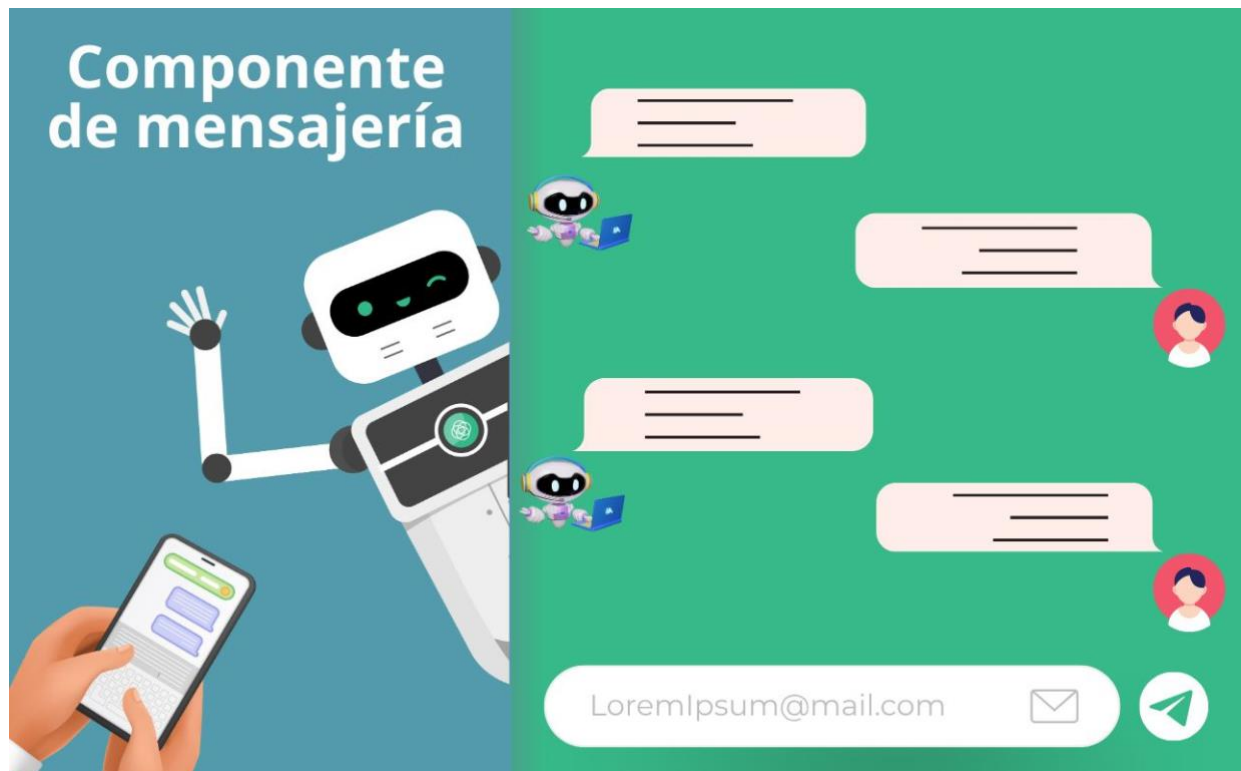


Documento de Propuesta de Diseño de Software I, II y II

Componente de mensajería

Autores: Jaider Martínez Paternina - María Camila Salgado Montiel - Andrés Felipe López León



Descripción del software

Este documento ofrece una exhaustiva exposición sobre la concepción, diseño y ejecución de un componente de mensajería de vanguardia, que integra un chat-bot impulsado por inteligencia artificial y un servicio de soporte brindado por expertos. La génesis de este componente responde a la necesidad imperante de optimizar la experiencia del usuario, a través de la entrega ágil y eficiente de respuestas pertinentes a sus interrogantes.

El enfoque innovador de este proyecto radica en la sinergia entre la avanzada tecnología de chat-bot basado en inteligencia artificial y la intervención humana de expertos en el servicio de soporte. Esta integración no solo permite la resolución instantánea de consultas comunes mediante el chat-bot, sino que también asegura una atención personalizada y de alta calidad cuando se requiere la intervención humana para cuestiones más complejas o específicas.

ETAPA 1 DISEÑO DE LA APLICACIÓN Y ANÁLISIS DE REQUISITOS	6
1. 6	
PROPÓSITO DEL DOCUMENTO	6
ALCANCE DEL PROYECTO MÓDULO DE PIZARRA COMPARTIDA	8
DEFINICIONES Y ACRÓNIMOS	8
2. 8	
OBJETIVOS DEL SISTEMA	10
FUNCIONALIDAD GENERAL	10
USUARIOS DEL SISTEMA	11
RESTRICCIONES	12
3. 10	
CASOS DE USO	13
DIAGRAMAS DE FLUJO DE CASOS DE USO	14
DESCRIPCIÓN DETALLADA DE CADA CASO DE USO	14
PRIORIDAD DE REQUERIMIENTOS	16
4. 26	
REQUISITOS DE DESEMPEÑO	18
REQUISITOS DE SEGURIDAD	19
REQUISITOS DE USABILIDAD	20
REQUISITOS DE ESCALABILIDAD	20
5. 30	
DIAGRAMA DE ENTIDAD-RELACIÓN	21
DIAGRAMA RELACIONAL	22
SCRIPT DE MODELO RELACIONAL	23
DESCRIPCIÓN DE ENTIDADES Y RELACIONES	24
REGLAS DE INTEGRIDAD REFERENCIAL	25
COLECCIONES (NoSQL)	28
6. 40	
DIAGRAMAS ADICIONALES	29
REFERENCIAS	29
ETAPA 2: PERSISTENCIA DE DATOS CON BACKEND	30
7. 41	
PROPÓSITO DE LA ETAPA	30
ALCANCE DE LA ETAPA	30
DEFINICIONES Y ACRÓNIMOS	30
8. 41	
DESCRIPCIÓN DE LA ARQUITECTURA PROPUESTA	30

COMPONENTES DEL BACKEND	30
DIAGRAMAS DE ARQUITECTURA	30
9. 41	
EVALUACIÓN DE OPCIONES (SQL o NoSQL)	31
JUSTIFICACIÓN DE LA ELECCIÓN	31
DISEÑO DE ESQUEMA DE BASE DE DATOS	31
10. 42	
ELECCIÓN DEL LENGUAJE DE PROGRAMACIÓN	31
CREACIÓN DE LA LÓGICA DE NEGOCIO	31
DESARROLLO DE ENDPOINTS Y APIs	31
AUTENTICACIÓN Y AUTORIZACIÓN	31
11. 42	
CONFIGURACIÓN DE LA CONEXIÓN	32
DESARROLLO DE OPERACIONES CRUD	32
MANEJO DE TRANSACCIONES	32
12. 43	
DISEÑO DE CASOS DE PRUEBA	32
EJECUCIÓN DE PRUEBAS UNITARIAS Y DE INTEGRACIÓN	32
MANEJO DE ERRORES Y EXCEPCIONES	32
ETAPA 3: CONSUMO DE DATOS Y DESARROLLO FRONTEND	33
13. 44	
PROPÓSITO DE LA ETAPA	33
ALCANCE DE LA ETAPA	33
DEFINICIONES Y ACRÓNIMOS	33
14. 44	
DISEÑO DE LA INTERFAZ DE USUARIO (UI) CON HTML Y CSS	33
CONSIDERACIONES DE USABILIDAD	33
MAQUETACIÓN RESPONSIVA	33
15. 45	
DESARROLLO DE LA LÓGICA DEL FRONTEND	34
MANEJO DE EVENTOS Y COMPORTAMIENTOS DINÁMICOS	34
USO DE BIBLIOTECAS Y FRAMEWORKS (SI APLICABLE)	34
16. 45	
CONFIGURACIÓN DE CONEXIONES AL BACKEND	34
OBTENCIÓN Y PRESENTACIÓN DE DATOS	34
ACTUALIZACIÓN EN TIEMPO REAL (SI APLICABLE)	34

17.	45	
		MANEJO DE FORMULARIOS Y VALIDACIÓN DE DATOS 35
		IMPLEMENTACIÓN DE FUNCIONALIDADES INTERACTIVAS 35
		MEJORAS EN LA EXPERIENCIA DEL USUARIO 35
18.	46	
		DISEÑO DE CASOS DE PRUEBA DE FRONTEND 35
		PRUEBAS DE USABILIDAD 35
		DEPURACIÓN DE ERRORES Y OPTIMIZACIÓN DEL CÓDIGO 35
19.	46	
		MIGRACIÓN DE LA LÓGICA DE NEGOCIO DESDE EL BACKEND (SI NECESARIO) 36
		VALIDACIÓN DE DATOS Y REGLAS DE NEGOCIO EN EL FRONTEND 36
20.	47	
		VERIFICACIÓN DE LA COMUNICACIÓN EFECTIVA CON EL BACKEND 36
		PRUEBAS DE INTEGRACIÓN FRONTEND-BACKEND 36
		ANEXOS 36

Etapa 1 Diseño de la Aplicación y Análisis de Requisitos

1. Introducción

Propósito del Documento

El propósito de este documento es recopilar de manera exhaustiva la información relacionada con la creación del componente de mensajería que se integrará en la plataforma plataforma de contenido digital. Esta documentación se enfoca en detallar las distintas fases necesarias para la obtención de un producto final, haciendo uso de la metodología de Design Thinking estructurada en 6 módulos, con el objetivo de clarificar y organizar las ideas de manera efectiva.

Alcance del Proyecto componente de mensajería

El proyecto tiene como objetivo principal desarrollar un componente de mensajería dentro de la plataforma plataforma de contenido digital, que permitirá a los usuarios interactuar de manera efectiva y eficiente con un chat-bot de respuestas rápidas, el soporte técnico (creador de la página) y otros usuarios. La implementación del componente se llevará a cabo en tres fases durante tres semestres académicos. En el primer semestre, se está enfocando en la documentación detallada de los componentes y sus funcionalidades.

Funcionalidades del componente:

1. Chat Bot de Respuestas Rápidas:

Los usuarios podrán interactuar con un chat-bot que proporcionará respuestas rápidas a preguntas frecuentes y consultas comunes.

2. Soporte Técnico:

Los usuarios podrán comunicarse con el soporte técnico, que es el creador de la página, para resolver problemas técnicos, recibir asistencia y hacer consultas específicas relacionadas con la plataforma plataforma de contenido digital.

3. Comunicación entre Usuarios:

Los usuarios podrán enviar mensajes entre ellos, facilitando la interacción y la colaboración en la plataforma.

Visión Futura:

- El componente de mensajería proporcionará una base sólida para futuras mejoras y expansiones en plataforma de contenido digital. Se podrían considerar las siguientes mejoras a futuro:
- Integración de Multimedia: Permitir a los usuarios enviar y recibir imágenes, vídeos y otros archivos multimedia a través del componente de mensajería.
- Funcionalidades Avanzadas del Chat Bot: Mejorar la inteligencia artificial del chat bot para comprender consultas más complejas y proporcionar respuestas más detalladas.
- Interacción con Otras Plataformas: Integrar el componente de mensajería con otras plataformas y servicios externos para una experiencia de usuario más completa.

Este alcance del proyecto proporciona una visión clara de las funcionalidades a implementar y ver el enfoque por fases para su desarrollo exitoso en el transcurso de tres semestres.

4. **plataforma de contenido digital:** Nombre de la plataforma para la cual se está desarrollando el componente de mensajería. Se refiere al entorno digital donde los usuarios interactúan y colaboran.
5. **Chat-Bot:** Programa informático diseñado para simular una conversación con usuarios, especialmente a través de Internet. En este contexto, se refiere al sistema automatizado que proporciona respuestas rápidas a las preguntas de los usuarios.
6. **Soporte Técnico:** Equipo o individuo responsable de brindar asistencia técnica a los usuarios de la plataforma plataforma de contenido digital. Se refiere al creador de la página o al equipo dedicado a resolver problemas y consultas técnicas.
7. **Componente de Mensajería:** Módulo o funcionalidad específica dentro de plataforma de contenido digital que permite a los usuarios enviar, recibir y gestionar mensajes de texto y otros tipos de contenido a través del chat-bot, el soporte técnico y otros usuarios.

2. Descripción General

Objetivos del Sistema

Chat-Bot

1. Interactividad y Engage:

Generar respuestas interactivas y atractivas para los usuarios, manteniendo su interés y participación en la conversación.

2. Comprensión y Respuesta Precisa:

Mejorar la capacidad de comprensión del chat-bot para proporcionar respuestas precisas y relevantes a las preguntas y consultas de los usuarios.

3. Personalización:

Personalizar las respuestas del chat-bot según el historial y preferencias del usuario para brindar una experiencia más individualizada.

Chat Usuario

Experiencia del Usuario:

Garantizar una interfaz de usuario intuitiva y fácil de usar que permita a los usuarios comunicarse de manera efectiva y cómoda.

Seguridad y Privacidad:

Garantizar la seguridad y privacidad de las conversaciones de los usuarios, implementando medidas adecuadas de encriptación y protección de datos.

Chat de Soporte:

Eficiencia en la Respuesta:

Proporcionar respuestas rápidas y precisas a las consultas y problemas de los usuarios para mejorar la satisfacción del cliente.

Resolución de Problemas:

Garantizar que el chat de soporte tenga la capacidad de resolver problemas y consultas de los usuarios de manera efectiva y un tiempo hábil de 3 días.

Reducción de Cargas de Trabajo:

Automatizar tareas rutinarias para reducir la carga de trabajo del personal de soporte y permitirles enfocarse en problemas más complejos y estratégicos.

Recopilación de Datos y Retroalimentación:

Recopilar datos y comentarios de los usuarios para identificar áreas de mejora en el servicio de soporte y optimizar la calidad de las respuestas.

Funcionalidad General

La funcionalidad general de un chat, en el contexto de las aplicaciones de mensajería y comunicación en línea, incluye una serie de características que permiten a los usuarios interactuar y comunicarse entre sí. Algunas de las funcionalidades típicas de un chat son las siguientes:

- Capacidad de enviar y recibir mensajes en tiempo real.
- Capacidad de reconocer y procesar las intenciones del usuario.
- Capacidad de responder automáticamente a las preguntas frecuentes.
- Capacidad para hacer preguntas al chat-bot
- Chat-bot redirige al usuario para enviar un mensaje a soporte
- Capacidad para hacer pregunta al chat soporte

Usuarios del Sistema

Los siguientes usuarios pueden interactuar con el chat dependiendo de las funcionalidades.

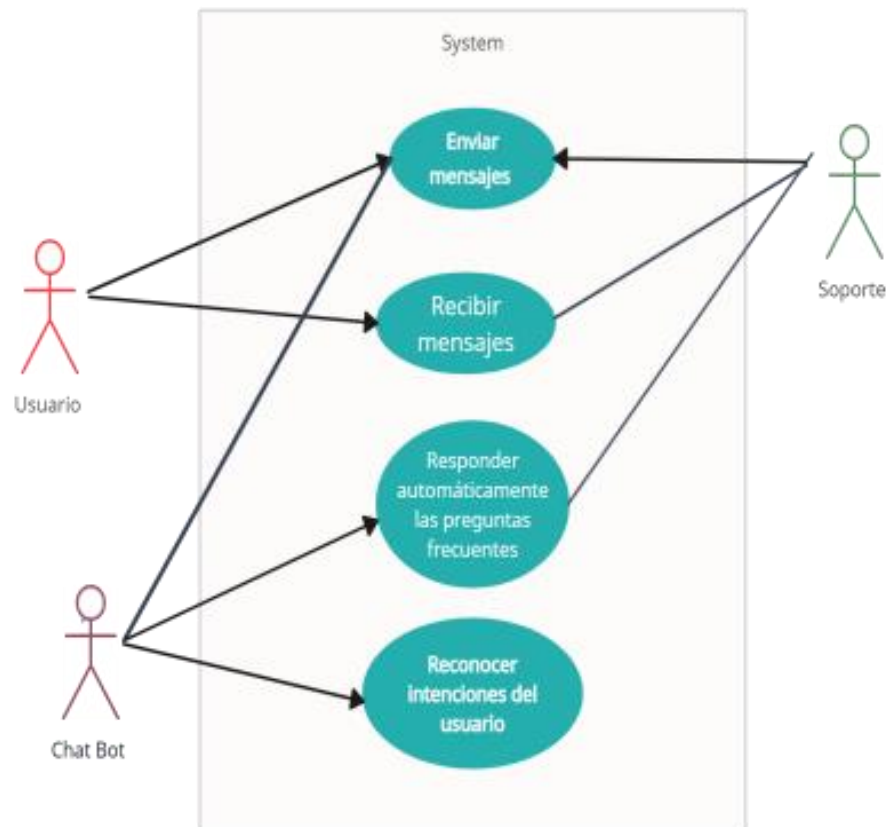
Funcionalidad	Sistema	Usuario	Chat-bot	Soporte
Enviar mensajes	✓	✓	✓	✓
Recibir mensajes	✓	✓	✓	✓
Responder automáticamente a las preguntas frecuentes	✓		✓	
Reconocer intención del usuario	✓		✓	

3. Requisitos Funcionales

Casos de Uso

1. Enviar mensajes
2. Recibir mensajes
3. Responder automáticamente las preguntas frecuentes
4. Reconocer intenciones del usuario

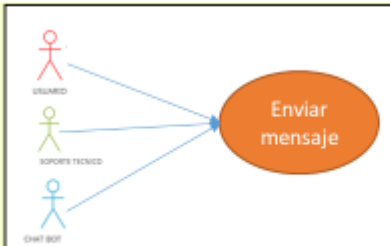
Diagrama de caso de uso

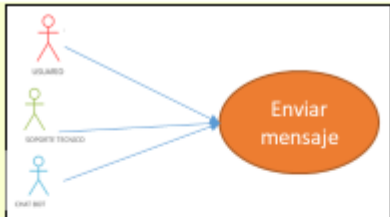


<https://app.creately.com/d/start/dashboard>

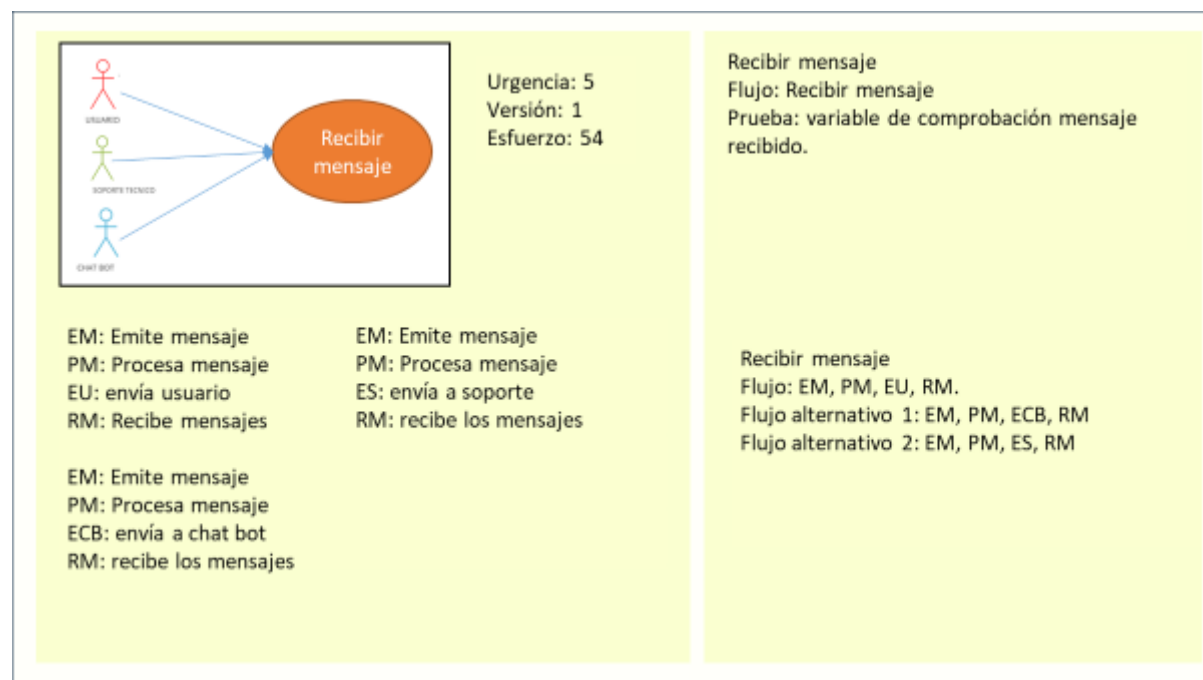
Diagramas de Flujo de Casos de Uso

CU-1

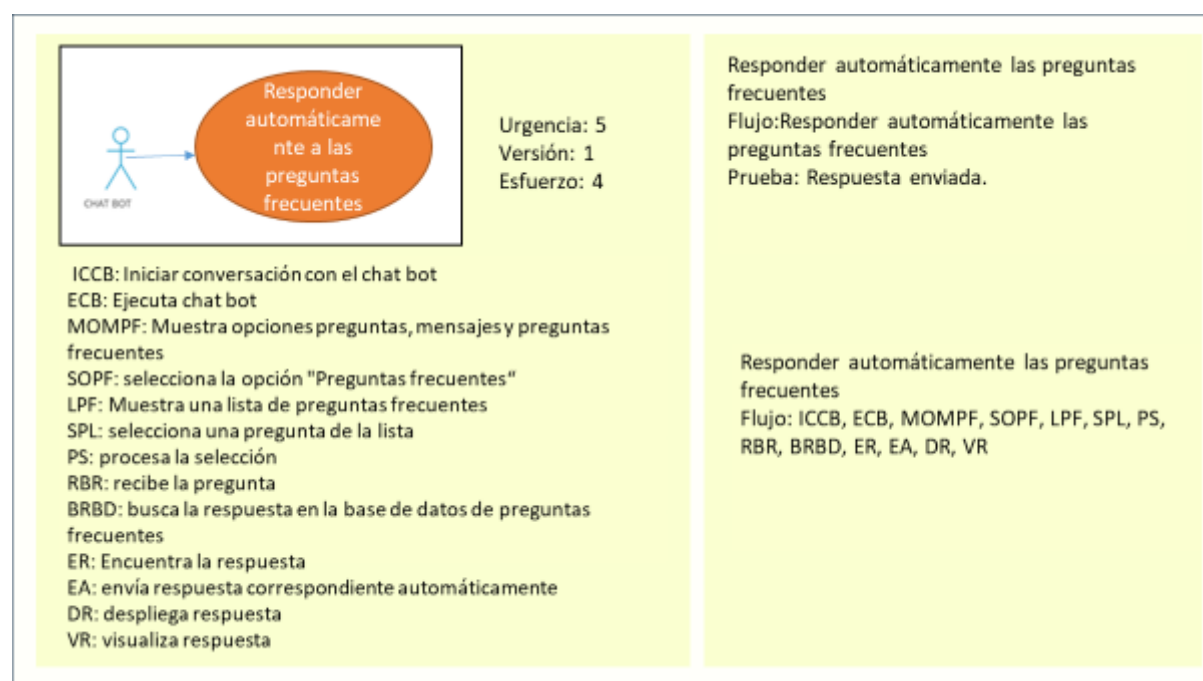
	<p>Urgencia: 5 Versión: 1 Esfuerzo: 4</p>	<p>Enviar mensaje Flujo: enviar mensaje Prueba: variable de comprobación mensaje enviado.</p>
<p>IC: Iniciar conversación MOEM: Muestra la opción de "Escribir mensaje" SOEM: Selecciona la opción "Escribir mensaje" EM: Escribe su mensaje SEM: enviar mensaje ECB: se envía a chat bot PM: procesa el mensaje GR: genera respuesta RRS: retorna respuesta al sistema RRU: retorna respuesta al usuario CC: cierra conversación</p>	<p>Enviar mensaje Flujo: IC, MOEM, SOEM, EM, SEM, ECB, PM, GR, RRS, RRU, CC.</p>	

	<p>Urgencia: 5 Versión: 1 Esfuerzo: 4</p>	<p>Enviar mensaje Flujo: enviar mensaje Prueba: variable de comprobación mensaje enviado.</p>
<p>IC: Iniciar conversación MOEM: Muestra la opción de "Escribir mensaje" SOEM: Selecciona la opción "Escribir mensaje" EM: Escribe su mensaje SEM: enviar mensaje ECB: se envía a chat bot PM: procesa mensaje NR: no puede responder mensaje RS: retorna mensaje al sistema ES: se envía a soporte PM: procesa mensaje GR: genera respuesta RS: retorna respuesta al sistema RU: retorna respuesta al usuario CC: cierra conversación</p>	<p>Enviar mensaje Flujo alternativo: IC, MOEM, SOEM, EM, SEM, ECB, PM, NR, RS, ES, PM, GR, RS, RU, CC.</p>	

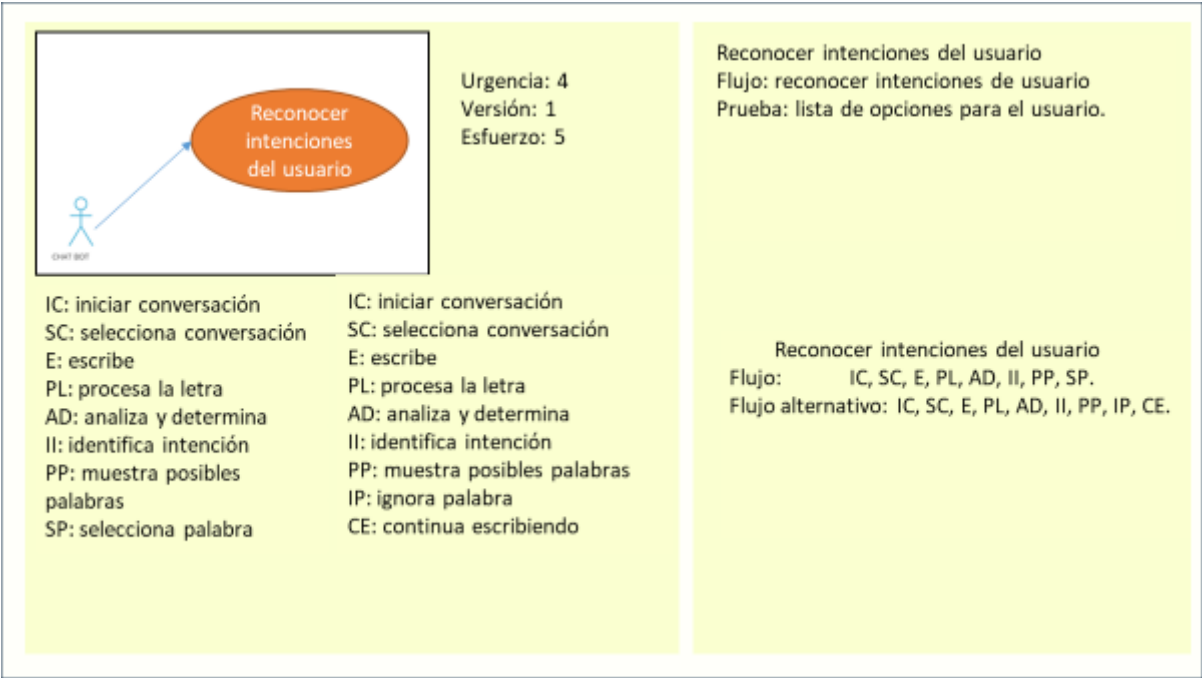
CU-2



CU-3



CU-4



CASO No. 1 Enviar mensajes

ID:	CU-1			
Nombre	Enviar mensajes			
Actores	Usuario, chat-bot, soporte			
Objetivo	Permitir enviar mensajes			
Urgencia	5			
Esfuerzo	4			
Pre- condiciones	Estar autenticado			
Flujo Normal	USUARIO	SISTEMA	CHAT-BOT	SOPORTE
	Iniciar conversación			
		Muestra la opción de "Escribir mensaje"		
	Selecciona la opción "Escribir mensaje"			
	Escribe su mensaje			
	Enviar mensaje			

		Se envía a chat-bot		
			Procesa el mensaje	
			Genera respuesta	
			Retorna respuesta al sistema	
		Retorna respuesta al usuario		
	Cierra conversación			
Flujo Alternativo 1	Iniciar conversación			
		Muestra la opción de "Escribir mensaje"		
	Selecciona la opción "Escribir mensaje"			
	Escribe su mensaje			
	Enviar mensaje			
		Se envía a chat-bot		

			Procesa el mensaje	
			No puede responder al mensaje	
			Retorna mensaje al sistema	
		Se envía a soporte		
				Procesa el mensaje
				Genera respuesta
				Retorna respuesta al sistema
		Retorna respuesta al usuario		
	Cierra conversación			

CASO No. 2 Recibir mensajes

ID:	CU-2
Nombre	Recibir mensajes
Actores	Usuario, chat-bot, soporte

Objetivo	Permitir recibir mensajes			
Urgencia	4			
Esfuerzo	4			
Pre- condiciones	Estar autenticado			
Flujo Normal	USUARIO	SISTEMA	CHAT-BOT	SOPORTE
	Emite mensajes			
		Procesa los mensajes		
		Envía a usuario		
	Recibe mensaje			
Flujo Alternativo 1	Emite mensajes			
		Procesa los mensajes		
		Envía a Chat-bot		
			Recibe los mensajes	
	Emite mensajes			

Flujo Alternativo 2		Procesa los mensajes		
		Envía a Soporte		
				Recibe los mensajes

CASO No. 3 Responder automáticamente las preguntas frecuentes

ID:	CU-4			
Nombre	Responder automáticamente las preguntas frecuentes			
Actores	Chat-bot			
Objetivo	Permitir responder preguntas frecuentes			
Urgencia	5			
Esfuerzo	4			
Pre-condiciones	Estar autenticado			
Flujo Normal	USUARIO	SISTEMA	CHAT-BOT	SOPORTE
	Iniciar una conversación con el chat-bot			
		Ejecuta al Chat-bot		
			Muestra opciones para	

			enviar mensajes y preguntas frecuentes	
	Selecciona la opción "Preguntas frecuentes"			
			Muestra una lista de preguntas frecuentes	
	Selecciona una pregunta de la lista			
		Procesa la selección		
			Recibe la pregunta	
			Busca la respuesta correspondiente en la base de datos de preguntas frecuentes	
			Encuentra respuesta	
			Envía la respuesta correspondiente	

			automáticamente	
		Despliega respuesta		
	Visualiza la respuesta			

CASO No. 4 Reconocer intenciones del usuario

ID:	CU-7			
Nombre	Reconocer intenciones del usuario			
Actores	Chat-bot			
Objetivo	Permitir reconocer las intenciones del usuario			
Urgencia	5			
Esfuerzo	5			
Pre-condiciones	Estar autenticado			
	USUARIO	SISTEMA	CHAT-BOT	SOPORTE
	Iniciar conversación			
	Selecciona conversación			
	Escribe			

Flujo Normal		Procesa las letras		
		Envía letras a chat-bot		
			Analiza las letras para determinar su intención	
			Identifica intención	
		Muestra posibles palabras		
	Selecciona palabra			
Flujo alternativo	Iniciar conversación			
	Selecciona conversación			
	Escribe			
		Procesa las letras		
		Envía letras a chat-bot		
			Analiza las letras para determinar su intención	
			Identifica intención	

		Muestra posibles palabras		
	Ignora palabras			
	Continúa escribiendo			

Prioridad de Requerimientos

A continuación, se presentan algunos requisitos

Obligatorios:

- Capacidad de enviar y recibir mensajes de texto.
- Capacidad de enviar y recibir mensajes multimedia (imágenes, videos, etc.).
- Capacidad de enviar y recibir mensajes en tiempo real.
- Capacidad de enviar y recibir mensajes en grupos.
- Capacidad de reconocer y procesar las intenciones del usuario.
- Capacidad de responder automáticamente a las preguntas frecuentes.

Importantes:

- Capacidad de enviar y recibir mensajes encriptados.
- Capacidad de enviar y recibir mensajes programados.
- Capacidad de enviar y recibir mensajes con confirmación de lectura.
- Capacidad de personalizar la experiencia del usuario.
- Capacidad de integrarse con sistemas de gestión de tickets y soporte.

Deseables:

- Capacidad de enviar y recibir mensajes con traducción automática.
- Capacidad de enviar y recibir mensajes con filtros de spam.
- Capacidad de enviar y recibir mensajes con análisis de sentimiento.
- Capacidad de enviar y recibir mensajes con recomendaciones personalizadas.
- Capacidad de enviar y recibir mensajes con integración de chat-bot de terceros.

A partir del análisis de requerimientos, funcionalidades y el proceso de design thinking, se concreta la siguiente matriz de prioridad de requerimientos.

Para la interpretación se tiene en cuenta la siguiente escala con sus valores.

Eje de Urgencia:

- Obligatoria (5)
- Alta (4)
- Moderada (3)
- Menor (2)
- Baja (1)

Eje de Esfuerzo:

- Muy alto (5)
- Alto (4)
- Medio (3)
- Bajo (2)
- Muy bajo (1)

	Urgencia					
Im pa cto		1-Baja	2-Menor	3- Moderada	4-Alta	5- Obligatoria
	5-Muy alto	5	10	15	20	25
					CU-1 CU-2	
	4-Alto	4	8	12	16	20
				CU-4		
	3-Medio	3	6	9	12	15
			CU-3			
	2-Bajo	2	4	6	8	10
	1-Muy bajo	1	2	3	4	5

4. Requisitos No Funcionales

Requisitos de Desempeño

Rendimiento de chat-bot: Este debe proporcionar un rendimiento de preguntas rápidas en tiempo real la cual deben de reflejarse con el usuario para resolver múltiples problemas asignadas y calificadas como las que siempre preguntan simultáneamente por pasos para devolverse por caso de que la persona cometa un error.

Rendimiento de chat (usuario con contactos): en esta parte se cargarán rápido los mensajes de los usuarios para que no tengan inconvenientes al momento de chatear, en caso de fallar plataforma de contenido digital cuenta la opción de chat-bot y chat al soporte en casos mayores.

Rendimiento de chat de soporte: Este debe proporcionar en tiempo real la ayuda brindada en caso de que no encuentren posibles soluciones con el chat-bot, en esta opción se comunicara un cliente que es llamado usuario con una persona administradora de la página plataforma de contenido digital con conocimiento avanzado en el tema para brindarle la ayuda requerida siempre y cuando este en los parámetros de la página.

Requisitos de seguridad:

- **Acceso seguro:** Se debe implementar una autenticación segura para garantizar que solo usuarios autorizados tengan acceso a el chat. Esto puede incluir autenticación de 3 factores, inicio de sesión único, autenticación de seguridad con extremo a bot, y Autenticación de seguridad con extremo y soporte.
- **Alojamiento:** La interfaz del chat-bot es un aspecto a tener en cuenta. Si el bot se utiliza a través de una plataforma de chat como plataforma de contenido digital, la información del cliente quedará expuesta y registrada por esta plataforma. Esto puede o no ser un problema de seguridad del chat-bot, pero es evidente que hay que tenerlo en cuenta.
- **Registro de actividades:** El sistema debe mantener registros de actividades, lo que incluye registros de cambios en la plataforma más que todo el chat, acceso de usuarios y eventos relevantes para la seguridad.
- **Políticas y procedimientos:** Por supuesto, es fundamental para todo lo anterior que se establezcan políticas y procedimientos pertinentes que rijan las normas de seguridad de la información. La seguridad de la información no se establece una sola vez, sino que es una actividad continua. Estas políticas y procedimientos regirán no sólo la configuración del software pertinente, sino que también especificarán cuándo y cómo se llevarán a

cabo las sesiones periódicas de formación y las pruebas de seguridad. Proteger un chat-bot no difiere de proteger cualquier otro programa informático. Al principio del proceso debe evaluarse el grado de confidencialidad de los datos subyacentes, lo que determinará las medidas que debe adoptar la organización para garantizar la seguridad de los datos.

Requisitos de Seguridad

Acceso seguro: Se debe implementar una autenticación segura para garantizar que solo usuarios autorizados tengan acceso a el chat. Esto puede incluir autenticación de 3 factores, inicio de sesión único, autenticación de seguridad con extremo a bot, y Autenticación de seguridad con Extremo y soporte

Alojamiento: La interfaz del chat-bot es un aspecto a tener en cuenta. Si el bot se utiliza a través de una plataforma de chat como plataforma de contenido digital, la información del cliente quedará expuesta y registrada por esta plataforma. Esto puede o no ser un problema de seguridad del chat-bot, pero es evidente que hay que tenerlo en cuenta.

Registro de actividades: El sistema debe mantener registros de actividades, lo que incluye registros de cambios en la plataforma más que todo el chat, acceso de usuarios y eventos relevantes para la seguridad.

Políticas y procedimientos: Por supuesto, es fundamental para todo lo anterior que se establezcan políticas y procedimientos pertinentes que rijan las normas de seguridad de la información. La seguridad de la información no se establece una sola vez, sino que es una actividad continua. Estas políticas y procedimientos regirán no sólo la configuración del software pertinente, sino que también especificarán cuándo y cómo se llevarán a cabo las sesiones periódicas de formación y las pruebas de seguridad. Proteger un chatbot no difiere de proteger cualquier otro programa informático. Al principio del proceso debe evaluarse el grado de confidencialidad de los datos subyacentes, lo que determinará las medidas que debe adoptar la organización para garantizar la seguridad de los datos.

Requisitos de Usabilidad

Facilidad de Ingreso y Registro:

- Los usuarios deben poder ingresar al chat de manera sencilla y rápida, sin obstáculos innecesarios.
- Proporciona opciones de registro simples, como un correo electrónico.

Navegación Intuitiva:

- El chat debe tener una interfaz de usuario fácil de usar y una navegación clara.
- Los usuarios deben poder alternar de manera intuitiva entre conversaciones con otros, Chatbot y chat de soporte.

Interfaz Conversacional:

- El diseño del chat debe imitar una conversación real para que los usuarios se sientan cómodos.
- Utiliza avatares de quién está hablando para que las conversaciones sean fáciles de seguir.

Retroalimentación Clara: El sistema debe proporcionar retroalimentación clara, como indicadores de escritura, confirmaciones de envío y notificaciones de mensajes no leídos.

integración de Chat-bot: El chat- bot debe ser capaz de comprender las consultas de los usuarios y proporcionar respuestas útiles y precisas.

Privacidad y Seguridad: Garantiza la privacidad de las conversaciones y proporciona opciones de seguridad.

Atención al Cliente Efectiva: El chat de soporte debe ser accesible y brindar respuestas rápidas y útiles a las preguntas y problemas de los usuarios.

Tiempo de Respuesta Rápida: Tanto los usuarios como los chat-bot y el chat de soporte deben responder de manera eficiente para mantener las conversaciones fluidas.

Requisitos de Escalabilidad

Gestión de Conversaciones Múltiples: El sistema debe ser capaz de gestionar múltiples conversaciones simultáneas entre usuarios y Chat-bot, así como entre usuarios y agentes de soporte. Esto implica mantener un estado eficiente para cada conversación y asegurarse de que las interacciones no se mezclen.

Distribución de Cargas Equitativas: Implementar un mecanismo de balanceo de carga que distribuya de manera equitativa las solicitudes de los usuarios entre los servidores o instancias del Chat-bot y los agentes de soporte.

Notificaciones: El sistema debe ser capaz de manejar notificaciones para los agentes de soporte, lo que significa que los agentes deben recibir alertas cuando un usuario solicita asistencia.

Integración de IA y Aprendizaje Automático: Si se utiliza IA o aprendizaje automático en el Chat-bot, asegúrate de que el sistema pueda escalar para manejar el entrenamiento y la inferencia de modelos a medida que la cantidad de datos y usuarios aumente.

5. Modelado E/R

A partir de la abstracción de los datos de las funcionalidades los datos preliminares recolectados son:

CU-1 Enviar mensaje	Variables	Datos
	Mensaje Enviado	Contenido: Hola, necesito ayuda para encontrar los contenidos.
	Verificación de enviado	True
	Verificación de recibido	True
	Hora de enviado	8:23 AM
	Fecha de enviado	15/03/2024

CU-2 Recibir mensaje		
	Mensaje recibido	Contenido: Hola, necesito ayuda para encontrar los contenidos.
	Hora de recibido	8:25 AM
	Fecha de recibido	15/03/2024

CU-3 Responder automáticamente las preguntas frecuentes		
	Mensaje de usuario	¿Dónde puedo encontrar los contenidos?
	Preguntas Frecuentes	Lista de preguntas. ¿Cómo buscar contenidos?

CU-4 Reconocer intenciones del usuario		
	texto asociado	Hola. Necesito ayuda para resolver un problema podrias contactarme con soporte
	funcion	darSoporteTecnico(){}
	Intención	Soporte Tecnico

Diagrama de Entidad-Relación

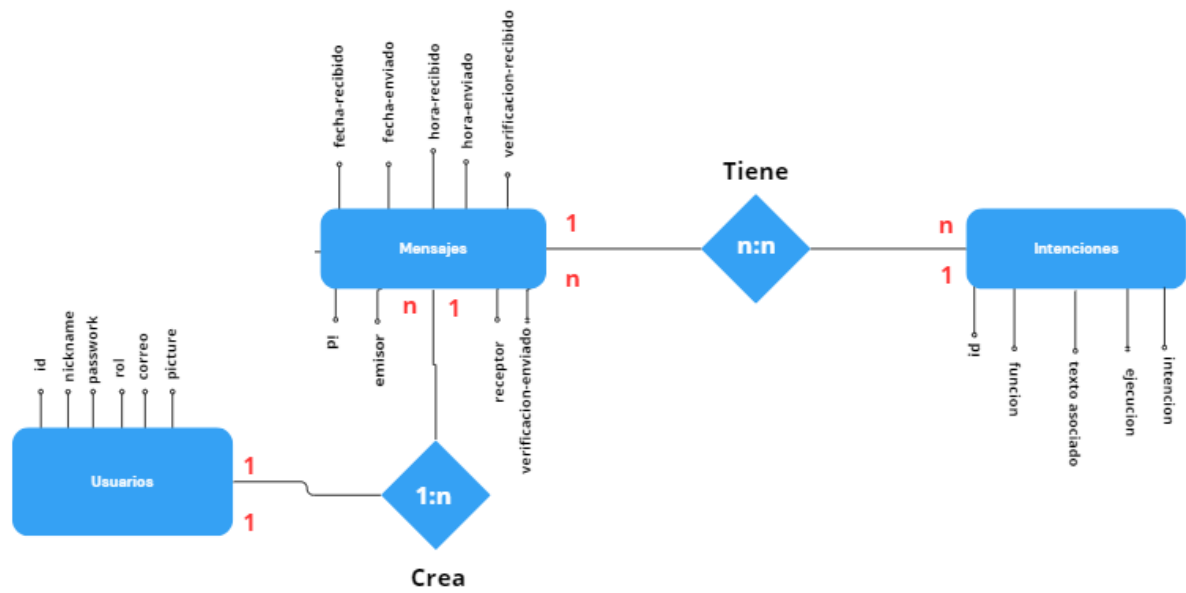
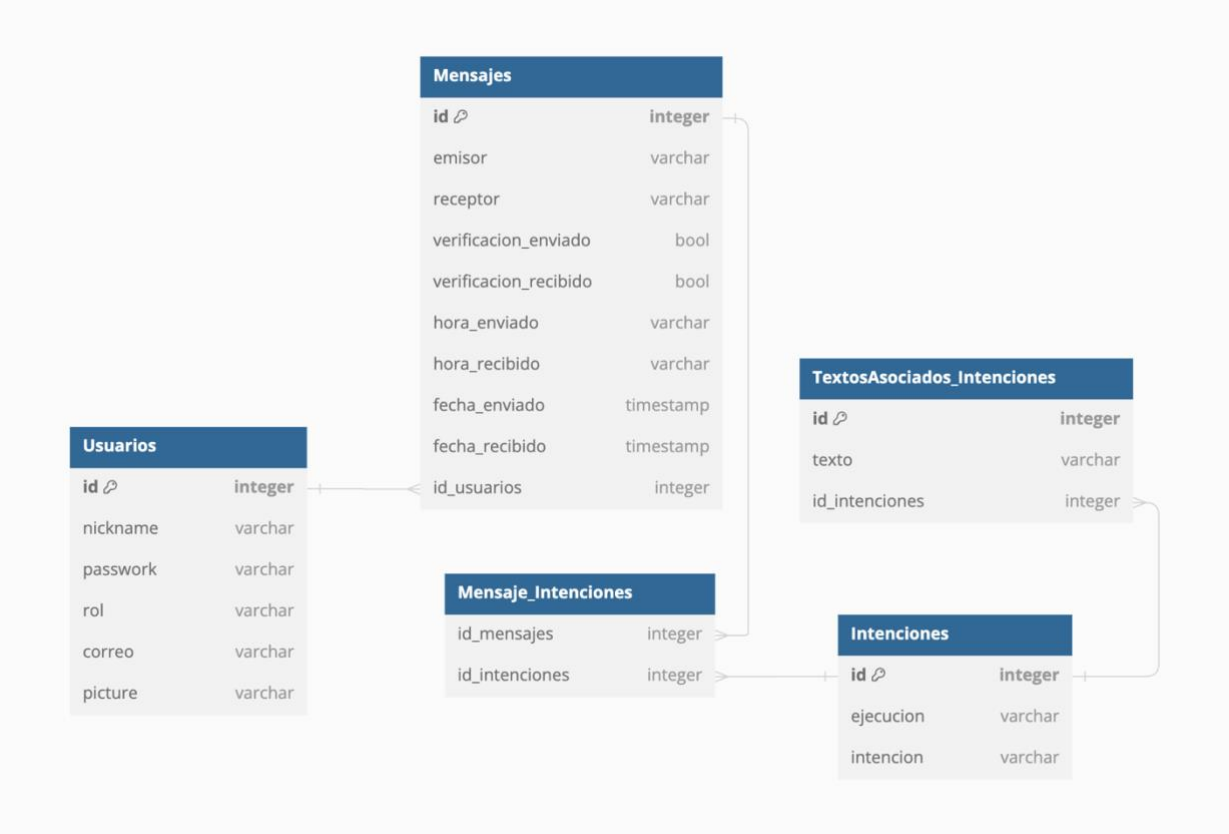


Diagrama Relacional



Script de modelo relacional

<https://dbdiagram.io/>

// Use DBML to define your database structure

// Docs: <https://dbml.dbdiagram.io/docs>

Table Usuarios {

id integer [primary key]

nickname varchar

password varchar

rol varchar

correo varchar

picture varchar

}

Table Mensajes {

id integer [primary key]

emisor varchar

receptor varchar

verificacion_enviado bool

verificacion_recibido bool

hora_enviado varchar

hora_recibido varchar

fecha_enviado timestamp

fecha_recibido timestamp

id_usuarios integer [ref: >Usuarios.id]

}

Table Intenciones {

id integer [primary key]

ejecucion varchar

intencion varchar

}

Table Mensaje_Intenciones {

id_mensajes integer [ref: > Mensajes.id]

```
id_intenciones integer [ref: > Intenciones.id]
}
```

```
Table TextosAsociados_Intenciones {
  id integer [primary key]
  texto varchar
  id_intenciones integer [ref: > Intenciones.id]
}
```

Descripción de Entidades y Relaciones

Entidades:

1. User (Usuario):

Almacena información sobre los usuarios que pueden acceder a la pizarra.

Atributos: ID (identificador único), nombre de usuario, rol (como administrador, docente, estudiante), fecha de creación.

Relaciones: Cada usuario puede estar asociado con varias pizarras a través de la entidad "WhiteboardMember."

2. Whiteboard (Pizarra):

Representa una pizarra en la aplicación de pizarra compartida.

Atributos: ID (identificador único), título de la pizarra, descripción, fecha de creación.

Relaciones: Cada pizarra puede contener contenido a través de la relación con la entidad "Content" y puede tener plugins asociados a través de la relación con la entidad "Plugin."

3. Content (Contenido):

Almacena contenido que se puede agregar a las pizarras, como texto, imágenes, videos, documentos, etc.

Atributos: ID (identificador único), tipo de contenido, contenido en sí, fecha de creación.

Relaciones: El contenido se asocia con un usuario a través de la relación con la entidad "User" y puede estar relacionado con comentarios.

4. Comment (Comentario):

Almacena comentarios realizados por los usuarios en relación con el contenido de la pizarra.

Atributos: ID (identificador único), texto del comentario, fecha de creación.

Relaciones: Cada comentario se relaciona con el contenido específico en la entidad "Content."

5. RevisionHistory (Historial de Revisiones):

Registra el historial de revisiones y cambios realizados en las pizarras.

Atributos: ID (identificador único), fecha de la revisión.

Relaciones: Cada entrada de historial se relaciona con una pizarra específica en la entidad "Whiteboard."

6. MultimediaResource (Recurso Multimedia):

Almacena recursos multimedia, como imágenes, videos, documentos, etc.

Atributos: ID (identificador único), tipo de recurso, ubicación o URL del recurso, título, descripción, fecha de carga.

Relaciones: Cada entrada de Recurso multimedia se relaciona con un contenido específico en la entidad "Content."

7. Plugin:

Almacena información sobre los plugins que pueden proporcionar funcionalidades personalizadas en las pizarras.

Atributos: ID (identificador único), nombre del plugin, descripción, autor, versión, configuración.

Relaciones: Cada pizarra puede tener asociado uno o varios plugins a través de la entidad "Whiteboard."

Relaciones:

- "User" se relaciona con "WhiteboardMember" para indicar la asociación de los usuarios con las pizarras.
- "Content" se relaciona con "User" para registrar los usuarios que pueden interactuar con el contenido.
- "Content" se relaciona con "Comment" para permitir comentarios en los contenidos.
- "Whiteboard" se relaciona con "Content" para indicar que una pizarra puede contener contenido.
- "Whiteboard" se relaciona con "Plugin" para permitir la asociación de plugins con las pizarras.
- "WhiteboardMember" se relaciona con "User" y "Whiteboard" para indicar la asociación de usuarios con pizarras y sus roles.
- "RevisionHistory" se relaciona con "Whiteboard" para registrar revisiones en las pizarras.
- "MultimediaResource" se relaciona con "Content" para registrar recursos asociados en el contenido.

Reglas de Integridad Referencial

1. **Integridad Referencial entre "User_Content" y "Users":** Cada registro en la tabla "User_Content" debe estar asociado con un usuario existente en la tabla "Users" a través de la clave foránea "user_id."
2. **Integridad Referencial entre "User_Content" y "Contents":** Cada registro en la tabla "User_Content" debe estar asociado con un contenido existente en la tabla "Contents" a través de la clave foránea "content_id."
3. **Integridad Referencial entre "Content_Comment" y "Contents":** Cada registro en la tabla "Content_Comment" debe estar asociado con un contenido existente en la tabla "Contents" a través de la clave foránea "content_id."

4. **Integridad Referencial entre "Content_Comment" y "Comments"**: Cada registro en la tabla "Content_Comment" debe estar asociado con un comentario existente en la tabla "Comments" a través de la clave foránea "comment_id."
5. **Integridad Referencial entre "Whiteboards_WhiteboardMembers" y "Whiteboards"**: Cada registro en la tabla "Whiteboards_WhiteboardMembers" debe estar asociado con una pizarra existente en la tabla "Whiteboards" a través de la clave foránea "Whiteboards_id."
6. **Integridad Referencial entre "Whiteboards_WhiteboardMembers" y "WhiteboardMembers"**: Cada registro en la tabla "Whiteboards_WhiteboardMembers" debe estar asociado con un miembro de la pizarra existente en la tabla "WhiteboardMembers" a través de la clave foránea "whiteboardMembers_id."
7. **Integridad Referencial entre "Whiteboard_Plugin" y "Whiteboards"**: Cada registro en la tabla "Whiteboard_Plugin" debe estar asociado con una pizarra existente en la tabla "Whiteboards" a través de la clave foránea "whiteboard_id."
8. **Integridad Referencial entre "Whiteboard_Plugin" y "Plugins"**: Cada registro en la tabla "Whiteboard_Plugin" debe estar asociado con un plugin existente en la tabla "Plugins" a través de la clave foránea "plugin_id."
9. **Integridad Referencial entre "Contents_Whiteboards" y "Contents"**: Cada registro en la tabla "Contents_Whiteboards" debe estar asociado con un contenido existente en la tabla "Contents" a través de la clave foránea "Contents_id."
10. **Integridad Referencial entre "Contents_Whiteboards" y "Whiteboards"**: Cada registro en la tabla "Contents_Whiteboards" debe estar asociado con una pizarra existente en la tabla "Whiteboards" a través de la clave foránea "whiteboard_id."
11. **Integridad Referencial entre "Content_MultimediaResource" y "Contents"**: Cada registro en la tabla "Content_MultimediaResource" debe estar asociado con un contenido existente en la tabla "Contents" a través de la clave foránea "content_id."
12. **Integridad Referencial entre "Content_MultimediaResource" y "MultimediaResources"**: Cada registro en la tabla "Content_MultimediaResource" debe estar asociado con un recurso multimedia existente en la tabla "MultimediaResources" a través de la clave foránea "MultimediaResources_id."
13. **Integridad Referencial entre "RevisionHistorys" y "Whiteboards"**: Cada registro en la tabla "RevisionHistorys" debe estar asociado con una pizarra existente en la tabla "Whiteboards" a través de la clave foránea "whiteboard_id."
14. **Integridad Referencial entre "WhiteboardMembers" y "Users"**: Cada registro en la tabla "WhiteboardMembers" debe estar asociado con un usuario existente en la tabla "Users" a través de la clave foránea "user_id."

Colecciones (NoSQL)

Mensajes

```
{
  Id: String,
  Emisor: String,
  Receptor: String,
  verificacion_enviado: boolean,
  Verificacion_recibido: boolean,
  Hora_enviado: Date,
  Hora_recibido: Date,
  Fecha_enviado: Date,
  Fecha_recibido: Date,
  id_usuario: Object.Id,
  Intenciones:[Object.Id]
}
```

```
Intenciones: {
  Id: Object: Id,
  Intencion: String,
  Mensajes: [Object.Id]
}
```

```
TextoAsociado_Intenciones:{
  Id:Object.Id,
  Texto: String,
  Id.Intencion:Object.Id
}
```

6. Anexos

Diagramas Adicionales

Referencias

Etapa 2: Persistencia de Datos con Backend

7. Introducción

Propósito de la Etapa

Alcance de la Etapa

Definiciones y Acrónimos

8. Diseño de la Arquitectura de Backend

Descripción de la Arquitectura Propuesta

Componentes del Backend

Diagramas de Arquitectura

9. Elección de la Base de Datos

Evaluación de Opciones (SQL o NoSQL)

Justificación de la Elección

Diseño de Esquema de Base de Datos

10. Implementación del Backend

Elección del Lenguaje de Programación

Creación de la Lógica de Negocio

Desarrollo de Endpoints y APIs

Autenticación y Autorización

11. Conexión a la Base de Datos

Configuración de la Conexión

Desarrollo de Operaciones CRUD

Manejo de Transacciones

12. Pruebas del Backend

Diseño de Casos de Prueba

Ejecución de Pruebas Unitarias y de Integración

Manejo de Errores y Excepciones

Etapa 3: Consumo de Datos y Desarrollo Frontend

13. Introducción

Propósito de la Etapa

Alcance de la Etapa

Definiciones y Acrónimos

14. Creación de la Interfaz de Usuario (UI)

Diseño de la Interfaz de Usuario (UI) con HTML y CSS

Consideraciones de Usabilidad

Maquetación Responsiva

15. Programación Frontend con JavaScript (JS)

Desarrollo de la Lógica del Frontend

Manejo de Eventos y Comportamientos Dinámicos

Uso de Bibliotecas y Frameworks (si aplicable)

16. Consumo de Datos desde el Backend

Configuración de Conexiones al Backend

Obtención y Presentación de Datos

Actualización en Tiempo Real (si aplicable)

17. Interacción Usuario-Interfaz

Manejo de Formularios y Validación de Datos

Implementación de Funcionalidades Interactivas

Mejoras en la Experiencia del Usuario

18. Pruebas y Depuración del Frontend

Diseño de Casos de Prueba de Frontend

Pruebas de Usabilidad

Depuración de Errores y Optimización del Código

19. Implementación de la Lógica de Negocio en el Frontend

Migración de la Lógica de Negocio desde el Backend (si necesario)

Validación de Datos y Reglas de Negocio en el Frontend

20. Integración con el Backend

Verificación de la Comunicación Efectiva con el Backend

Pruebas de Integración Frontend-Backend

ANEXOS

Diagramas UML

- **Diagrama de Casos de Uso (Use Case Diagram):** Este diagrama muestra las interacciones entre los actores (usuarios) y el sistema. Puede ayudar a identificar las funcionalidades clave y los actores involucrados.
- **Diagrama de Secuencia (Sequence Diagram):** Estos diagramas muestran la interacción entre objetos y actores a lo largo del tiempo. Puedes utilizarlos para representar cómo los usuarios interactúan con la pizarra en un flujo de trabajo específico.
- **Diagrama de Clases (Class Diagram):** Puedes utilizar este diagrama para modelar las clases y estructuras de datos subyacentes en el sistema, como usuarios, pizarras, comentarios, revisiones, etc.

- **Diagrama de Estados (State Diagram):** Este diagrama puede ser útil para modelar el comportamiento de la pizarra en diferentes estados, como "edición", "visualización", "comentario", etc.
- **Diagrama de Despliegue (Deployment Diagram):** Puedes utilizar este diagrama para representar cómo se despliega la aplicación en servidores y cómo interactúa con otros componentes del sistema, como el CMS.
- **Diagrama de Componentes (Component Diagram):** Este diagrama puede ayudar a representar la estructura de componentes del software, como la interfaz de usuario, la lógica de negocio, las bibliotecas y los servicios utilizados.
- **Diagrama de Actividad (Activity Diagram):** Puedes usar este diagrama para modelar flujos de trabajo o procesos específicos, como el flujo de trabajo de creación y edición de contenido en la pizarra.
- **Diagrama de Comunicación (Communication Diagram):** Similar a los diagramas de secuencia, estos diagramas muestran interacciones entre objetos y actores, pero pueden ser más simples y enfocados en la comunicación.
- **Diagrama de Paquetes (Package Diagram):** Este diagrama puede ayudar a organizar y visualizar los paquetes y módulos del software, lo que es útil para el diseño modular.
- **Diagrama de Objetos (Object Diagram):** Puedes utilizar este diagrama para representar instancias de clases y cómo interactúan en un escenario específico.