Relación con Otros Sistemas

El sistema está diseñado para ser flexible y autónomo, pero ofrece la posibilidad de conectarse a través de **API** a otros sistemas que las instituciones educativas puedan utilizar, tales como:

- **Sistemas de Gestión Académica:** Integración con plataformas que gestionan los procesos educativos y académicos de la institución, permitiendo el acceso a horarios, registros de estudiantes y normativas.
- **Sistemas de Recursos Humanos:** Para la integración de permisos y solicitudes con los sistemas internos de recursos humanos que manejan el personal administrativo y docente.
- **Sistemas de Bases de Datos:** El sistema puede utilizar las bases de datos institucionales para almacenar información relacionada con los permisos, las solicitudes de los usuarios y las consultas procesadas.

#### Interfaces Principales del Producto

- **Interfaz de Usuario (Front-end):** Disponible para dispositivos móviles (Android) como iniciador, donde los usuarios pueden gestionar permisos, realizar consultas y recibir respuestas automatizadas. La interfaz es intuitiva y accesible para estudiantes, docentes y personal administrativo.
- **API para Integración:** Proporciona un conjunto de servicios fastAPI que permiten integrar el sistema con otras plataformas de gestión educativa y administrativa.
- **Interfaz Administrativa:** Un portal accesible para los administradores del sistema, donde podrán supervisar el funcionamiento general, revisar las solicitudes pendientes.

## Dependencias y Suposiciones

- **Infraestructura de Red:** El sistema requiere que los usuarios tengan acceso a internet para poder acceder a las funcionalidades de la plataforma, tanto desde dispositivos móviles como desde el portal web.
- **Bases de Datos Institucionales:** Se espera que la institución cuente con una base de datos donde se almacene la información de los usuarios (estudiantes, docentes y personal administrativo) para facilitar la gestión de permisos y consultas.

### 2.2.2.2. Funciones del Producto

El **Sistema Inteligente para la Optimización de Procesos institucionales en Entornos Educativos** proporcionará las siguientes funciones clave para usuarios, administradores y el sistema en general:

- **Gestión de Permisos y Roles**
  - **Solicitud de Permisos:** Los usuarios (estudiantes) podrán solicitar permisos una vez registrados en la plataforma, para poder acceder a las funciones del chat y así realizar todo lo relacionado con la gestión académica.
  - **Aprobación/Rechazo de permisos:** Los usuarios (admin) podrán revisar las solicitudes de permisos y aprobarlas o rechazarlas directamente desde la plataforma, con notificación automática a los usuarios.
- **Chatbot Automatizado para Consultas Frecuentes**
  - **Consultas Automáticas:** Los usuarios podrán interactuar con un chatbot inteligente para realizar consultas frecuentes, como horarios de clase, cargar materias, realizar quejas, etc.
  - **Resolución Automática de Consultas:** El chatbot, utilizando técnicas de IA y procesamiento de lenguaje natural (NLP), resolverá automáticamente las consultas basadas en la información pre-cargada en el sistema.
- **Gestión de Usuarios**
  - **Registro de Usuarios:** Los estudiantes, docentes y personal administrativo podrán registrarse en la plataforma, ingresando su información personal y varios parámetros requeridos para el mismo.
  - **Gestión de Perfiles:** Los admin podrán activar o eliminar perfiles de usuarios en el sistema. Esto incluye la asignación de roles y permisos.
  - **Control de Accesos:** Los usuarios solo tendrán acceso a las funciones del sistema que estén asociadas con su rol. Por ejemplo, los estudiantes no podrán cargar materias, mientras que los administrativos sí.

- **Administración del Sistema**
  - **Panel de Control para Administrativos:** Los administrativos tendrán acceso a un panel donde podrán supervisar las diferentes quejas realizadas por los estudiantes o docentes
- **Integración con Otros Sistemas**
  - **Interfaz de Integración con APIs:** El sistema proporcionará APIs para su integración con otros sistemas educativos o administrativos, como sistemas de gestión académica o plataformas de recursos humanos.
  - **Sincronización de Datos:** El sistema podrá sincronizar datos con las bases de datos de la institución .json, como los registros de estudiantes y docentes, para facilitar la gestión de permisos y roles.
- **Reportes y Análisis**
  - **Generación de Reportes:** El sistema permitirá generar reportes sobre el uso del sistema, las solicitudes de permisos, y las interacciones con el chatbot.

### 2.2.2.3. Características de los usuarios

El **Sistema Inteligente para la Optimización de Procesos Interinstitucionales en Entornos Educativos** está diseñado para ser utilizado por diferentes tipos de usuarios dentro de una institución educativa. A continuación, se describen las características de cada uno de ellos:

- **Estudiantes**
  - **Descripción:**  
Los estudiantes son los usuarios finales que interactúan con el sistema para realizar consultas frecuentes (horarios, normativas) y gestionar permisos relacionados con su actividad académica.
  - **Nivel Técnico:**  
Bajo. Se espera que los estudiantes tengan conocimientos básicos sobre el uso de dispositivos móviles o aplicaciones web. La interfaz será intuitiva y fácil de usar para que no requiera conocimientos técnicos avanzados.
  - **Necesidades:**
    - Acceso rápido a información académica, como horarios y normativas.
    - Solicitud de permisos (como permisos de salida o acceso a instalaciones).
    - Respuesta rápida y automatizada a consultas frecuentes.
  - **Responsabilidades:**
    - Utilizar el sistema para realizar consultas.
    - Gestionar sus solicitudes de permisos.

- Interactuar con el chatbot para recibir respuestas a sus consultas.

- **Docentes**

- **Descripción:**

Los docentes tendrán acceso a la plataforma para gestionar permisos y realizar consultas sobre normativas o procedimientos administrativos relacionados con su trabajo.

- **Nivel Técnico:**

Medio. Se espera que los docentes tengan habilidades intermedias en el uso de aplicaciones web y móviles, lo que les permitirá gestionar sus consultas y permisos de manera eficiente.

- **Necesidades:**

- Gestión de permisos de acceso y solicitudes.
- Consultas sobre procedimientos administrativos y normativas internas.
- Posibilidad de recibir notificaciones y actualizaciones sobre los procesos gestionados.

- **Responsabilidades:**

- Gestionar sus solicitudes de permisos.
- Revisar y aprobar o rechazar las solicitudes de sus estudiantes cuando corresponda.
- Interactuar con el sistema para acceder a la información administrativa relevante.

- **Personal Administrativo**

- **Descripción:**

El personal administrativo tiene un rol clave en la administración y supervisión del sistema. Tendrán acceso a funcionalidades avanzadas, como la gestión de solicitudes de permisos y la configuración del chatbot.

- **Nivel Técnico:**

Medio-Alto. Se espera que tengan un conocimiento intermedio de sistemas de gestión y aplicaciones web, ya que deberán administrar los permisos, configurar el sistema y supervisar su funcionamiento.

- **Necesidades:**

- Revisar y procesar las solicitudes de los estudiantes y docentes.
- Generar reportes sobre el uso del sistema, permisos gestionados y tiempos de respuesta.

- **Responsabilidades:**

- Administrar las solicitudes de permisos y los perfiles de usuario.
- Asegurar el correcto funcionamiento del sistema, realizando ajustes y supervisiones periódicas.
- Generar reportes y análisis del rendimiento del sistema.

- **Administradores del Sistema**

- **Descripción:**

Los administradores del sistema son los responsables técnicos que supervisan y gestionan la operatividad general de la plataforma. Tienen acceso completo a todas las funcionalidades, incluyendo la configuración técnica y la integración con otros sistemas.

- **Nivel Técnico:**

Alto. Se espera que los administradores del sistema tengan conocimientos avanzados en el uso de herramientas de gestión de sistemas y bases de datos, así como en la integración con APIs externas.

- **Necesidades:**

- Configuración y mantenimiento de la plataforma, asegurando su correcto funcionamiento.
- Integración con otros sistemas académicos y administrativos.
- Monitoreo del rendimiento del sistema y ajustes en las configuraciones de permisos y chatbot.

- **Responsabilidades:**

- Gestionar la infraestructura técnica del sistema.
- Configurar los parámetros del chatbot y las reglas de automatización de procesos.
- Supervisar la integración del sistema con diferentes plataformas y gestionar el acceso de los usuarios.

### **2.2.2.3.1. Funciones Específicas y Validaciones por Rol**

El rol del estudiante tiene como objetivo principal permitir al usuario acceder a la información académica que le concierne, realizar consultas relacionadas con su desempeño y comunicarse con la administración institucional. A continuación se describen detalladamente las funcionalidades habilitadas para este rol y las validaciones que realiza el sistema para garantizar integridad, seguridad y coherencia en los datos.

### 1. Consultar Horario

Descripción: Permite al estudiante ver su horario semanal con materias, horas y docentes.

Validaciones:

- El `user\_id` debe existir en `horarios\_estudiantes.json`.
- Si no existe horario registrado, se notifica al usuario.
- Se recorren los días y se ordenan las clases cronológicamente por día.

#### Excepciones tratadas:

- Si el archivo no está disponible o presenta errores de lectura, se informa al usuario.

### 2. Consultar Materias Inscritas

Descripción: Muestra todas las materias que el estudiante tiene matriculadas.

Validaciones:

- Se realiza una petición a la ruta: `/estudiante/materias/{user\_id}`.
- El sistema busca el `user\_id` en `materias\_asignadas\_estudiantes.json`.
- Si no hay materias asociadas, se informa al usuario.

#### Excepciones tratadas:

- Problemas de conexión con el backend o archivos faltantes.

### 3. Consultar Docentes

Descripción: Lista los docentes asociados a las materias matriculadas por el estudiante.

Validaciones:

- Se realiza una petición GET al endpoint `/estudiante/docentes/{user\_id}`.
- Se vinculan materias con los docentes desde los datos de inscripción.
- Solo se muestran docentes que correspondan a materias del estudiante.

#### Excepciones tratadas:

- Ausencia de docentes o error al mapearlos con las materias.

### 4. Consultar Notas

Descripción: Permite al estudiante visualizar sus calificaciones actuales por materia y docente.

Validaciones:

- Petición a `/estudiante/notas/{user\_id}`.
- El sistema busca en `notas\_estudiantes.json`.
- Cada nota se vincula con materia y docente.
- Se calcula promedio si hay varios cortes.
- Excepciones tratadas:
- Usuario sin notas registradas.

- Docente desconocido para una materia.

## 5. Enviar Queja a Administración

Descripción: Permite enviar quejas o consultas al equipo administrativo.

Validaciones:

- El sistema verifica que el `rol` sea "estudiante" usando `usuarios\_pendientes.json`.
- El mensaje se guarda en `consultas\_quejas\_estudiantes.jsonl` con timestamp.
- El mensaje no debe estar vacío.

### Excepciones tratadas:

- Archivo inexistente o inaccesible.
- Error de escritura en disco.

## 6. Consultar Horario de un Docente

Descripción: Permite al estudiante consultar los horarios de un profesor ingresando su nombre o materia.

Validaciones:

- Se normaliza la entrada del mensaje ("profe", "programación") para mapear con el nombre de usuario del docente.
- Consulta en `horarios\_docentes.json`.
- Devuelve días, horas y materias del docente buscado.

### Excepciones tratadas:

- Docente no encontrado.
- Docente sin horario registrado.

## 7. Solicitar Permiso

Descripción: Permite al estudiante enviar una solicitud de permiso justificada.

Validaciones:

- El bot solicita el motivo del permiso.
- Se guarda la solicitud en un diccionario temporal con `user\_id`, fecha y motivo.

### Excepciones tratadas:

- Inactividad del usuario durante el flujo.

## 8. Cambiar Contraseña

Descripción: Flujo completo y seguro para cambiar la clave del estudiante.

Validaciones:

- Confirmación de intención: "¿Estás seguro? (Sí/No)".
- Verificación de pregunta de seguridad (con `bcrypt`).
- Envío de código por correo institucional.

- Validación del código.
- Ingreso de nueva contraseña dos veces.
- Hash con `bcrypt` y actualización en `usuarios\_pendientes.json`.

**Excepciones tratadas:**

- Pregunta incorrecta.
- Fallo en envío de correo.
- Código erróneo.
- Las contraseñas no coinciden.
- Usuario no encontrado.

El rol docente tiene como objetivo central la gestión académica de sus estudiantes, incluyendo la asignación de notas, la visualización de alumnos inscritos y la posibilidad de reportar situaciones relevantes a la administración institucional. A continuación se enumeran sus funcionalidades clave:

### 1. Consultar Horario

Descripción: Permite al docente visualizar su propio horario de clases.

Validaciones:

- El `user\_id` debe estar registrado en `horarios\_docentes.json`.
- Se recupera el horario semanal incluyendo día, hora, materia y grado.

**Excepciones tratadas:**

- Horario no registrado.
- Error al leer archivo.

### 2. Ver Estudiantes por Materia

Descripción: Muestra los estudiantes inscritos en cada materia que imparte el docente.

Validaciones:

- Se recuperan las materias del docente desde `horarios\_docentes.json`.
- Se recorren las materias y se agrupan estudiantes sin repetir.
- Se valida que cada estudiante esté correctamente asignado.

**Excepciones tratadas:**

- Materia no asignada.
- Materias sin estudiantes.

### 3. Cargar Notas por Corte

Descripción: Flujo de ingreso de calificaciones divididas por corte.

Validaciones:

- El docente debe seleccionar el periodo (1°, 2°, 3° o final).
- Cada corte debe incluir:
  - Nota por aulas (30%)
  - Evaluaciones (30%)
  - Examen (30%)
  - Asistencia (10%)
- El sistema calcula automáticamente el porcentaje del corte y actualiza la nota final.
- Guarda la nota en `notas\_estudiantes.json`.
- Registra el historial con fecha en `historial\_notas.json`.

**Excepciones tratadas:**

- Estudiante no existe en sus materias.
- Formato de nota inválido.

#### 4. Reportar Estudiante

Descripción: Permite reportar conductas o eventos significativos de un estudiante.

Validaciones:

- El usuario debe ser de rol "docente".
- Se solicita el `user\_id` del estudiante y el motivo del reporte.
- Se guarda en `reportes\_docentes\_a\_estudiantes.json`.

**Excepciones tratadas:**

- Reporte sin motivo.
- Estudiante inexistente.

#### 5. Enviar Queja

Descripción: Permite al docente enviar una queja o comentario a la administración.

Validaciones:

- Se guarda en `quejas\_docentes.json` con timestamp.
- El mensaje debe ser válido (no vacío).

**Excepciones tratadas:**

- Archivo inaccesible.
- Error de escritura.

#### 6. Cambiar Contraseña

Descripción: Flujo seguro de cambio de clave.

Validaciones:

- Confirmación de intención.

- Pregunta de seguridad (`bcrypt`).
- Código al correo institucional.
- Validación de código.
- Nueva contraseña ingresada y confirmada.
- Hash y actualización en `usuarios\_pendientes.json`.

**Excepciones tratadas:**

- Fallo en respuesta de seguridad.
- Código incorrecto.
- Fallo en envío de correo.
- Las contraseñas no coinciden.

El rol administrativo está orientado a la gestión académica operativa de los usuarios del sistema (estudiantes y docentes), así como a la administración de consultas institucionales. Este rol posee permisos para cargar datos críticos, visualizar reportes y coordinar horarios. A continuación se describen en detalle sus funcionalidades, validaciones internas y manejo de excepciones.

### 1. Cargar Materias a Estudiantes

Descripción: Permite registrar las materias que cursará un estudiante.

Validaciones:

- El usuario debe tener el rol "administrativo".
- El estudiante debe existir en `usuarios\_pendientes.json`.
- Se permiten materias repetidas (ya que un estudiante puede ver la misma asignatura en diferentes niveles o grupos).

**Excepciones tratadas:**

- Estudiante no registrado.
- Archivo `materias\_asignadas\_estudiantes.json` corrupto o inexistente.

### 2. Cargar Materias a Docentes

Descripción: Asigna las materias que el docente impartirá.

Validaciones:

- El docente debe existir en `usuarios\_pendientes.json` y tener rol "docente".
- No se pueden repetir materias ya asignadas previamente a ese docente.
- Excepciones tratadas:
  - Docente no registrado o con rol inválido.
  - Materia ya asignada.

### 3. Cargar Horario a Estudiantes

Descripción: Registra el horario académico del estudiante según las materias asignadas.

Validaciones:

- El estudiante debe estar registrado en el sistema.
- El sistema verifica si ya tiene horario y evita colisiones.
- Se toman automáticamente los datos del horario del docente (día, hora, materia, grado) y se asocian al estudiante.
- Se actualiza el campo `estudiantes` dentro de `horarios\_docentes.json` para mantener coherencia.

**Excepciones tratadas:**

- Choques de horario detectados.
- Docente sin horario asociado.
- Materia no válida o estudiante no matriculado.

#### 4. Cargar Horario a Docentes

Descripción: Permite registrar el horario completo semanal de clases de un docente.

Validaciones:

- La materia debe haber sido previamente asignada al docente.
- Se solicita día, hora de inicio, hora de fin, materia y grado.
- No se permite registrar materias no asociadas previamente al docente.

**Excepciones tratadas:**

- La materia no pertenece al docente.
- Error de formato o en la escritura del archivo.

#### 5. Cargar Horario de Atención

Descripción: El administrativo registra su disponibilidad para atender estudiantes.

Validaciones:

- Debe ingresar día, hora\_inicio y hora\_fin.
- Se valida formato correcto de hora y día.
- No se permiten días duplicados para el mismo usuario.
- Se guarda en `horarios\_administrativos.json`.

**Excepciones tratadas:**

- Formato incorrecto.
- Día ya registrado previamente.

#### 6. Ver Quejas o Consultas de Estudiantes

Descripción: Permite visualizar todas las quejas registradas por los estudiantes.

Validaciones:

- Se cargan desde `consultas\_quejas\_estudiantes.jsonl`.

- Se ordenan por timestamp de manera cronológica.

**Excepciones tratadas:**

- Archivo vacío.
- Error de lectura del ``jsonl``.

**7. Eliminar Quejas**

Descripción: Elimina una queja específica del listado.

Validaciones:

- Se solicita número de queja a eliminar ("eliminar queja 2", etc.).
- El número debe estar dentro del rango de quejas existentes.
- El archivo se actualiza con el nuevo listado.

**Excepciones tratadas:**

- Número fuera de rango.
- Archivo inaccesible.

**8. Ver Reportes de Docentes**

Descripción: Muestra todos los reportes generados por docentes sobre estudiantes.

Validaciones:

- Se accede a ``reportes_docentes_a_estudiantes.json``.
- Se muestran campos como: docente, estudiante, motivo, estado.

**Excepciones tratadas:**

- No existen reportes.
- Error de lectura del archivo.

**9. Eliminar Reportes de Docentes**

Descripción: Permite borrar reportes específicos realizados por docentes.

Validaciones:

- Requiere comando como "eliminar reporte 1".
- Verifica si el índice es válido.
- El archivo se sobrescribe con los datos restantes.

**Excepciones tratadas:**

- Índice inválido.
- Falla de escritura.

**10. Cambiar Contraseña**

Descripción: Permite cambiar su propia contraseña mediante un flujo seguro.

Validaciones:

- Confirmación de intención del usuario.

- Validación de respuesta a pregunta secreta (verificada con `bcrypt`).
- Envío de código de verificación al correo.
- Ingreso de nueva contraseña dos veces para confirmar.
- La contraseña se encripta con `bcrypt` y se actualiza en `usuarios\_pendientes.json`.

**Excepciones tratadas:**

- Pregunta incorrecta.
- Código inválido.
- Fallo en el envío del correo (SMTP).
- Contraseñas no coinciden.
- Usuario no encontrado.

El rol administrador tiene como objetivo central gestionar los accesos, seguridad y control de los usuarios registrados en el sistema. Se encarga de la aprobación, eliminación, desbloqueo y modificación de cuentas institucionales. A continuación se presentan sus funciones detalladas, junto con sus respectivas validaciones y excepciones tratadas.

## 1. Eliminar Usuario

Descripción: Permite eliminar completamente a un usuario del sistema.

Validaciones:

- El administrador debe confirmar la intención de eliminación (“¿Estás seguro?”).
- El sistema verifica que el `user\_id` proporcionado exista en `usuarios\_pendientes.json`.
- Se eliminan todos los registros asociados en:
  - `usuarios\_pendientes.json`
  - `horarios\_estudiantes.json` o `horarios\_docentes.json`
  - `materias\_asignadas\_estudiantes.json` o `materias\_asignadas\_docentes.json`
  - `consultas\_quejas\_estudiantes.json`
  - `reportes\_docentes\_a\_estudiantes.json`
  - `notas\_estudiantes.json`
- Se envía un correo automático de despedida al usuario afectado.

**Excepciones tratadas:**

- Usuario inexistente.
- Falla en la escritura de archivos.
- Error en envío de correo.

## 2. Desbloquear Usuario

Descripción: Restaura el acceso a un usuario que ha sido bloqueado.

Validaciones:

- El usuario debe estar listado en `cuentas\_bloqueadas.json`.

- Se cambia el estado a `"bloqueado": false`` y se limpia `fecha_bloqueo``.

**Excepciones tratadas:**

- Usuario no bloqueado.
- Archivo corrupto o inaccesible.

### 3. Activar Usuario y Asignar Rol

Descripción: Activa una cuenta registrada y le asigna un rol válido.

Validaciones:

- El `user_id`` debe existir en `usuarios_pendientes.json``.
- El estado del usuario debe ser `"activo": false``.
- El rol ingresado debe ser uno de los siguientes válidos:
  - `estudiante``
  - `docente``
  - `administrativo``
- El sistema actualiza `"activo": true`` y `"rol": asignado``.

**Excepciones tratadas:**

- Usuario no encontrado.
- Rol inválido.
- Usuario ya activo.

### 4. Cambiar Contraseña

Descripción: Permite al administrador cambiar su propia contraseña siguiendo un flujo seguro.

Validaciones:

- Confirmación del proceso ("¿Estás seguro?").
- Validación de la pregunta secreta (respuesta validada con `bcrypt``).
- Envío de código de verificación al correo institucional.
- Validación del código.
- Ingreso de la nueva contraseña dos veces.
- Encriptación con `bcrypt`` y actualización en `usuarios_pendientes.json``.

**Excepciones tratadas:**

- Pregunta incorrecta.
- Código inválido.
- Las contraseñas no coinciden.
- Fallo en envío de correo.
- Usuario no encontrado.

#### 2.2.2.4. Restricciones

El desarrollo e implementación del **Sistema Inteligente para la Optimización de Procesos Interinstitucionales en Entornos Educativos** está sujeto a las siguientes restricciones, que limitan las decisiones relacionadas con su diseño, infraestructura y funcionalidad.

- **Restricciones Técnicas:**

- **Dependencia de Conexión a Internet:**

El sistema depende de una conexión estable a internet tanto para su uso web como móvil. Esto significa que el acceso a las funcionalidades del sistema puede verse afectado en áreas con baja o inestable conectividad. No se contempla una versión offline del sistema en esta primera etapa.

- **Compatibilidad de Dispositivos:**

El sistema debe ser compatible con las versiones actuales de navegadores web (Chrome, Firefox, Safari) y sistemas operativos móviles (iOS y Android). Esto puede limitar el soporte a versiones más antiguas de estos navegadores o sistemas operativos, lo que podría excluir a ciertos usuarios con dispositivos desactualizados.

- **Limitaciones en el Procesamiento del Lenguaje Natural (NLP):**

Aunque el sistema utiliza técnicas de procesamiento de lenguaje natural, este se limitará en un principio a consultas comunes y estructuradas. Consultas más complejas o ambiguas pueden no ser respondidas correctamente, y es posible que se necesite intervención humana para resolverlas.

- **Restricciones de Infraestructura:**

- **Capacidad del Servidor:**

La institución educativa debe contar con servidores o infraestructura en la nube con suficiente capacidad para manejar el tráfico de usuarios simultáneos, especialmente durante los periodos de alta demanda, como el inicio de semestre. Un mal dimensionamiento de los servidores podría afectar la respuesta del sistema.

- **Seguridad de los Datos:**

El sistema debe cumplir con normativas de protección de datos y seguridad, tales como el Reglamento General de Protección de Datos (GDPR) en caso de que la institución lo requiera. Esto impone restricciones en la forma en que se recopilan, almacenan y transmiten los datos de los usuarios, y puede implicar el uso de cifrado y autenticación de dos factores.

- **Restricciones de Diseño:**

- Interfaz de Usuario Simplificada:  
La interfaz de usuario debe ser lo suficientemente simple e intuitiva para que usuarios con bajo conocimiento tecnológico puedan interactuar con el sistema sin dificultad. Esto puede limitar el uso de características avanzadas en la interfaz para evitar sobrecargar al usuario.
- Acceso Basado en Roles:  
Las funcionalidades del sistema deben estar limitadas según los roles de los usuarios (estudiantes, docentes, personal administrativo). Los usuarios solo podrán acceder a las funciones correspondientes a su nivel de permisos, lo que impone restricciones sobre el diseño y configuración de las funcionalidades.
- **Restricciones Operativas:**
  - Mantenimiento y Actualizaciones:  
Se debe establecer un programa de mantenimiento regular para el sistema, que incluya actualizaciones de software, correcciones de errores y mejoras de seguridad. El sistema podría estar inactivo durante estos periodos de mantenimiento, lo que afectaría la disponibilidad para los usuarios.
  - Capacitación de Usuarios:  
Se requiere que el personal administrativo reciba una capacitación inicial sobre el uso del sistema. Esto implica que los plazos de implementación pueden depender de la disponibilidad de este personal para recibir dicha capacitación.
- **Restricciones Legales y Regulatorias:**
  - Cumplimiento con Normativas de Privacidad de Datos:  
El sistema debe cumplir con las normativas locales e internacionales de privacidad de datos. Cualquier cambio en estas normativas podría requerir ajustes en la arquitectura de seguridad del sistema.
  - Licencias de Software:  
El sistema debe cumplir con las licencias de los componentes de software utilizados (frameworks, bibliotecas de NLP, etc.). Cualquier cambio en los términos de las licencias podría limitar el uso de ciertos componentes o imponer costos adicionales.

#### 2.2.2.5. Suposiciones y Dependencias

El desarrollo y funcionamiento del **Sistema Inteligente para la Optimización de Procesos Interinstitucionales en Entornos Educativos** dependen de varios factores externos e internos. Estas suposiciones y dependencias son cruciales para garantizar el correcto desempeño del sistema.

- **Suposiciones**

- **Acceso a Internet:**

Se asume que los usuarios (estudiantes, docentes y personal administrativo) tienen acceso estable a internet, ya que el sistema requiere conexión para interactuar con la plataforma en tiempo real. Tanto la versión web como la versión móvil del sistema dependen de una conectividad adecuada.
- **Disponibilidad de Dispositivos Electrónicos:**

Se supone que los usuarios contarán con dispositivos adecuados para acceder al sistema, tales como smartphones, tablets o computadoras con navegadores compatibles. El sistema está diseñado para ser accesible desde múltiples dispositivos.
- **Disponibilidad de Datos Institucionales:**

Se asume que la institución educativa proporcionará bases de datos actualizadas con la información necesaria para la gestión de usuarios, como registros de estudiantes, docentes y personal administrativo. Esto es fundamental para la correcta configuración y operación del sistema.
- **Capacitación de los Usuarios:**

Se espera que los usuarios reciban una capacitación mínima sobre el uso del sistema, especialmente el personal administrativo, que será responsable de gestionar el sistema y resolver consultas complejas.
- **Escalabilidad del Sistema:**

Se asume que el sistema está diseñado para ser escalable, por lo que será capaz de adaptarse al crecimiento de la institución en términos de número de usuarios y volumen de consultas o solicitudes.
- **Dependencias**
  - **Infraestructura Tecnológica:**

El sistema depende de una infraestructura tecnológica adecuada en la institución, lo que incluye servidores para alojar la plataforma, bases de datos para almacenar la información de los usuarios, y una red de internet confiable que permita el acceso remoto al sistema. El rendimiento del sistema puede verse afectado si la infraestructura no cumple con los requisitos.
  - **Integración con Otros Sistemas:**

El sistema puede requerir integrarse con otros sistemas de la institución, como sistemas de gestión académica o de recursos humanos. La disponibilidad de las API o interfaces de estos sistemas es una dependencia crítica para asegurar una interoperabilidad fluida.
  - **Actualizaciones de Software y Hardware:**

El sistema dependerá de actualizaciones regulares tanto del software (plataforma web y móvil) como del hardware (servidores, dispositivos de

usuario) para mantener su funcionamiento óptimo y seguridad. Las actualizaciones deben ser gestionadas correctamente por los administradores del sistema.

- **Seguridad y Protección de Datos:**

El sistema depende de la implementación de políticas de seguridad de datos adecuadas para proteger la información personal y sensible de los usuarios. Se espera que la institución implemente estándares de seguridad como cifrado de datos, autenticación de usuarios y gestión segura de contraseñas.

- **Disponibilidad de Personal Administrativo:**

Se requiere la participación activa del personal administrativo para supervisar las solicitudes, resolver consultas que no puedan ser gestionadas automáticamente, y mantener el sistema actualizado. La falta de personal puede afectar la eficiencia del sistema.

#### 2.2.2.6. Requisitos Futuros

El **Sistema Inteligente para la Optimización de Procesos Interinstitucionales en Entornos Educativos** está diseñado para ser flexible y escalable. A medida que las necesidades de la institución evolucionen, se prevén posibles expansiones y mejoras en el sistema. A continuación, se describen algunos de los requisitos futuros que podrían ser implementados:

- **Expansión de la Funcionalidad del Chatbot**

- **Mejora de la Capacidad de Procesamiento del Lenguaje Natural (NLP):**

En futuras versiones, el chatbot podría ser mejorado para comprender y procesar preguntas más complejas, detectando no solo palabras clave, sino también el contexto de las consultas, brindando respuestas más personalizadas y precisas.

- **Integración con Nuevos Sistemas**

- **Integración con Plataformas de Aprendizaje en Línea (LMS):**

En el futuro, el sistema podría integrarse con plataformas de aprendizaje en línea (como Moodle, Blackboard, u otras) para ofrecer funcionalidades adicionales relacionadas con la gestión académica, como el seguimiento de asignaciones, resultados académicos y asistencia a clases.

- **Integración con Herramientas de Comunicación Institucional:**

El sistema podría conectarse con herramientas de mensajería y correo electrónico institucional para permitir el envío automatizado de notificaciones importantes sobre cambios en horarios, normativas, o actualizaciones administrativas.

- **Reportes Avanzados y Analítica Predictiva**

- **Analítica Avanzada y Predicciones:**

En futuras versiones, se podrían implementar algoritmos avanzados de aprendizaje

automático para realizar análisis predictivos. Esto incluiría la identificación de patrones en las solicitudes de permisos o consultas, anticipándose a necesidades de los usuarios y proporcionando soluciones proactivas.

- **Dashboards Personalizados:**

Los administradores podrían acceder a paneles personalizados que les permitan visualizar datos en tiempo real sobre el uso del sistema, el comportamiento de los usuarios, y el rendimiento del chatbot. Estas métricas podrían incluir tiempos de respuesta, número de consultas resueltas automáticamente y satisfacción de los usuarios.

- **Mejoras en la Gestión de Permisos**

- **Permisos Automáticos Basados en Inteligencia Artificial:**

En versiones futuras, se podría implementar un sistema de aprobación automática de permisos basados en patrones de comportamiento y criterios predefinidos por la institución. Esto reduciría aún más la intervención humana en la gestión de permisos.

- **Expansión a Otros Sectores**

- **Escalabilidad a Otras Industrias:**

Aunque el sistema está diseñado para instituciones educativas, en el futuro podría ser adaptado para su uso en otros sectores, como empresas o entidades gubernamentales, donde se requiera la automatización de procesos internos similares (gestión de permisos, roles y consultas).

- **Soporte para Instituciones Multisede:**

Para instituciones con múltiples campus o sedes, se podría implementar una funcionalidad que permita gestionar las solicitudes de permisos y consultas desde diferentes ubicaciones, centralizando la información pero adaptándose a las particularidades de cada sede.

### 2.2.3 REQUISITOS ESPECÍFICOS

Esta sección contiene los requisitos técnicos y funcionales detallados que el **Sistema Inteligente para la Optimización de Procesos Interinstitucionales en Entornos Educativos** debe cumplir. Estos requisitos se agrupan en diferentes categorías para facilitar su comprensión y análisis.

- **Interfaces Externas**

- **Interfaces de Usuario:**

- **Móvil:** El sistema deberá ser accesible tanto desde navegadores web FastApi como desde aplicaciones móviles flutter. Las interfaces deben ser intuitivas, fáciles de navegar y adaptadas a las necesidades de los

diferentes tipos de usuarios (estudiantes, docentes, personal administrativo).

- **APIs para Integración con Otros Sistemas:**
  - El sistema debe ofrecer una API que permita la integración con otros sistemas institucionales, como los sistemas de gestión académica (SGA), recursos humanos, y bases de datos de la institución educativa.
- **Sistema de Notificaciones:**
  - El sistema debe ser capaz de enviar notificaciones automáticas por correo electrónico o SMS cuando se aprueben o rechacen solicitudes de permisos o cuando existan actualizaciones importantes.

- **Funciones**

Las siguientes funciones describen las principales acciones que el sistema debe realizar:

- **Gestión de Permisos:**
  - Los usuarios deben poder solicitar permisos a través de la plataforma, ya sea desde la aplicación móvil o la versión web.
  - Los administradores deben poder aprobar o rechazar permisos, y los usuarios recibirán una notificación con el resultado de su solicitud.
  - El sistema debe mantener un registro histórico de todas las solicitudes de permisos.
- **Chatbot para Resolución de Consultas:**
  - El sistema debe incluir un chatbot basado en inteligencia artificial que pueda resolver consultas frecuentes de los usuarios sobre normativas, horarios y manuales de convivencia.
  - El chatbot debe ser capaz de redirigir consultas más complejas al personal administrativo, cuando no pueda resolverlas automáticamente.
- **Gestión de Usuarios:**
  - El sistema debe permitir la creación, edición, eliminación y consulta de perfiles de usuarios. Los administradores podrán gestionar los roles y permisos asignados a cada usuario.
  - Los usuarios deben poder registrarse y autenticarse en la plataforma mediante credenciales proporcionadas por la institución.
- **Generación de Reportes:**
  - Los administradores deben poder generar reportes personalizados sobre el uso del sistema, incluyendo estadísticas sobre solicitudes de permisos, interacciones con el chatbot, y tiempos de respuesta.

- **Requisitos de Rendimiento**

El sistema debe cumplir con los siguientes requisitos de rendimiento:

- **Tiempos de Respuesta:**
  - El sistema debe procesar y responder a las solicitudes de permisos en un tiempo máximo de 5 segundos.
  - Las consultas realizadas al chatbot deben ser resueltas en menos de 2 segundos para las consultas simples y en menos de 10 segundos para las consultas redirigidas.
- **Usuarios Simultáneos:**
  - El sistema debe ser capaz de soportar un mínimo de 1000 usuarios simultáneamente sin que el rendimiento se vea afectado.
- **Capacidad de Almacenamiento:**
  - El sistema debe ser capaz de almacenar y gestionar al menos 100,000 solicitudes de permisos y 500,000 interacciones con el chatbot sin degradar el rendimiento.

- **Restricciones de Diseño**

El diseño del sistema está sujeto a las siguientes restricciones:

- **Interfaz de Usuario Simple:**
  - La interfaz debe estar diseñada de manera que sea fácil de usar para personas con conocimientos técnicos básicos. Debe cumplir con las pautas de accesibilidad, asegurando que los estudiantes, docentes y personal administrativo puedan utilizarla sin problemas.
- **Seguridad de Datos:**
  - El sistema debe implementar medidas de seguridad como el cifrado de datos en tránsito y en reposo. Todos los datos personales y sensibles deben estar protegidos para cumplir con normativas de protección de datos (por ejemplo, GDPR).
- **Acceso Basado en Roles:**
  - Los usuarios solo deben poder acceder a las funcionalidades que correspondan a su rol (estudiante, docente, administrador). El acceso a los datos y funciones sensibles debe estar restringido a los administradores del sistema.

- **Atributos del Sistema**

Los atributos no funcionales del sistema incluyen:

- **Escalabilidad:**
  - El sistema debe ser escalable, permitiendo agregar nuevos módulos o aumentar el número de usuarios sin afectar el rendimiento.
- **Mantenibilidad:**
  - El código debe estar bien documentado y estructurado para facilitar futuras modificaciones y actualizaciones. Se debe garantizar que el sistema sea fácil de mantener.
- **Disponibilidad:**
  - El sistema debe estar disponible al menos el 99% del tiempo, con periodos de inactividad programados para mantenimiento que no excedan 1 hora por mes.
- **Portabilidad:**
  - El sistema debe ser compatible con los principales navegadores web (Chrome, Firefox, Safari) y plataformas móviles (iOS y Android), asegurando una experiencia de usuario consistente en todos los dispositivos.
- **Fiabilidad:**
  - El sistema debe manejar correctamente cualquier fallo que pueda ocurrir (como la pérdida de conectividad) y garantizar que los datos no se pierdan o se corrompan.
- **Otros Requisitos**
  - **Soporte Multilenguaje:**
    - El sistema debe tener la capacidad de soportar múltiples idiomas, permitiendo a las instituciones ofrecer acceso en diferentes lenguas según su población estudiantil.
  - **Actualizaciones Automáticas:**
    - El sistema debe permitir actualizaciones automáticas para garantizar que las nuevas funcionalidades y parches de seguridad se apliquen sin necesidad de intervención manual por parte del personal administrativo.

## 2.3. ESPECIFICACIÓN DEL DISEÑO

### 2.3.1. MODELO DE ENTIDAD RELACIÓN (MER) -TABLAS PARA LA LÓGICA DE NEGOCIO DEL PROYECTO

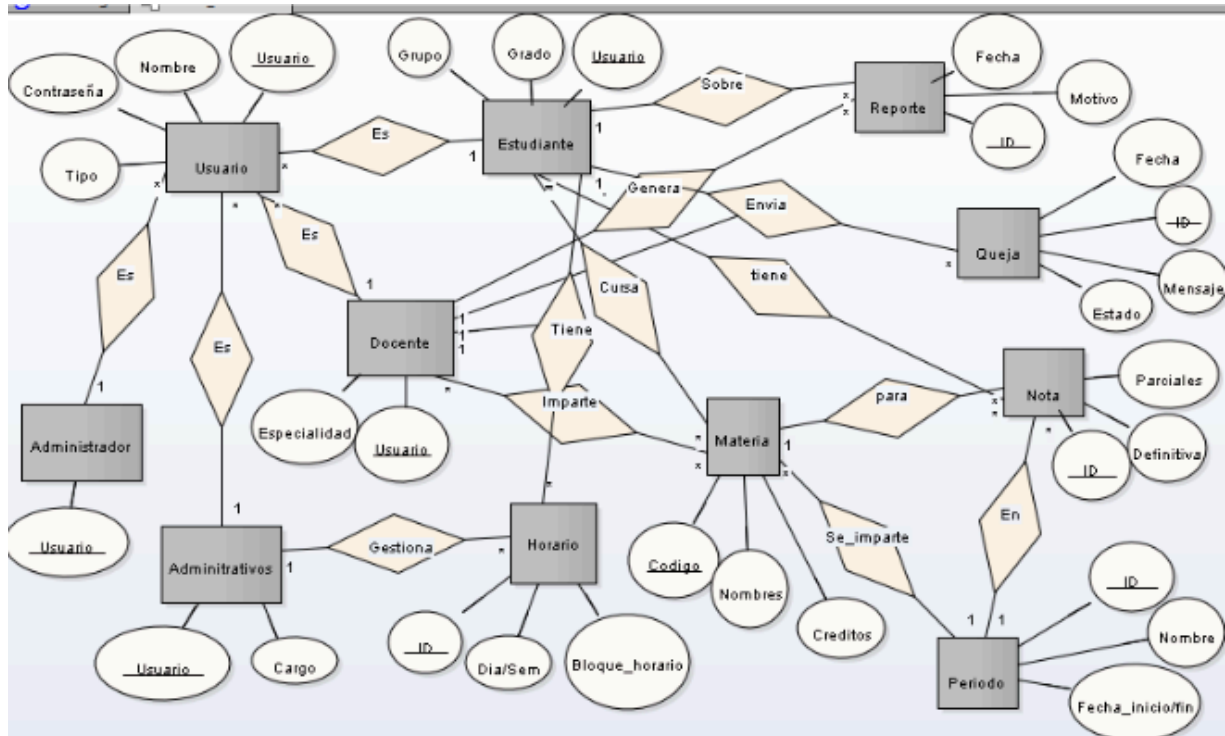


Diagrama 1 MER. fuente propia

#### 2.3.1.2. Descripción de las tablas de la base de datos del proyecto

A continuación se describen los archivos JSON que actúan como tablas dentro del sistema de chatbot institucional, indicando su propósito y contenido.

- Tabla “usuarios\_pendientes”
  - Registra los usuarios que han solicitado acceso al sistema y están pendientes de aprobación por parte del administrador.
- Tabla “solicitudes\_permisos”
  - Almacena las solicitudes de permisos realizadas por los usuarios, incluyendo estado, motivo y respuesta.
- Tabla “codigos\_verificacion”
  - Contiene los códigos de verificación por correo usados durante el registro o recuperación de cuenta.
- Tabla “codigos temporales”
  - Registra códigos de validación de un solo uso para procesos sensibles como autenticaciones secundarias.

- Tabla “codigos\_cambio\_contrasena”
  - Guarda los códigos emitidos para recuperación de contraseña, asociados al usuario.
- Tabla “cuentas\_bloqueadas”
  - Lista de usuarios bloqueados temporalmente por intentos fallidos o razones de seguridad.
- Tabla “admin\_actualizado”
  - Datos activos del usuario administrador del sistema, incluyendo acceso y configuración.
- Tabla “docente\_actualizado”
  - Contiene los datos de docentes registrados y sus credenciales de acceso.
- Tabla “estudiante\_actualizado”
  - Información de estudiantes activos, con vínculo a materias, horarios o notas.
- Tabla “administrativo\_actualizado”
  - Usuarios administrativos con acceso a seguimiento de procesos y gestión interna.
- Tabla “horarios\_estudiantes”
  - Horarios asignados a estudiantes por día, materia y periodo.
- Tabla “horarios\_docentes”
  - Distribución de horarios para docentes en relación a materias.
- Tabla “horarios\_administrativos”
  - Define horarios de atención o trabajo del personal administrativo.
- Tabla “materias\_disponibles”
  - Lista completa de materias disponibles en la institución.
- Tabla “materias\_asignadas\_estudiantes”
  - Relación entre estudiantes y las materias que cursan actualmente.
- Tabla “materias\_asignadas\_docentes”
  - Relación entre docentes y las materias que tienen asignadas.
- Tabla “notas\_estudiantes”
  - Registro de notas de estudiantes, organizadas por materia y docente.
- Tabla “historial\_notas\_estudiantes”
  - Historial académico de calificaciones por estudiante a lo largo del tiempo.
- Tabla “reportes\_docentes\_a\_estudiantes”
  - Reportes que los docentes hacen sobre estudiantes a efectos disciplinarios o de seguimiento.
- Tabla “quejas\_docentes”
  - Registro de quejas registradas por docentes respecto a estudiantes o al sistema.
- Tabla “consultas\_quejas\_estudiantes\_agrupado”
  - Agrupación estadística o por categoría de las quejas estudiantiles para análisis institucional.

- Tabla “chat\_logs”
  - Historial de conversaciones entre el usuario y el chatbot, usado para auditoría y mejora de respuestas.
- Tabla “intents\_admin”
  - Intents (intenciones de usuario) asociadas al rol administrador, para el entrenamiento del chatbot.
- Tabla “intents\_docente”
  - Intents relacionados al rol docente.
- Tabla “intents\_administrativo”
  - Intents configurados para usuarios administrativos.

### **2.3.2. DIAGRAMAS DE CASOS DE USO**

Es la forma en la que se describe lo que hace un sistema, enfocándose entre las interacciones de los usuarios o actores y del sistema para realizar una acción en concreto.

**Participantes:**

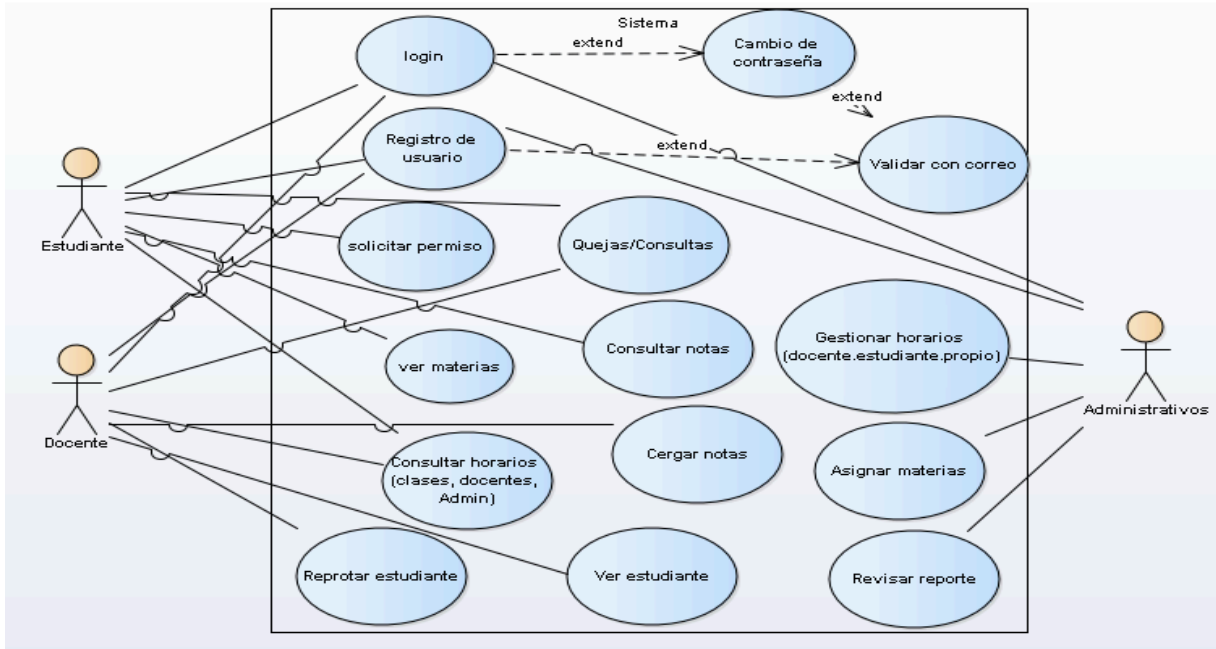


Diagrama 2. Casos de uso participantes. fuente propia  
**Administrador:**

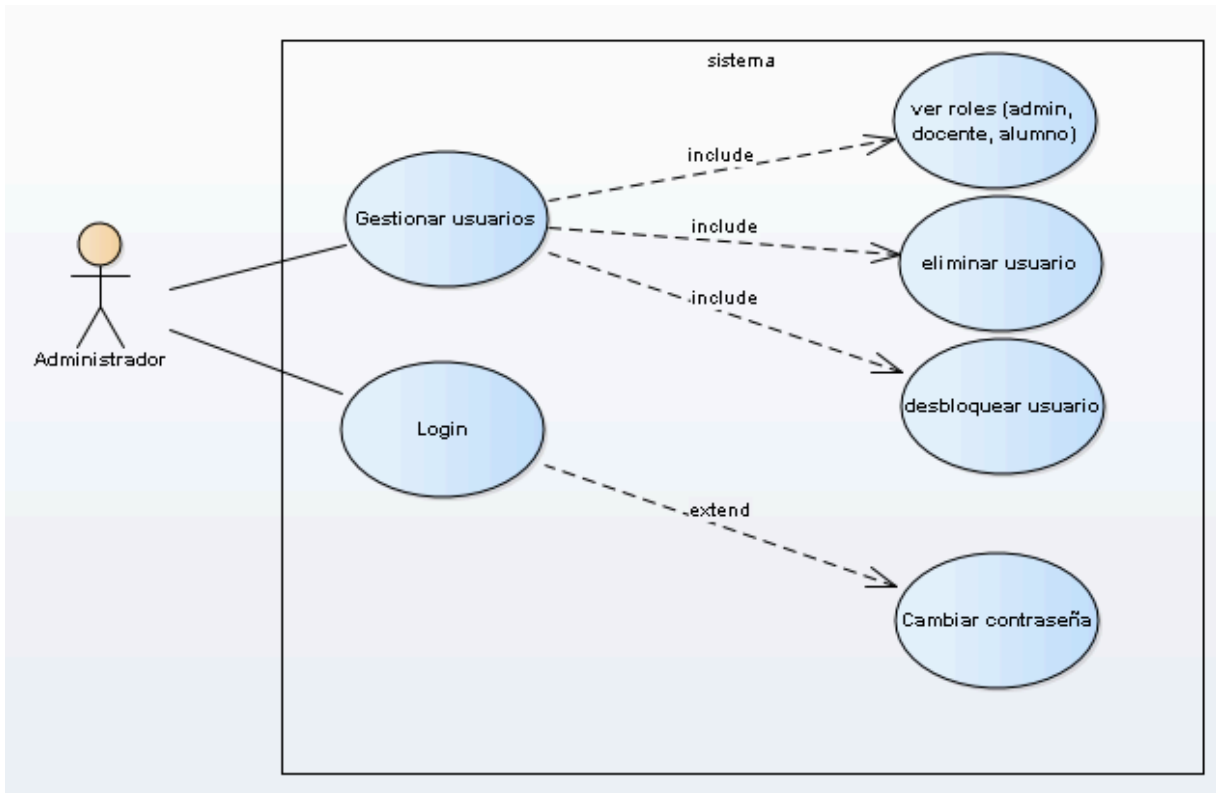


Diagrama 3. Casos de uso administrador. fuente propia

### 2.3.3. DIAGRAMAS DE SECUENCIAS

Esta herramienta nos muestra cómo se comunican los objetivos entre sí en un determinado orden visualizando cómo fluyen los procesos dentro de un sistema.

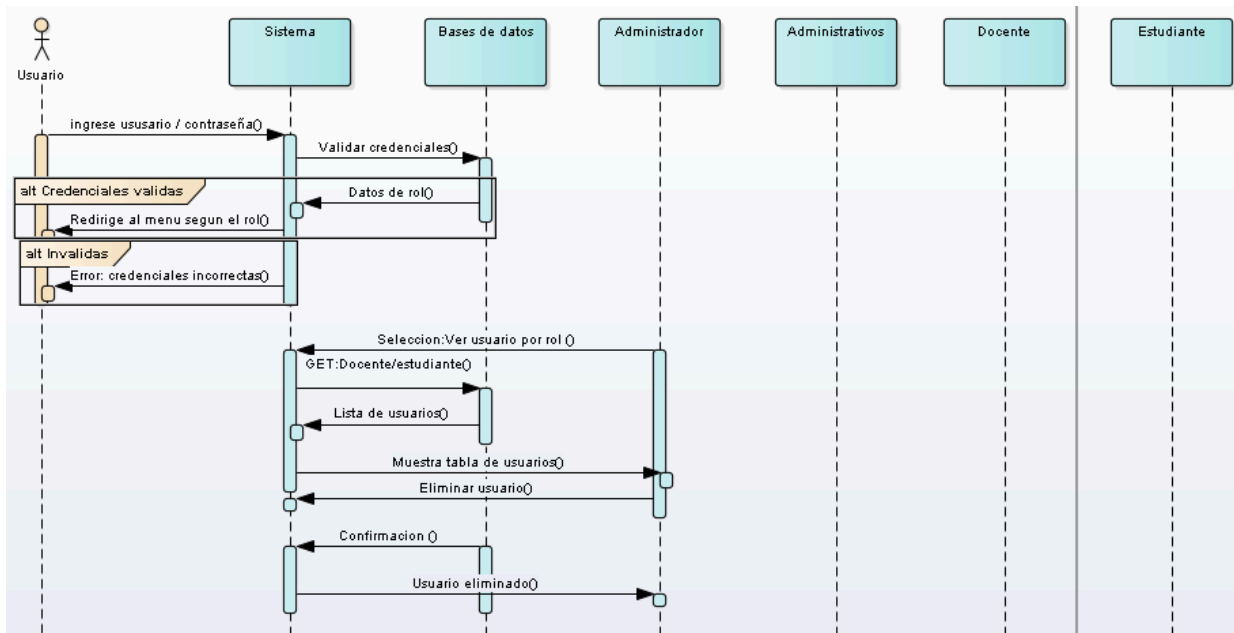


Diagrama 4 secuencia 1. fuente propia

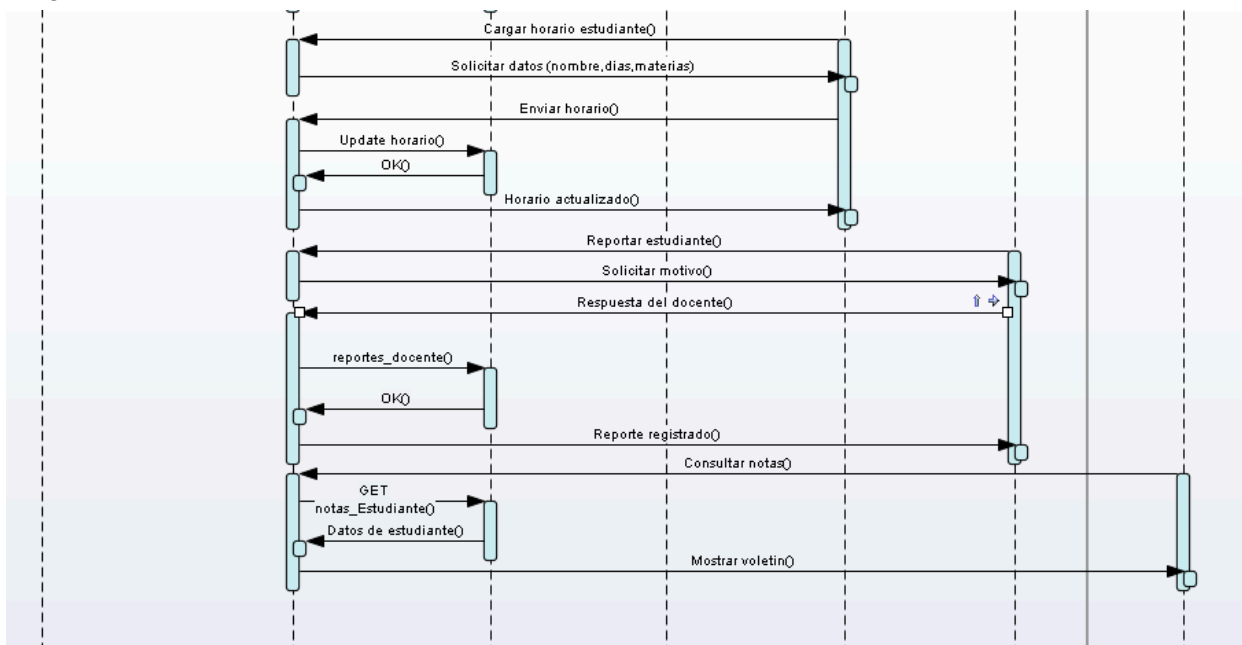


Diagrama 5 secuencia 2. fuente propia

### 2.3.4. DIAGRAMAS DE ACTIVIDADES

Presenta el flujo de trabajo o los pasos a seguir para realizar un proceso o tarea, es similar a un diagrama de flujo tradicional, pero usado en el contexto de modelado de software (UML), que describe la actividad de una acción a otra, incluyendo decisiones, ciclos, y actividades paralelas.

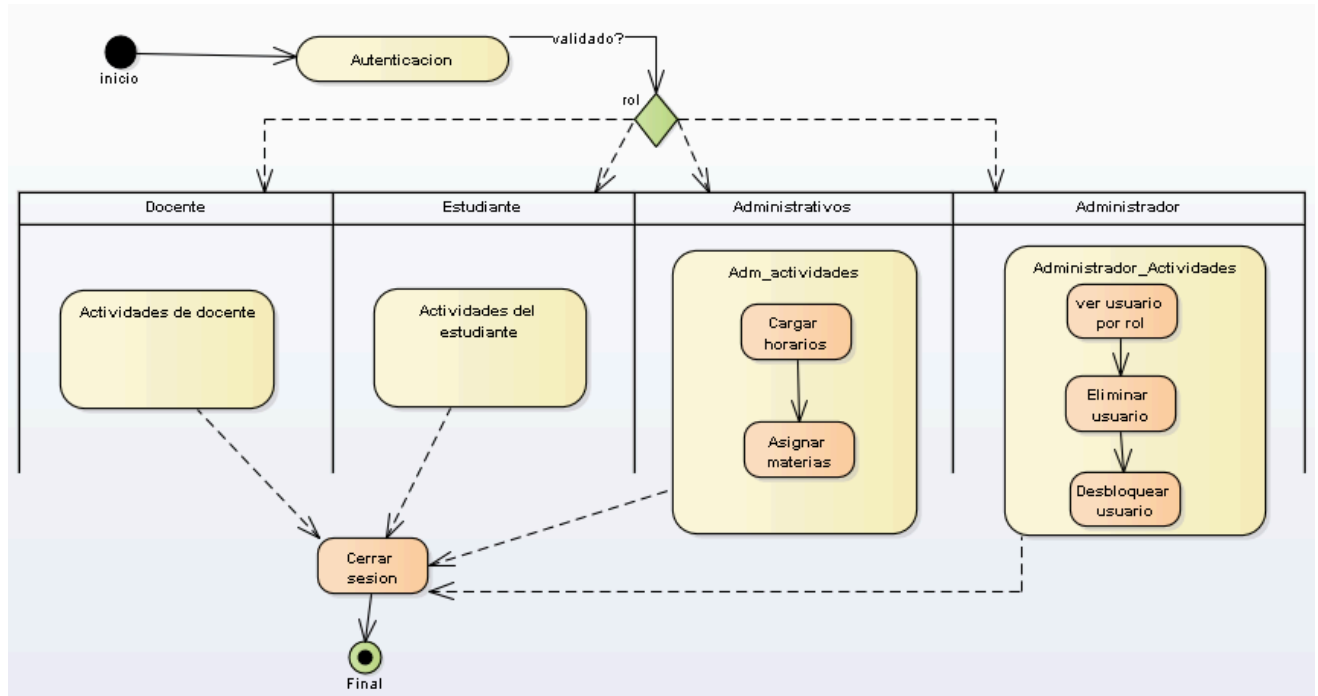


Diagrama 6 actividades. fuente propia

### 2.3.5. DIAGRAMA DE CLASES

Este muestra los componentes que se encuentran dentro del sistema, en donde se incluyen las clases, acciones y propiedades, así como la relación que hay entre los distintos elementos presentes.

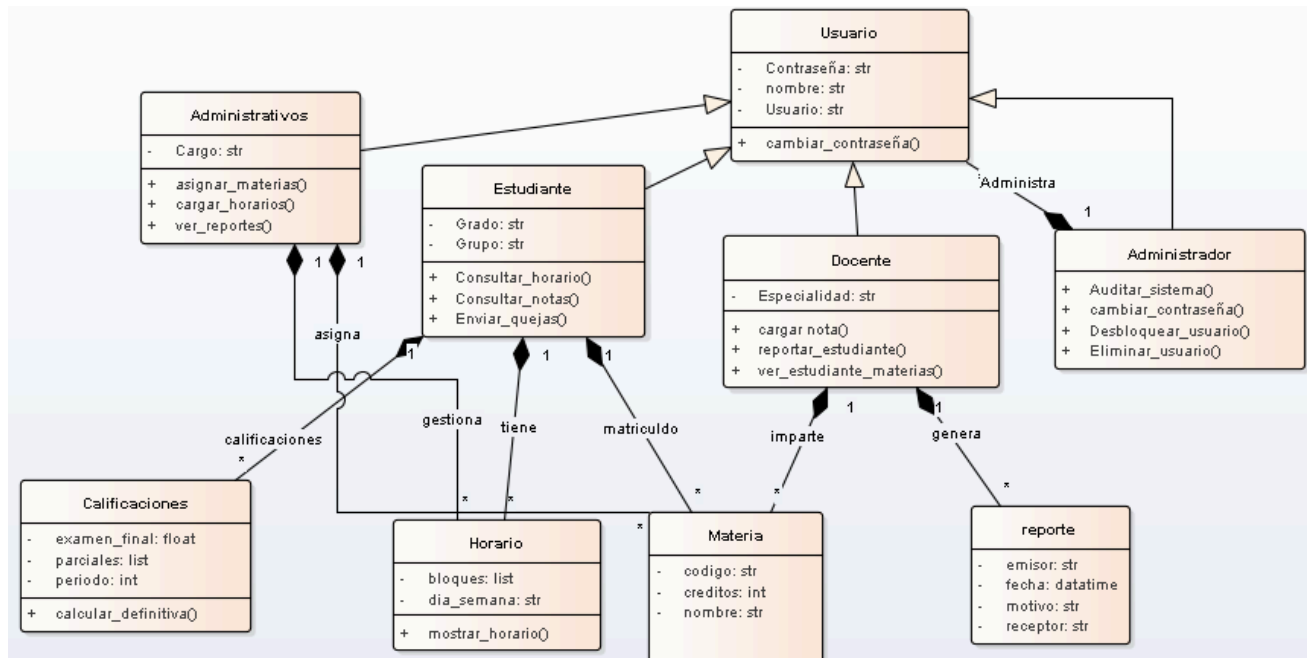


Diagrama 7 Clases. fuente propia

### 2.3.6. DESCRIPCIÓN DE LAS CLASES DEL PROYECTO

- Usuario
  - Contiene los atributos generales de los usuarios como nombre, contraseña, tipo de usuario, así como métodos de autenticación y cambio de contraseña. Es la clase base para roles específicos.
- Estudiante
  - Extiende a EUsuario. Incluye atributos como grado y grupo. Define métodos como consultar horario, consultar notas y enviar quejas. Se relaciona con materias, notas, reportes y quejas.
- Docente
  - Extiende a EUsuario. Incluye la especialidad del docente y métodos como cargar notas, generar reportes, ver estudiantes y asignar materias. Se relaciona con materias y reportes.
- Administrador
  - Extiende a EUsuario. Posee métodos administrativos como auditar el sistema, cambiar contraseñas de otros usuarios, desbloquear y eliminar cuentas.
- Administrativo
  - Extiende a EUsuario. Permite asignar materias, cargar horarios y ver reportes. Su rol facilita la gestión académica desde el área administrativa.

- **Materia**
  - Contiene atributos como código, nombre y créditos. Se relaciona con docentes, estudiantes y horarios. Incluye lógica para matrícula y validación de asignación.
- **Horario**
  - Define los bloques de horario y los días de la semana asociados a materias, docentes y estudiantes. Incluye métodos para mostrar horarios y validar disponibilidad.
- **Calificaciones**
  - Registra los exámenes, notas parciales, periodo y cálculo de la nota definitiva. Se relaciona directamente con estudiantes y materias.
- **Reporte**
  - Clase que almacena los reportes emitidos por los docentes, incluyendo motivó, fecha, emisor y receptor del reporte.

## 2.4. DISEÑO DE LOS CASOS DE PRUEBA

### 2.4.1 Pruebas de Calidad del Código

Para garantizar la calidad y mantenibilidad del sistema desarrollado, se realizaron revisiones manuales de código sobre los componentes clave tanto del backend (FastAPI) como del frontend (Flutter). Aunque no se utilizaron herramientas automatizadas como SonarQube, se aplicaron principios básicos de buenas prácticas en desarrollo de software.

En el backend, se verificó que cada ruta o endpoint estuviera separada por responsabilidad (por ejemplo, rutas específicas para login, registro, verificación y recuperación), y se implementaron respuestas estandarizadas con manejo de errores mediante try/except, códigos de estado HTTP y uso de JsonResponse para consistencia en la salida.

También se validó el cumplimiento de buenas prácticas de seguridad:

- Uso de cifrado de contraseñas y respuestas de seguridad con bcrypt.
- Autenticación basada en tokens JWT para asegurar las sesiones.
- Validaciones de entrada en formularios (ej. correos institucionales, campos vacíos, duplicados).
- Protección de recursos sensibles como usuarios\_pendientes y admin\_panel para que solo los roles autorizados puedan acceder.

En el frontend, se mantuvo una arquitectura modular donde cada pantalla (login\_screen, register\_screen, chat\_screen, etc.) cumple una función específica. Los servicios (auth\_service.dart, chat\_service.dart) están separados del UI, permitiendo mayor claridad, reutilización y mantenibilidad.

Aunque no se utilizó un linter automatizado, se realizó una revisión manual del código Dart para asegurar que las funciones estén bien nombradas, los widgets estén correctamente estructurados y se reutilicen componentes como validadores, colores y estilos definidos en carpetas dedicadas (utils/, theme/).

La validación de calidad fue también apoyada por pruebas manuales funcionales para cada uno de los flujos principales: registro, autenticación, recuperación, interacción con el chatbot y funciones de panel administrativo. Estas pruebas ayudaron a identificar errores lógicos o de interfaz que fueron corregidos durante el desarrollo.

El resultado de estas revisiones fue un sistema funcional, seguro y modular, que puede escalarse o integrarse fácilmente con nuevas funcionalidades futuras.

#### 2.4.2. DEFINICIÓN DE LOS CASOS DE PRUEBA FUNCIONALES DE LA APLICACIÓN

A continuación se presentan los casos de prueba funcionales aplicados a los flujos principales del sistema desarrollado, evaluando entradas, procesos, salidas y validación de resultados esperados.

Tabla 5. casos de prueba.fuente propia.

Caso de Prueba	Entrada	Resultado Esperado	Resultado Real
Registro de usuario nuevo	Formulario completo con correo institucional, pregunta y respuesta de seguridad	El usuario es almacenado en usuarios_pendientes.json y recibe mensaje de verificación	Correcto, se valida correo y se guarda en el backend
Inicio de sesión exitoso	Usuario y contraseña correctos de cuenta activa	Se genera token JWT y se permite acceso al sistema	Correcto, acceso permitido y token recibido
Inicio de sesión fallido	Contraseña incorrecta	Se deniega acceso y se registra intento fallido	Correcto, se muestra mensaje de error y se evita acceso
Recuperación de contraseña	Usuario válido, respuesta correcta, nuevo código de seguridad	Permite registrar una nueva contraseña encriptada	Correcto, nueva contraseña guardada

Acceso al chatbot	Usuario con rol validado	Se muestra interfaz de chat personalizada según el rol	Correcto, respuestas del bot según el tipo de usuario
Consulta de horario	Estudiante autenticado escribe 'ver horario'	El bot responde con el horario correspondiente	Correcto, devuelve JSON con días y materias
Asignación de rol por administrador	Administrador selecciona usuario y rol	El usuario cambia de estado pendiente a activo con nuevo rol	Correcto, usuario actualizado en JSON
Envío de mensaje sin intent válido	Usuario escribe mensaje no reconocido	El sistema indica que no entiende y sugiere opciones	Correcto, bot responde con mensaje de ayuda

### 2.4.3. EVALUACIÓN TÉCNICA DEL CÓDIGO

A continuación se presenta una evaluación técnica basada en criterios de calidad y estructura del código implementado, con base en las buenas prácticas observadas durante el desarrollo del proyecto.

- **Fiabilidad del código:**  
Se realizaron pruebas funcionales manuales sobre cada uno de los módulos del sistema, incluyendo registro, inicio de sesión, recuperación de contraseña, interacción con el chatbot y gestión por parte del administrador. Los resultados evidencian un comportamiento consistente del sistema, sin errores inesperados ni interrupciones durante el uso normal.
- **Seguridad:**  
El sistema cuenta con múltiples mecanismos de seguridad: cifrado de contraseñas y respuestas de seguridad usando bcrypt, autenticación mediante tokens JWT, validación de identidad por correo institucional y preguntas de seguridad, y control de acceso basado en roles. Estas prácticas aseguran la protección de los datos y el acceso adecuado según perfil.
- **Mantenibilidad del código:**  
Tanto el backend como el frontend fueron organizados por responsabilidades, permitiendo que cada componente (servicios, pantallas, rutas) sea independiente y fácil de modificar. Se utilizaron carpetas separadas para los archivos de usuarios, intents, horarios, materias y configuración, facilitando la lectura y futuras extensiones del sistema.

- Código duplicado:  
Aunque no se aplicaron herramientas automáticas de análisis, durante la construcción del sistema se cuidó evitar la duplicación de lógica, reutilizando funciones y estructuras comunes como validadores, controladores y servicios en ambos entornos.
- Tamaño del proyecto:  
El sistema completo está compuesto por más de 25 archivos funcionales, entre rutas FastAPI, pantallas de Flutter, servicios, archivos JSON y modelos de intent. Esto refleja una arquitectura modular y escalable, adecuada para entornos educativos en crecimiento.
- Cobertura de código:  
No se implementaron pruebas automatizadas ni herramientas de análisis de cobertura, por lo cual este criterio no fue evaluado en esta versión del proyecto.

## 2.5. ESTIMACIÓN DE RECURSOS

Tabla 6 estimación de recursos. fuente propia.

Actividad	Duración estimada
Reunión inicial y planificación	1 semana
Investigación y análisis de requerimientos	2 semanas
Documentación de requerimientos	1 semana
Diseño del prototipo	3 semanas
Desarrollo de módulos (registro, CRUD, permisos, chatbot)	12 semanas
Pruebas unitarias e integración	2 semanas
Pruebas con usuarios reales	1 semana
Análisis de resultados	1 semana
Ajustes, mejoras e implementación final	2 semanas
Documentación final	2 semanas

Para el desarrollo del presente sistema de chatbot inteligente multirrol, se estimaron y utilizaron recursos humanos y técnicos durante un período total de 6 meses (34 semanas), según el cronograma de ejecución.

- Recursos Humanos:
  - desarrolladores: encargados del backend (FastAPI), frontend (Flutter), entrenamiento de intents y pruebas.
  - 1 asesor académico: seguimiento técnico y metodológico.
  
- Recursos Técnicos:
  - Lenguajes: Python 3.10 y Dart.
  - Frameworks: FastAPI (backend), Flutter (frontend).
  - Herramientas: Visual Studio Code, GitHub, Postman, Figma, draw.io.
  - Librerías principales: bcrypt, JWT, smtplib, TensorFlow para NLP.
  
- Recursos de Almacenamiento:
  - Archivos JSON para gestionar usuarios, materias, horarios, notas, permisos, intents y chatlogs.
  
- Estimación de Tiempo (Distribución por semanas del cronograma):

Total estimado: 27 semanas activas de desarrollo técnico y validación + 7 semanas distribuidas en investigación, planificación y documentación.

## **2.6. VALIDACIÓN INTERNA DEL PROYECTO**

Durante la fase de pruebas finales, se realizaron validaciones internas con usuarios simulados que asumieron los distintos roles definidos en el sistema: estudiante, docente, administrativo y administrador. Estos usuarios interactuaron con el chatbot en diferentes escenarios, como consulta de horario, recuperación de contraseña, solicitudes de permisos y asignación de roles.

Se recopilaron observaciones clave sobre la experiencia de usuario, identificando mejoras necesarias en los mensajes de retroalimentación y ajustes en la interpretación de algunos intents. También se evaluaron los tiempos de respuesta del sistema, los cuales fueron satisfactorios en pruebas controladas. Las correcciones aplicadas durante esta etapa mejoraron la precisión del chatbot y la usabilidad general del sistema.

## **2.7. CONCLUSIONES Y RECOMENDACIONES**

### **Conclusiones**

El desarrollo del sistema de chatbot inteligente multirol permitió comprobar la viabilidad de implementar soluciones tecnológicas basadas en inteligencia artificial y procesamiento de lenguaje natural (PLN) para mejorar los procesos administrativos y académicos dentro de instituciones educativas.

Se logró estructurar una solución modular, segura y funcional que permite automatizar consultas, gestionar permisos, facilitar la interacción entre usuarios y reducir la carga operativa de los

administrativos. El chatbot respondió adecuadamente a las necesidades específicas de cada rol, ofreciendo una experiencia personalizada y eficiente.

Durante la fase de validación, se evidenció que el sistema cumple con los objetivos propuestos, mostrando tiempos de respuesta adecuados, correcta distribución de roles, control de acceso y comunicación efectiva a través del chat. Las pruebas funcionales confirmaron la estabilidad del backend, la integración correcta con el frontend y el uso efectivo de los archivos JSON como medio de almacenamiento.

La metodología de desarrollo aplicada, centrada en el enfoque Design Thinking, permitió entender las necesidades reales de los usuarios desde etapas tempranas y asegurar una solución alineada a su contexto institucional.

### **Recomendaciones**

- Migrar el sistema de archivos JSON a una base de datos relacional para mejorar el rendimiento y escalabilidad.
- Implementar pruebas automatizadas y medición de cobertura de código para asegurar mayor fiabilidad y control.
- Incorporar capacidades de aprendizaje automático para que el chatbot pueda mejorar su comprensión con el tiempo.
- Expandir el sistema a nuevas funcionalidades como manejo de notas, matrículas, integración con sistemas académicos (LMS).
- Realizar capacitaciones institucionales para promover el uso del sistema y asegurar su adopción.
- Registrar la propiedad intelectual del sistema para facilitar su transferencia tecnológica o implementación en otras instituciones.

### **2.8. LIMITACIONES DEL PROTOTIPO**

Aunque el sistema cumple con sus objetivos funcionales, presenta algunas limitaciones propias de esta versión inicial:

- Actualmente, el sistema utiliza archivos JSON como método de almacenamiento. Esto limita la escalabilidad y concurrencia, en comparación con una base de datos relacional.
- No se desarrolló un modo de funcionamiento completamente offline; el sistema depende de conexión a internet para validar acciones e interactuar con el backend.
- Las pruebas realizadas fueron de tipo funcional, ejecutadas manualmente. No se integraron herramientas de pruebas automatizadas ni análisis de cobertura de código.
- El modelo de procesamiento de lenguaje natural está entrenado con intents personalizados, pero no se incorporaron motores de IA avanzados con aprendizaje automático continuo.

## 2.9. PROYECCIONES FUTURAS

Para garantizar la evolución y sostenibilidad del sistema, se proponen las siguientes líneas de mejora y expansión tecnológica:

- Migración a una base de datos relacional (por ejemplo, PostgreSQL o MongoDB) para garantizar mayor rendimiento, integridad y escalabilidad de los datos.
- Implementación de pruebas automatizadas (pytest para backend, integration tests en Flutter) y medición de cobertura de código para fortalecer la validación continua.
- Despliegue multiplataforma (web, escritorio y móvil) para facilitar el acceso en diversos entornos educativos.
- Integración con sistemas académicos existentes (como Moodle o plataformas LMS) para automatizar aún más los procesos de consulta y gestión.
- Incorporación de capacidades avanzadas de IA, como aprendizaje contextual y modelos generativos que mejoren la comprensión y adaptabilidad del chatbot.

### 3.1. BIBLIOGRAFÍA

- Ballesteros Vargas, Elkin Humberto, B.R & C.E. (2018). Diseño de Prototipo de Chat Bot para la Orientación en la Selección de una Carrera de Pregrado.  
<https://repository.udistrital.edu.co/items/b51ddb76-7dbb-45f7-94a1-1aed9425937c>
- Batista, A. R. (2017). Uso de chatbots como apoyo para la comunicación en el aula.  
<http://sedici.unlp.edu.ar/handle/10915/63541>
- Bullon Miguel, Yeni Carmen. (2022). Percepción del modelo CRM y la gestión educativa en la comunicación con estudiantes.  
<https://hdl.handle.net/20.500.12692/77561>
- Castillo Neira, L. (2024). Cliobot, un chatbot de convicción en las aulas.  
<https://hdl.handle.net/1992/74207>
- Castro Quijano, L.A et al. (2021). Chatbot conversacional de apoyo al uso del aula extendida.  
<https://bonga.unisimon.edu.co/items/ffabbce6-f03c-48aa-ba9a-dba734c153d9>
- Congreso de Colombia. (1982). Ley 23 de 1982.  
<https://www.funcionpublica.gov.co/eva/gestornormativo/norma.php?i=331>
- Congreso de Colombia. (2012). Ley 1581 de 2012.  
<https://www.funcionpublica.gov.co/eva/gestornormativo/norma.php?i=49981>
- Congreso de Colombia. (2013). Decreto 1377 de 2013.  
<https://www.funcionpublica.gov.co/eva/gestornormativo/norma.php?i=53762>
- Espinosa Peña, R. A., & Castillo Diaz, M. A. (2024). Implementación de un Chat Bot basado en analítica de datos.  
<https://hdl.handle.net/20.500.12494/55165>
- Flores Orozco, Angel Patricio, G.G & D.E. (2023). Propuesta de mejora en el proceso de atención estudiantil.  
<https://repositorio.epnewman.edu.pe/handle/20.500.12892/706>
- G. Casseres, J. et al. (2018). Chatbot que facilita la información en la Facultad de Ingenierías.  
<https://revistas.unisimon.edu.co/index.php/identific/article/download/3480/4144>

- García-Reina, L. F. (2018). Asistente virtual de tipo ChatBot. <https://repository.ucatolica.edu.co/entities/publication/16e194ec-44d2-40c7-abf0-554958b9652e>
- García, J., López, M., & Pérez, A. (2021). Transformación digital en instituciones educativas: Retos y oportunidades. Revista de Educación y Tecnología.
- Gibson, R. (2024). The Impact of AI in Advancing Accessibility for Learners with Disabilities. EDUCAUSE Review. <https://er.educause.edu/articles/2024/9/the-impact-of-ai-in-advancing-accessibility-for-learners-with-disabilities>
- Gil Campoverde, L. A. (2022). Chatbot para mejorar la interacción académico-virtual. <https://hdl.handle.net/20.500.12692/100353>
- IBM. (2024). Conversational AI Chatbot. <https://www.ibm.com/cloud/learn/chatbot>
- IBM. (2024). Natural Language Processing (NLP) Solutions. <https://www.ibm.com/cloud/learn/natural-language-processing>
- IBM. (2024). What Is Artificial Intelligence (AI)?. <https://www.ibm.com/cloud/learn/what-is-artificial-intelligence>
- Isusqui, J. C. P. et al. (2023). La Inteligencia Artificial al servicio de la gestión educativa. <https://doi.org/10.31219/osf.io/z2y7c>
- Jamanoy Bacca, B.C., & Montenegro Rosero, J. F. (2023). Chatbot aplicando PLN. <http://repositorio.unicesmag.edu.co:8080/xmlui/handle/123456789/998>
- León Granizo, O. D., & Pazmiño Muguersa, D. A. (2021). Propuesta de un chatbot académico. <http://repositorio.unemi.edu.ec/handle/123456789/6024>
- Lino Noboa, J. F. (2023). Chatbot con inteligencia artificial para documentos escolares. <https://hdl.handle.net/20.500.12892/834>
- Martínez, A., López, R., & Torres, S. (2022). Retos para la gestión educativa. <https://www.semanticscholar.org/paper/Retos-para-la-gestión-de-las-instituciones-de-y-Sánchez-Buitrago-Narváez/a9dfbbc3dfc26066933f65c63b73ca1a98c23efd>
- Pazmiño Ricaurte, J. D. (2020). Chatbot integrado a un LMS. <http://dspace.udla.edu.ec/handle/33000/13235>
- Peñaherrera Acurio, W. P. et al. (2022). Implementación de IA como recurso educativo. [https://doi.org/10.26820/recimundo/6.\(2\).abr.2022.402-413](https://doi.org/10.26820/recimundo/6.(2).abr.2022.402-413)
- Ramírez, J., Gómez, P., & Silva, T. (2015). Prototipo de software de gestión educativa. <https://www.semanticscholar.org/paper/Diseño-y-desarrollo-de-un-prototipo-de-software-de-Torres-Romero/efc978097cb3b76a4e64810ec8bac0656aea4ae7>
- Red Hat. (2022). What is an API?. <https://www.redhat.com/en/topics/api/what-are-application-programming-interfaces>
- Red Hat. (2023). Back-end Development. <https://www.redhat.com/en/topics/cloud-native-apps/back-end-development>
- Rincón Rico, E., & Molina González, M. (2022). Chatbot para competencias ciudadanas. <https://repository.usta.edu.co/handle/11634/47547>
- Sánchez Hurtado, R. A. (2021). Chatbot para comunicación universitaria. <https://hdl.handle.net/20.500.14138/4882>
- Techopedia. (2024). Create, Read, Update and Delete (CRUD). <https://www.techopedia.com/definition/23879/create-read-update-and-delete-crud>
- Techopedia. (2024). What is an Information System (IS)?. <https://www.techopedia.com/definition/4357/information-system-is>

- TensorFlow. (2024). TensorFlow License - Apache License 2.0.  
<https://www.tensorflow.org/about/license>
- Tiangolo, S. (2018). FastAPI License (MIT).  
<https://github.com/tiangolo/fastapi/blob/master/LICENSE>
- UNESCO. (2020). Educación en tiempos de pandemia.  
<https://unesdoc.unesco.org/ark:/48223/pf0000374075>

## 4. ANEXOS

### 4.1. MANUALES

#### **4.1.1. *Manual técnico***

**Desarrollo de un Prototipo de chat  
inteligente para la Optimización y Mejora de  
Procesos Administrativos y Operativos en  
una Institución Educativa.**

*Fecha: 2025/04/28*



**Manual tecnico**

**Desarrollo de Sistemas de Información  
Dirección de Sistemas y Tecnología**

## TABLA DE CONTENIDO

Presentación.....	3
1.Resumen.....	3
2.Objetivo.....	3
3.Finalidad.....	3
4 Introducción.....	4
5,Aspectos Tecnicos.....	4
5.1Herramientas usadas.....	4
5.1.1 Visual Studio Code (VS Code).....	4
5.1.2 Flutter.....	4
5.1.3 FastAPI.....	5
6. Requerimientos de software.....	5
6.1 Requisitos mínimos.....	5
Bibliografía.....	5

### Presentación

El manual está presente para solucionar cualquier duda al respecto de la forma de organización en la que se **opera la página del inventario** teniendo en cuenta los entornos los entornos que se implementaron para su funcionalidad y las reglas a seguir del sistema.

### 1.Resumen

El manual detalla los aspectos técnicos e informáticos y se busca la forma de explicar la forma en cómo se estructura la forma en la que se planteó el cómo se organiza y se formó, se mostrarán las herramientas usadas para la planeación y creación de software las cuales manejan diferentes funcionalidades dentro del aplicativo creado.

## 2.Objetivo

Es dar a conocer el uso correcto de la plataforma web de un inventariado, mostrando de una forma más técnica y con una descripción más gráfica los componentes los cuales son los capaces de permitir el funcionamiento del programa.

## 3.Finalidad

Lo que se espera lograr con el manual es lograr que se pueda instruir correctamente sobre las funciones y sobre que tipo de requerimiento busca satisfacer.

## 4 Introducción

Se busca en este manual realizar la explicación de las partes técnicas que conforman el programa ya que estos explican la forma en que se organizaron las ideas esenciales para formular la estructura que se debe programar.

## 5,Aspectos Tecnicos

De Acuerdo a la finalidad del software de inventario planteado se espera mejorar en el manejo de los productos allí ingresados los cuales serán recibidos por medio de proveedores debidamente registrados a través de una plataforma en el mismo programa por lo cual se espera que cualquier información relacionada a los productos o de donde provienen sea manipulada por un administrador también registrado en la plataforma.

### 5.1Herramientas usadas

El desarrollo y creación del programa fue necesario el uso de herramientas de software para su modelado y ejecución.

5.1.1 Visual Studio Code (VS Code). Es un editor de código fuente desarrollado por Microsoft, reconocido por ser ligero, rápido y altamente personalizable. Está diseñado para facilitar la escritura, edición y depuración de código en una amplia variedad de lenguajes de programación, como JavaScript, Python, C + +, entre muchos otros.

La versión utilizada fue la 1.97.2 en la cual se utilizó los depuradores y compiladores

5.1.2 Flutter. Un framework de desarrollo de software creado por Google que permite construir aplicaciones nativas para dispositivos móviles (Android e iOS), web y escritorio a partir de una única base de código, utilizando el lenguaje de programación Dart.

5.1.3 FastAPI. Un framework moderno y de alto rendimiento para construir APIs web utilizando Python. Se destaca por su velocidad, facilidad de uso y soporte nativo para programación asíncrona, lo que lo convierte en una opción ideal para desarrollar APIs RESTful

## 6. Requerimientos de software

Se detallan los requisitos necesarios necesarios para que el programa pueda ser ejecutado correctamente

### 6.1 Requisitos mínimos

Requisitos para ejecutar la aplicación en producción.

.Memoria RAM : 4GB

.Disco duro : 4 GB

.Sistema operativo: windows 7 /8 /10. Android 14

.Procesador : intel core 3 / Ryzen 3

## Bibliografía

Zúñiga. (2024) ¿Qué es Visual Studio Code y cuáles son sus ventajas?.recuperado de.<https://www.arsys.es/blog/que-es-visual-studio-code-y-cuales-son-sus-ventajas>

Lesly.(2025).Mis Preferencias de Visual Studio Code para Desarrollo en Flutter.recuperado de.[https://es.linkedin.com/posts/leslysamaritano\\_mis-preferencias-de-visual-studio-code-para-activity-7290555223746510848-hWFC](https://es.linkedin.com/posts/leslysamaritano_mis-preferencias-de-visual-studio-code-para-activity-7290555223746510848-hWFC)

refine (2025),An Introduction to Using FastAPI. recuperado  
de:<https://refine.dev/blog/introduction-to-fast-api/#django-vs-fastapi>

### **DATOS DE CONTACTO SOPORTE APLICACIÓN**

Dirección de sistemas y tecnología

Áreas de sistemas de información

Extensiones telefónicas: 170 y 256

correo: [bapatino@ucundinamarca.edu.co](mailto:bapatino@ucundinamarca.edu.co)

[luisalejandrogonzalez@ucundinamarca.edu.co](mailto:luisalejandrogonzalez@ucundinamarca.edu.co)



**DIRECCIÓN DE SISTEMAS  
Y TECNOLOGÍA**

#### **4.1.2. Manual de usuario**

**Desarrollo de un Prototipo de chat inteligente  
para la Optimización y Mejora de Procesos  
Administrativos y Operativos en una Institución  
Educativa.**

*Fecha: 2025/04/28*



**Manual de Usuario  
Desarrollo de Sistemas de Información  
Dirección de Sistemas y Tecnología**

**TABLA DE CONTENIDO**

1. Página de inicio.....	3
2. Interfaz de usuario.....	3
• ¿Cómo registrar usuario?.....	3
• ¿Cómo recuperar la contraseña del usuario?.....	7
3. Interfaz de aplicativo.....	11
• Página principal.....	11
• Botón para cerrar la sesión.....	11
4. Eliminar usuario (únicamente para el administrador).....	12
5. Activar y asignar rol a un usuario (únicamente para el administrador).....	14
DATOS DE CONTACTO SOPORTE APLICACIÓN.....	15

### Tabla de figuras

Figura 1.....	3
Figura 2.....	4
Figura 3.....	5
Figura 4.....	5
Figura 5.....	6
Figura 6.....	7
Figura 7.....	8
Figura 8.....	8
Figura 9.....	9
Figura 10.....	9
Figura 11.....	10
Figura 12.....	11
Figura 13.....	12
Figura 14.....	12
Figura 15.....	13
Figura 16.....	14

#### 1. Página de inicio

#### 2. Interfaz de usuario

- ¿Cómo registrar usuario?

En la parte inferior del login se encuentra el link para empezar el registro y crear una cuenta.

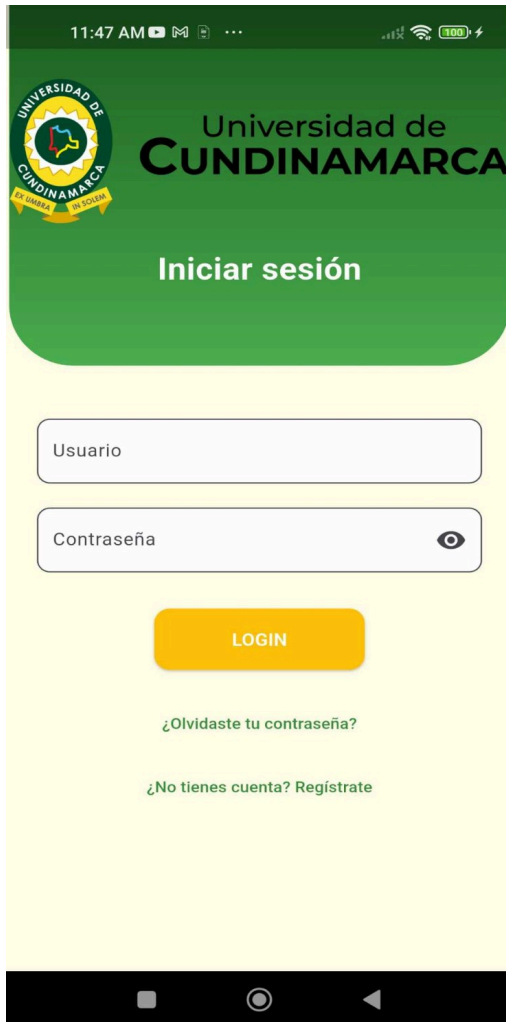


Figura 1. inicio de sesión. fuente propia

Una vez dentro ingresamos los datos solicitados

The image shows a registration form with a green header bar containing a back arrow and the title 'Registro'. The form fields are as follows:

- Nombre completo
- Usuario
- Correo institucional
- Enviar código (button)
- Código recibido
- Contraseña
- Confirmar contraseña
- Pregunta de seguridad (dropdown menu)
- Respuesta
- Acepto los términos y condiciones
- Registrarse (button)

Figura 2. registro. fuente propia

También se solicita ingresar un código que fue enviado al correo registrado previamente.

Hola,

Gracias por registrarte en nuestro Chat Inteligente.

Tu código de verificación es:

🔒 162696 🔒

Por favor ingrésalo en la aplicación para completar tu registro.

Si no solicitaste este código, puedes ignorar este mensaje.

¡Bienvenido!

Atentamente,  
Equipo Chat Inteligente

Figura 3 Verificación. fuente propia



Figura 4 en espera. fuente propia

Al final ya con toda la información registrada aceptamos los términos y condiciones relacionados al tratamiento de los datos personales.

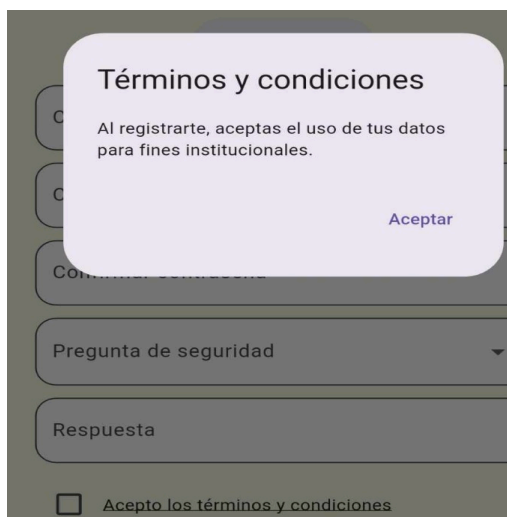


Figura 5 términos y condiciones. fuente propia

Se espera a la asignación que se confirma a través de un correo

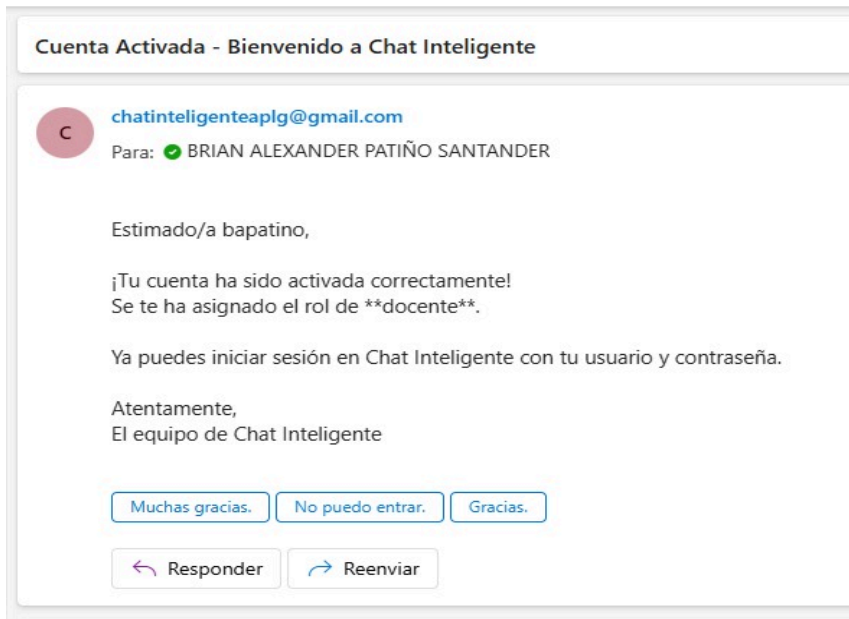
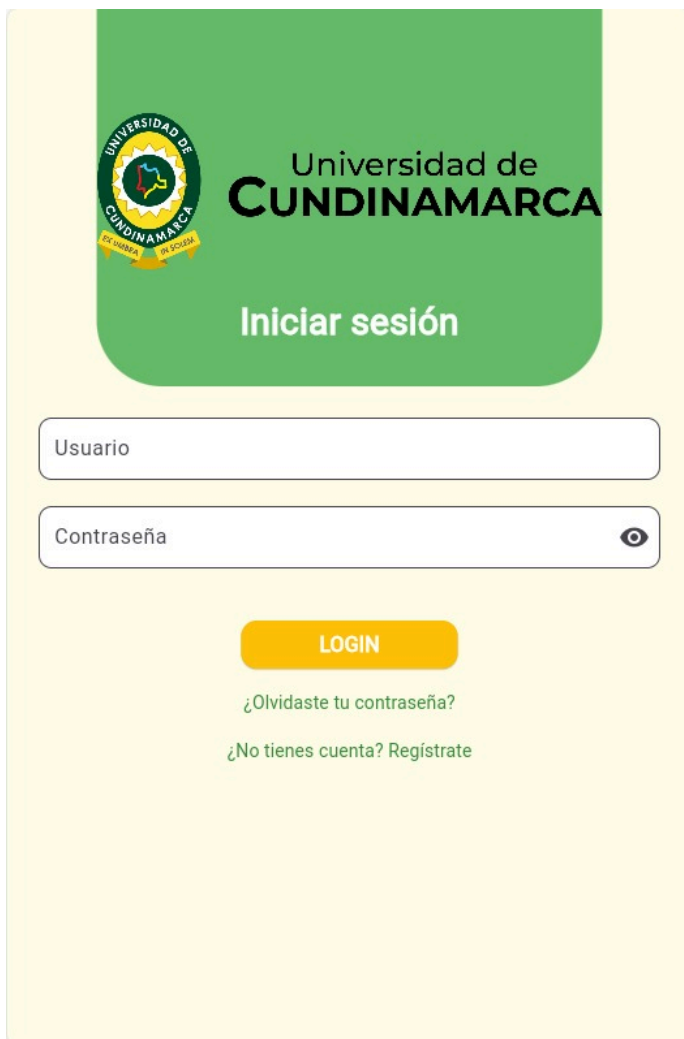


Figura 6 Asignación de rol. fuente propia

- ¿Cómo iniciar sesión el usuario?

Una vez realizado el proceso de registro volvemos al inicio e ingresamos usuario y contraseña en los espacios asignados.



The image shows a login interface for the Universidad de Cundinamarca. At the top left is the university's logo, which features a circular emblem with a stylized figure and the text 'UNIVERSIDAD DE CUNDINAMARCA' and 'ESTUDIA EN FUERZA'. To the right of the logo, the text 'Universidad de CUNDINAMARCA' is displayed in a bold, sans-serif font. Below this, the text 'Iniciar sesión' is centered in a white font on a green background. The main form area has a light yellow background and contains two input fields: 'Usuario' and 'Contraseña'. The 'Contraseña' field includes a toggle icon for visibility. Below the fields is a yellow 'LOGIN' button. At the bottom, there are two links: '¿Olvidaste tu contraseña?' and '¿No tienes cuenta? Regístrate'.

UNIVERSIDAD DE  
CUNDINAMARCA  
ESTUDIA EN FUERZA

Universidad de  
**CUNDINAMARCA**

Iniciar sesión

Usuario

Contraseña

LOGIN

[¿Olvidaste tu contraseña?](#)

[¿No tienes cuenta? Regístrate](#)

Figura 7 inicio de sesión. fuente propia  
y finalmente iniciamos

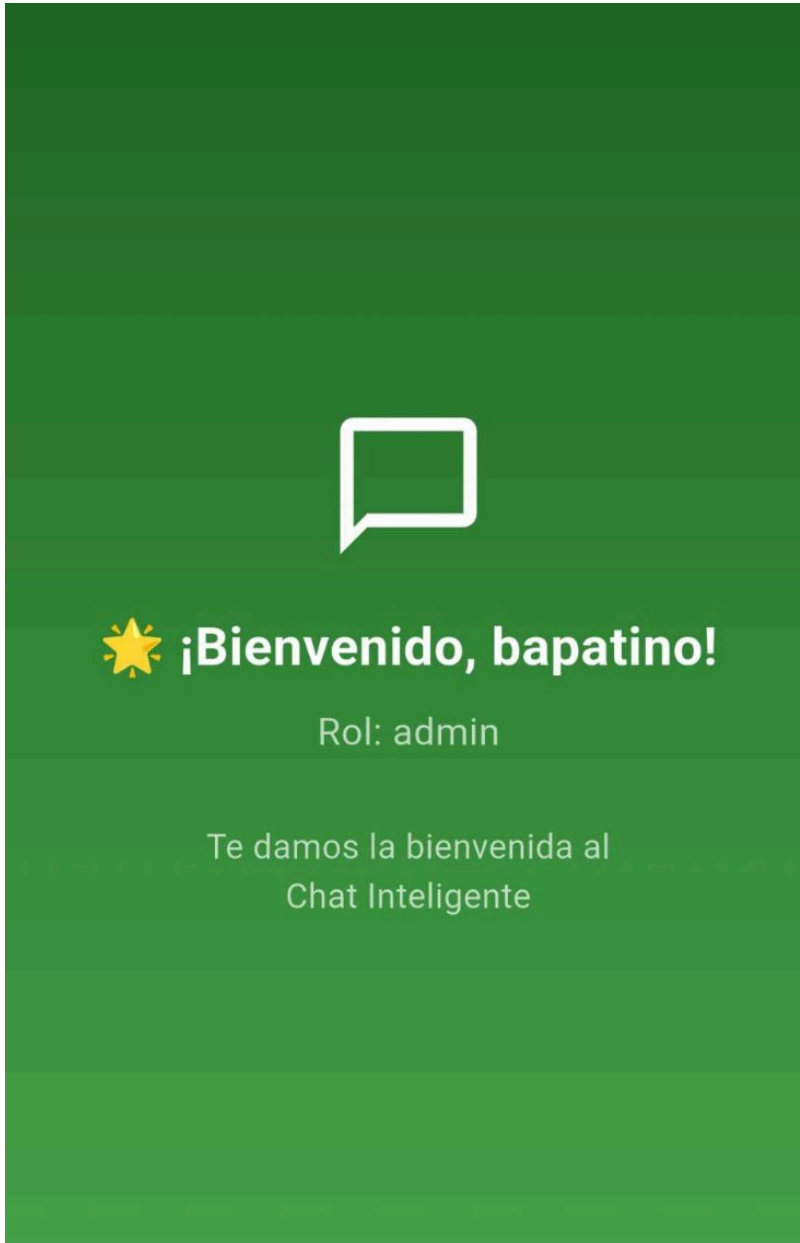
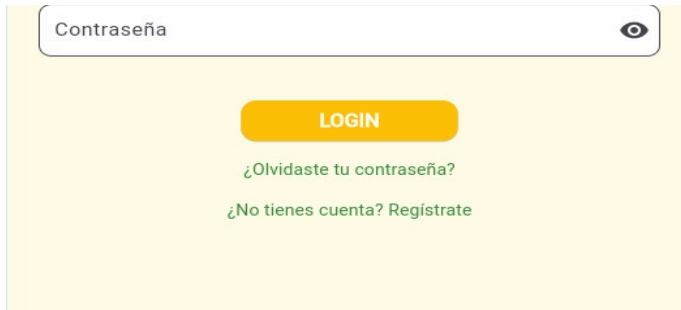


Figura 8 pantalla de inicio.fuente propia

- ¿Cómo recuperar la contraseña del usuario?

Para recuperar contraseña ingresamos en la sección que nos pregunta si la hemos olvidado.



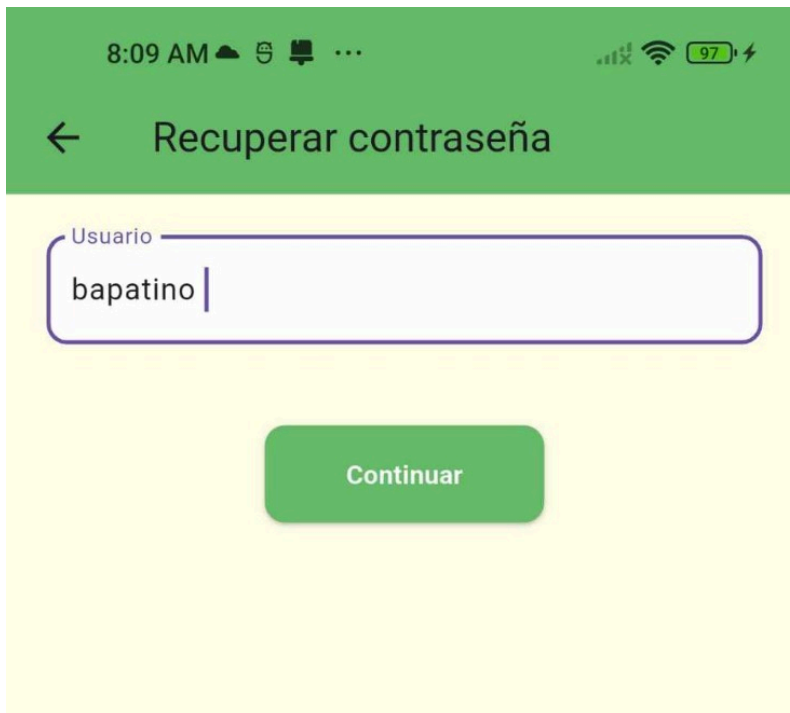
Contraseña

LOGIN

¿Olvidaste tu contraseña?

¿No tienes cuenta? Regístrate

Figura 9. Recuperar. Fuente propia  
Luego escribimos el usuario de la persona



8:09 AM

← Recuperar contraseña

Usuario

bapatino

Continuar

Figura 10. Recuperar contraseña. Fuente propia

Luego respondemos a la pregunta de seguridad que ya se había establecido en el registro

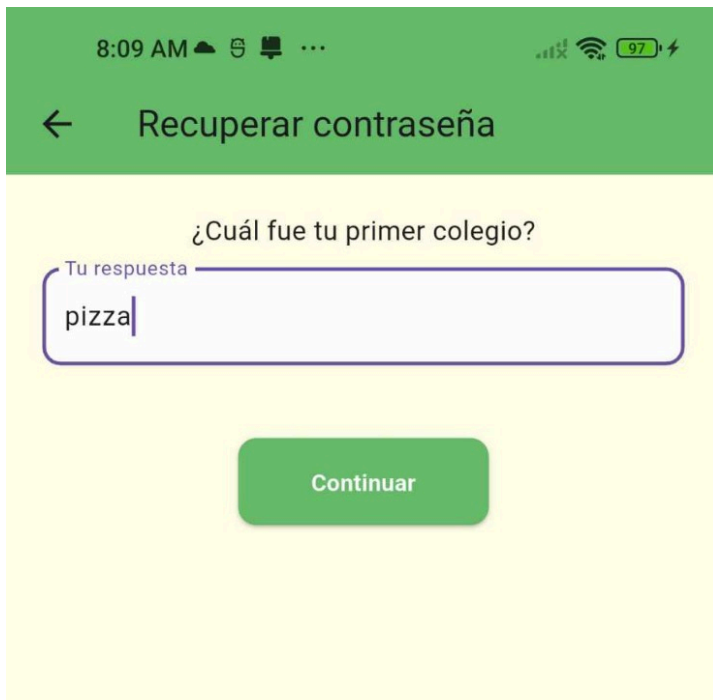


Figura 11. Pregunta. Fuente propia

Se envía un código al correo para ingresar

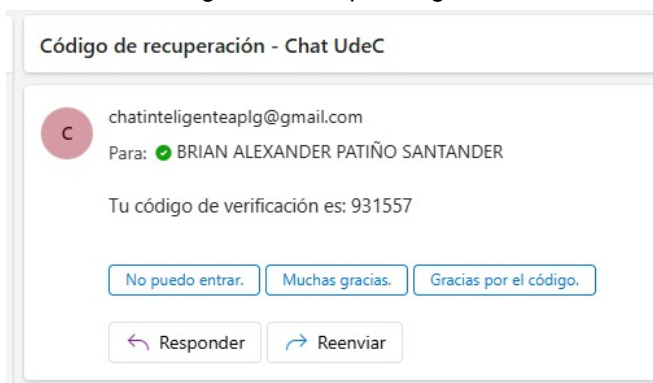


Figura 12. Código correo. Fuente propia



10:13 AM

← Recuperar contraseña

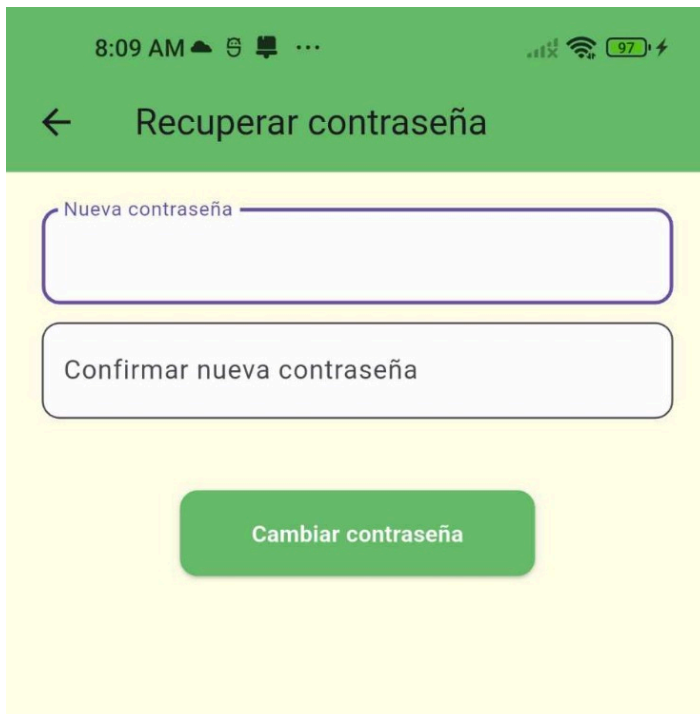
Código recibido por correo

105067

Continuar

Figura 13. Código de recuperación. Fuente propia

Luego ingresamos la nueva contraseña



8:09 AM

← Recuperar contraseña

Nueva contraseña

Confirmar nueva contraseña

Cambiar contraseña

97%

Figura 14 nueva contraseña. Fuente propia

Luego se regresa al inicio e ingresa las nuevas credenciales de autenticación.

### 3. Interfaz de aplicativo

- Página principal

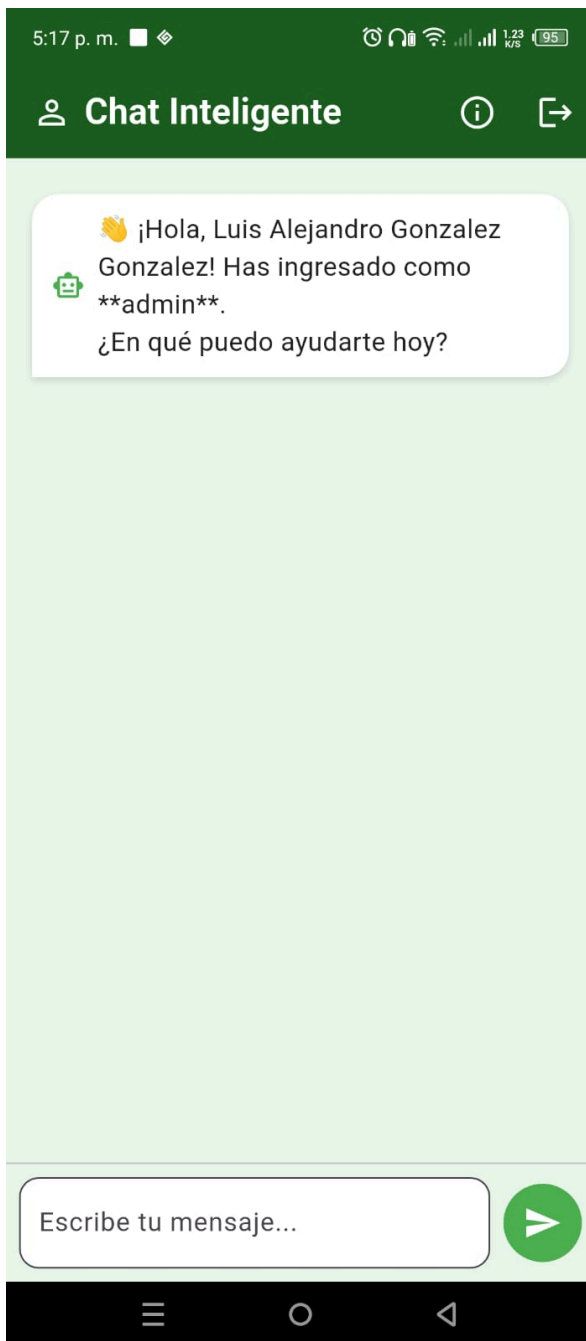


Figura 15 interfaz. Fuente propia

- Botón para cerrar la sesión

En la parte superior de la derecha se encuentra el botón para regresar al login

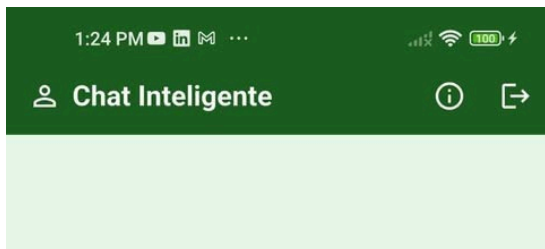


Figura 16 fin de sesión. Fuente propia

#### 4. Eliminar usuario (únicamente para el administrador)

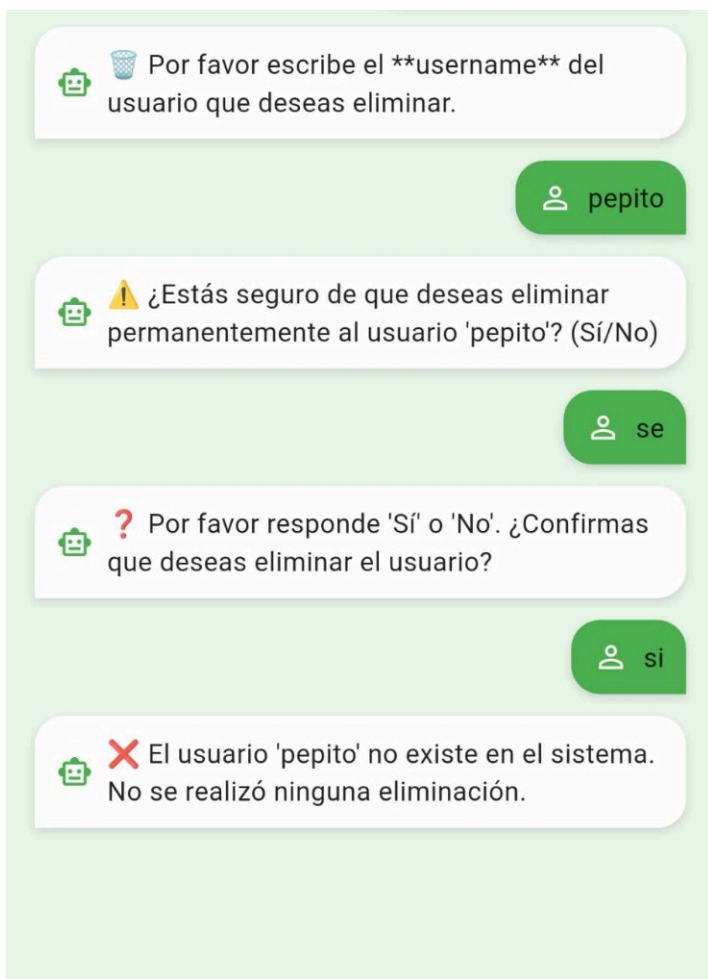


Figura 17 eliminar. Fuente propia

Se enviará un correo de confirmación cuando esto ocurra

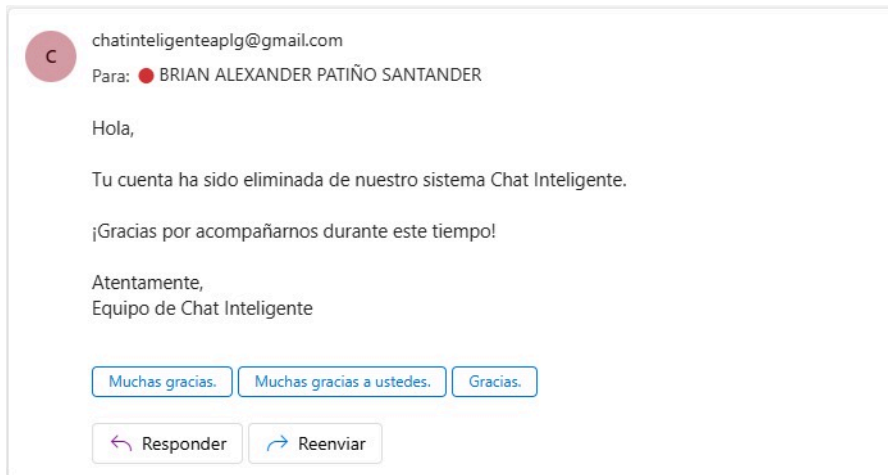


Figura 18. Confirmación. Fuente propia

### 5. Activar y asignar rol a un usuario (únicamente para el administrador)

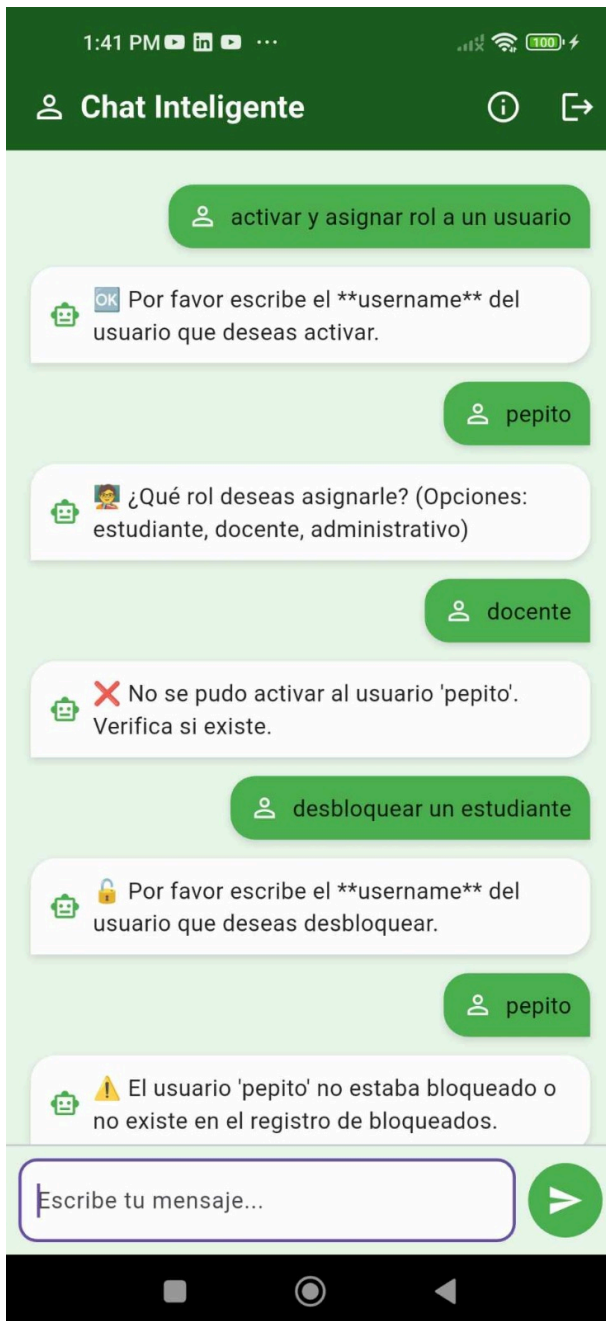


Figura 19 asignación. Fuente propia

### DATOS DE CONTACTO SOPORTE APLICACIÓN

Dirección de sistemas y tecnología  
Áreas de sistemas de información  
Extensiones telefónicas: 170 y 256  
correo: [bapatino@ucundinamarca.edu.co](mailto:bapatino@ucundinamarca.edu.co)  
[luisalejandrogonzalez@ucundinamarca.edu.co](mailto:luisalejandrogonzalez@ucundinamarca.edu.co)

## 4.2 SOLICITUD DE DERECHOS DE AUTOR

cvLAC:

Luis Alejandro González González

 Con los datos indicados ya se encuentra una hoja de vida registrada en el aplicativo CvLAC. A continuación, se presentan los datos básicos:

### Currículo encontrado

Nombres	Luis Alejandro
Primer apellido	Gonzalez
Segundo apellido	Gonzalez
Nombre en citas bibliográficas	Luis Gonzalez
Nacionalidad	Colombiana
Tipo documento	Cédula de Ciudadanía
Documento de identificación	1003777589


Brian Alexander Patiño Santander

 Con los datos indicados ya se encuentra una hoja de vida registrada en el aplicativo CvLAC. A continuación, se presentan los datos básicos:

### Currículo encontrado

Nombres	Brian Alexander
Primer apellido	Patiño
Segundo apellido	Santander
Nombre en citas bibliográficas	Alexander Patiño
Nacionalidad	Colombiana
Tipo documento	Cédula de Ciudadanía
Documento de identificación	1007509943

[Nueva búsqueda](#)

 Nota: Puede recuperar la contraseña a través de la opción "[¿Olvidó su contraseña?](#)". Allí el sistema remitirá la contraseña al correo registrado previamente en el aplicativo. En caso de que el correo haya cambiado o presente inconvenientes, puede enviar la solicitud por medio del formulario de PQRDS ubicado en la ruta <https://minciencias.gov.co/contact>

## 4.3 PRUEBAS TÉCNICAS

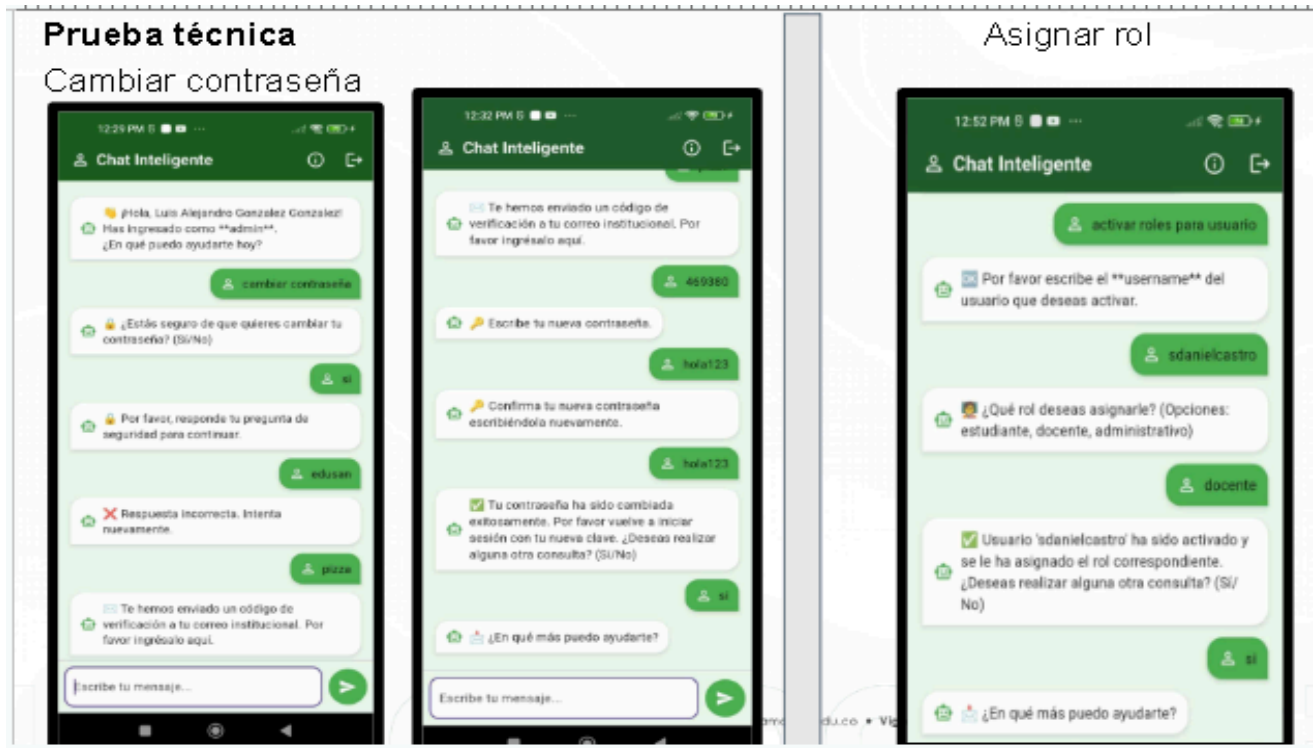


Figura 20. Prueba técnica #1. Fuente propia

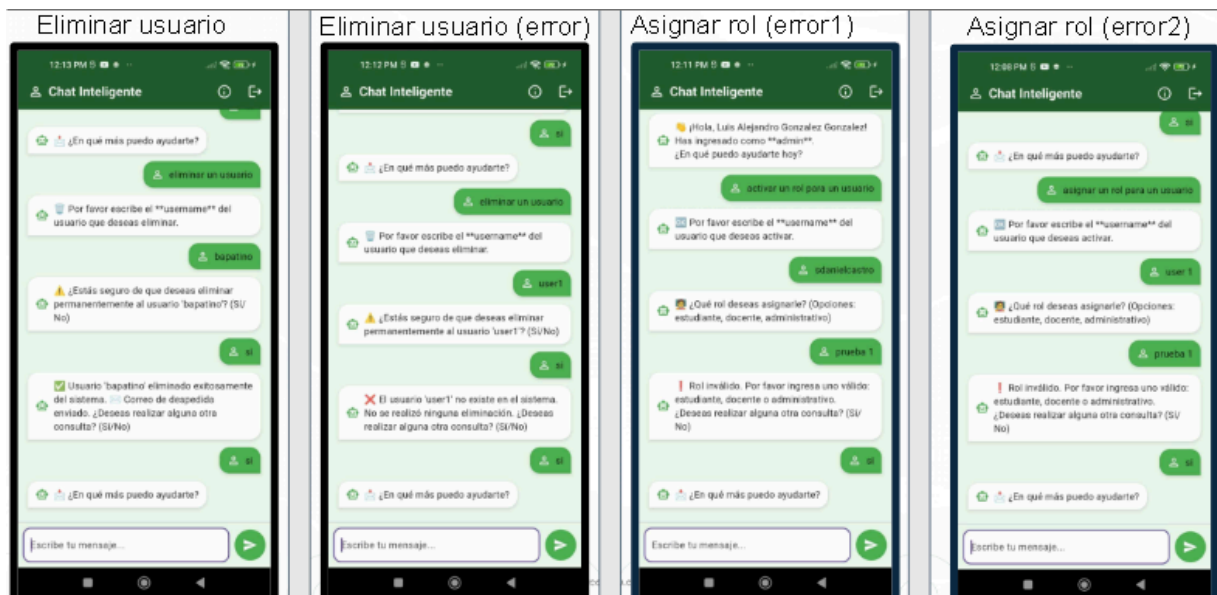


Figura 21. Prueba técnica #2. Fuente propia

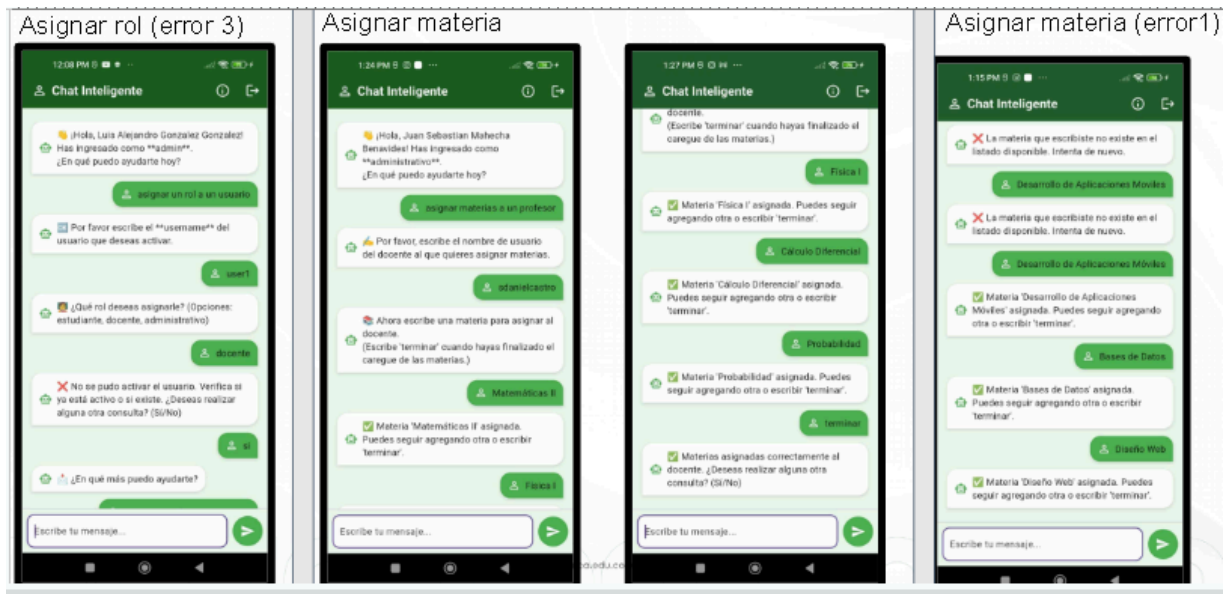


Figura 22. Prueba técnica #3 propia

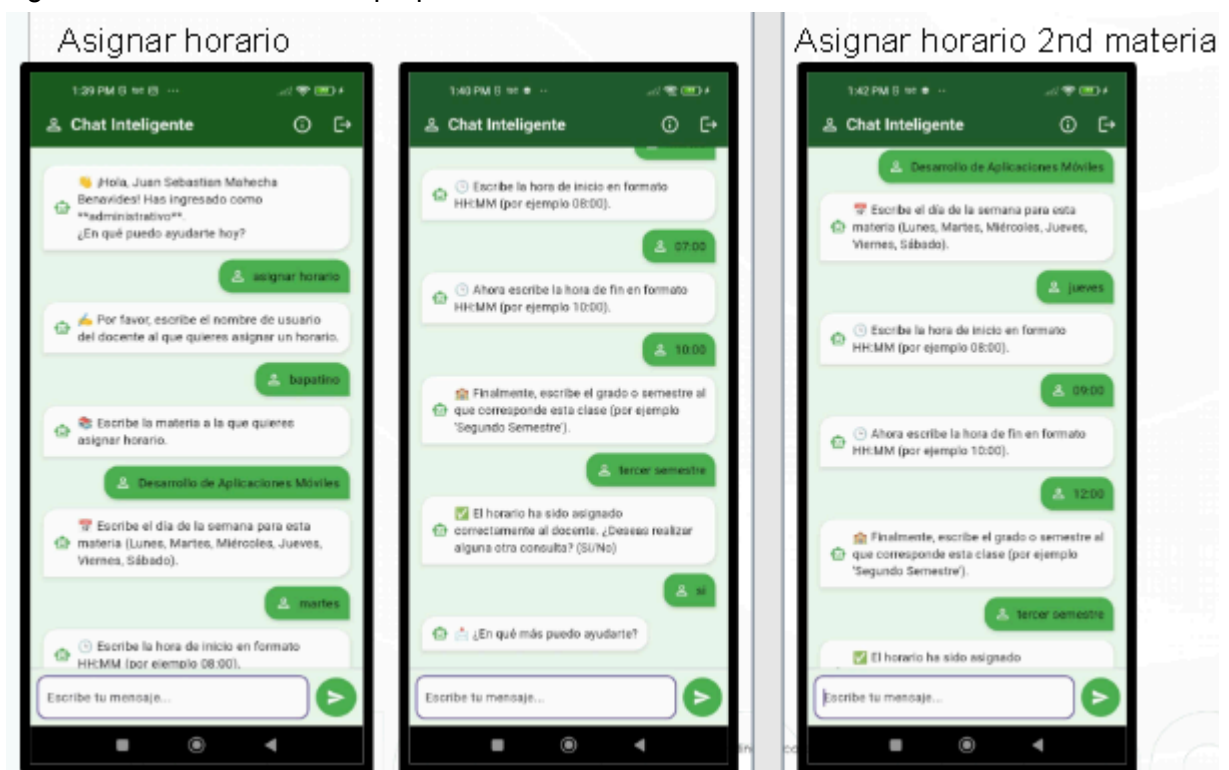


Figura 23. Prueba técnica #4. Fuente propia

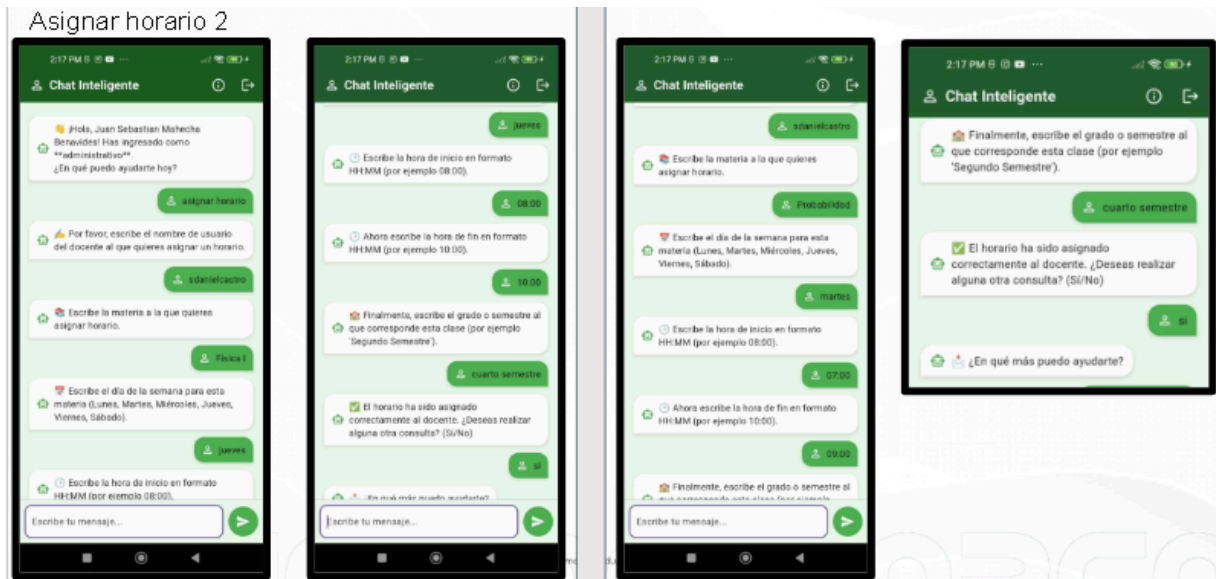


Figura 24. Prueba técnica #5. Fuente propia



Figura 25. Prueba. Prueba técnica #6. Fuente propia

Asignar horario estudiante

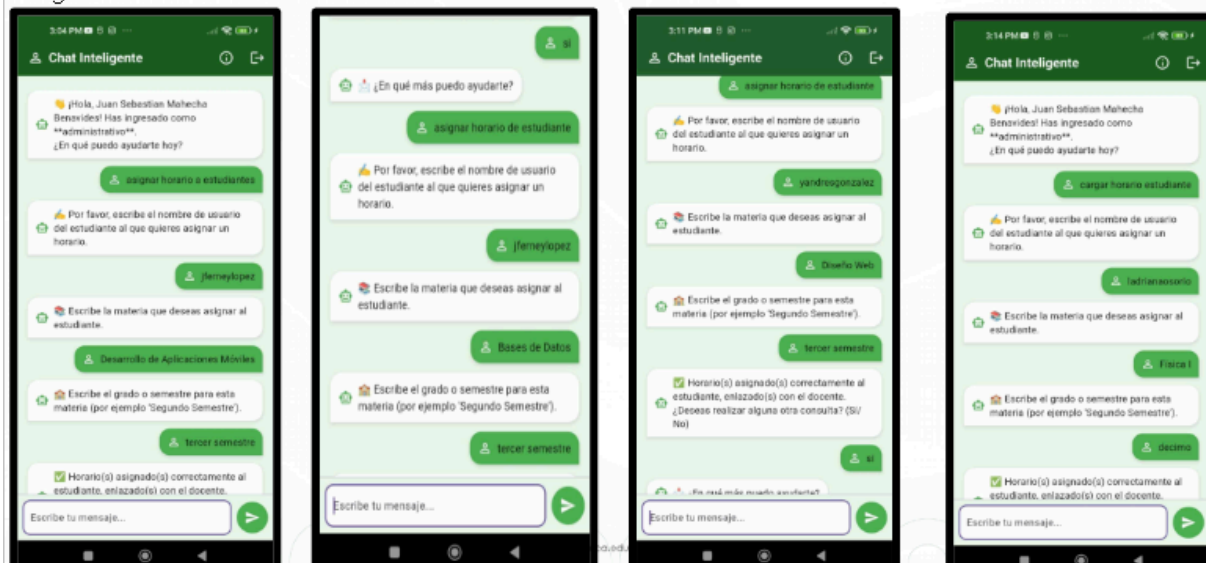


Figura 26. Prueba técnica #7. Fuente propia

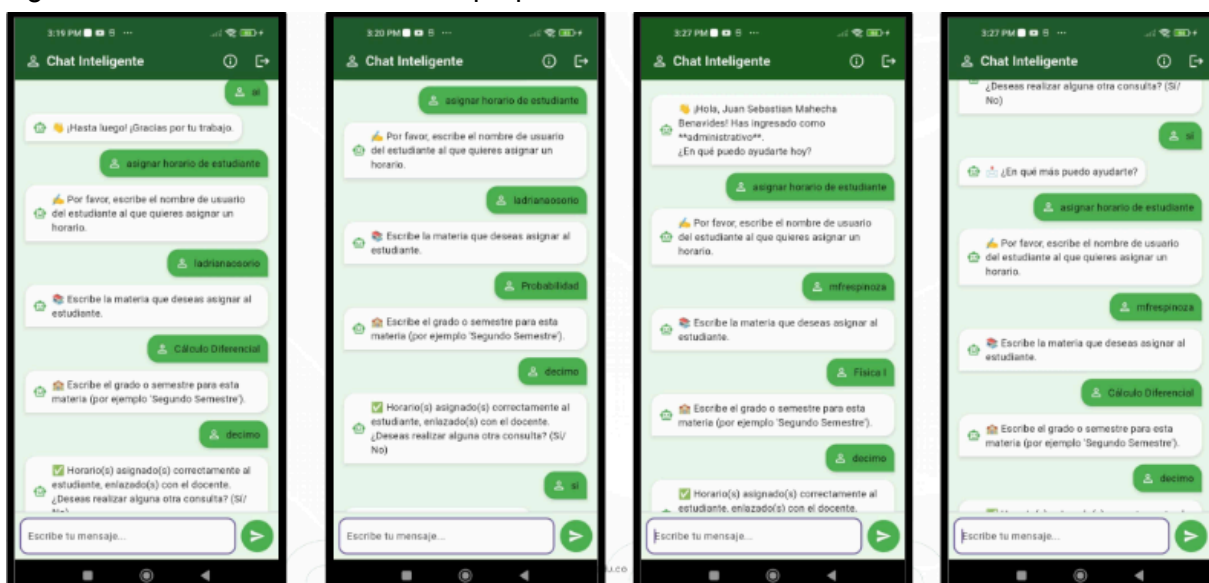


Figura 27. Prueba técnica #8. Fuente propia



Figura 28. Prueba técnica #9 propia







Figura 33. Prueba técnica #14. Fuente propia

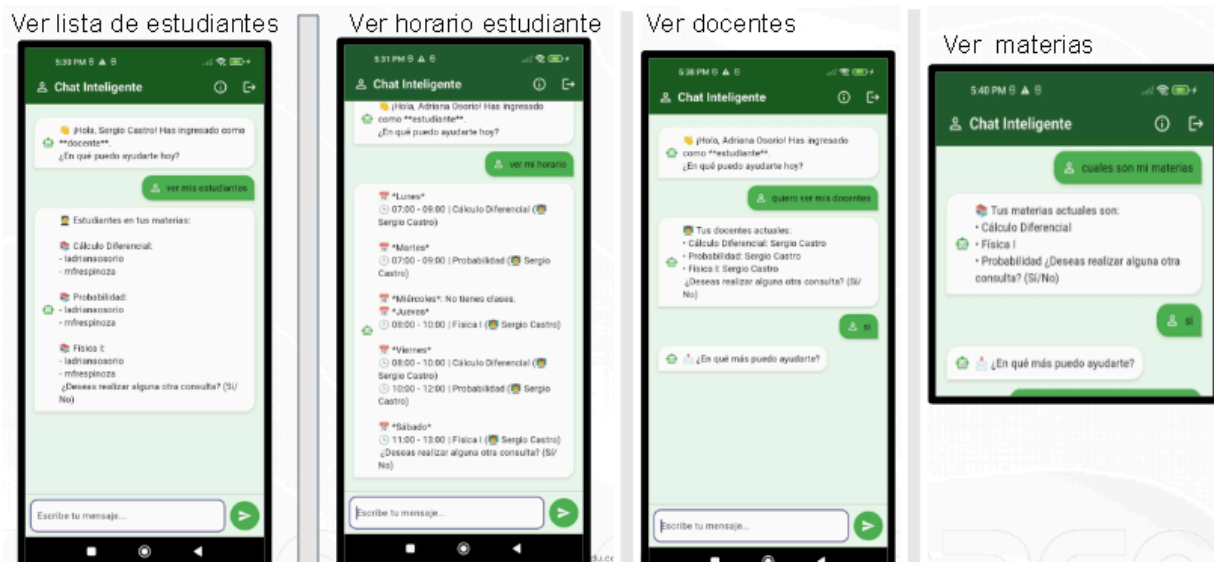


Figura 34. Prueba técnica #15. Fuente propia

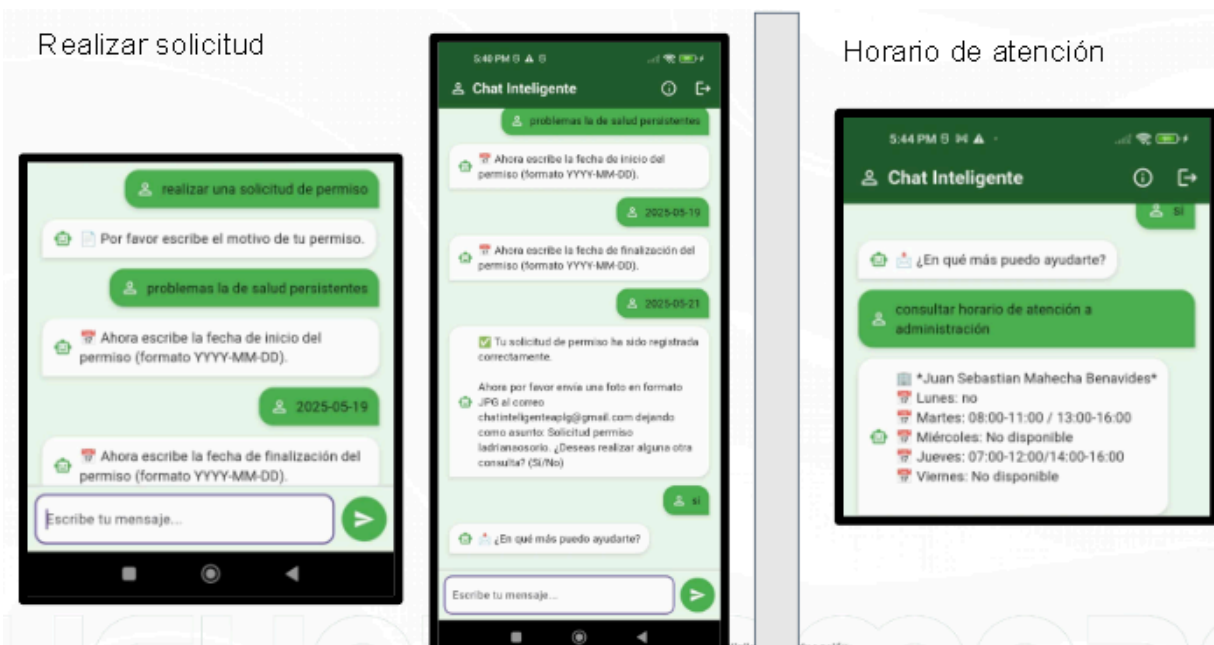


Figura 35. Prueba técnica #16. Fuente propia

Ver mis notas

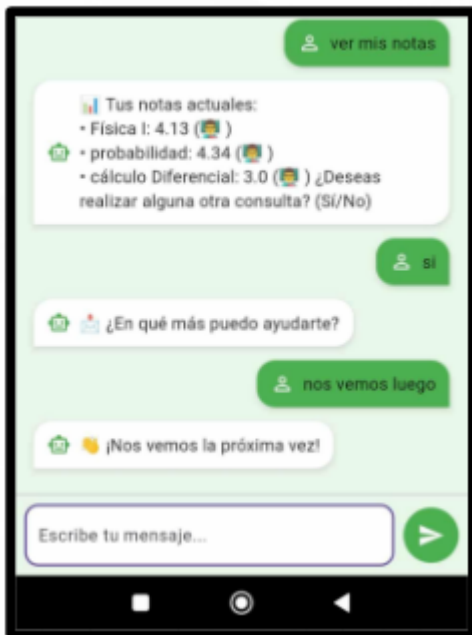


Figura 36. Prueba técnica #17. Fuente propia