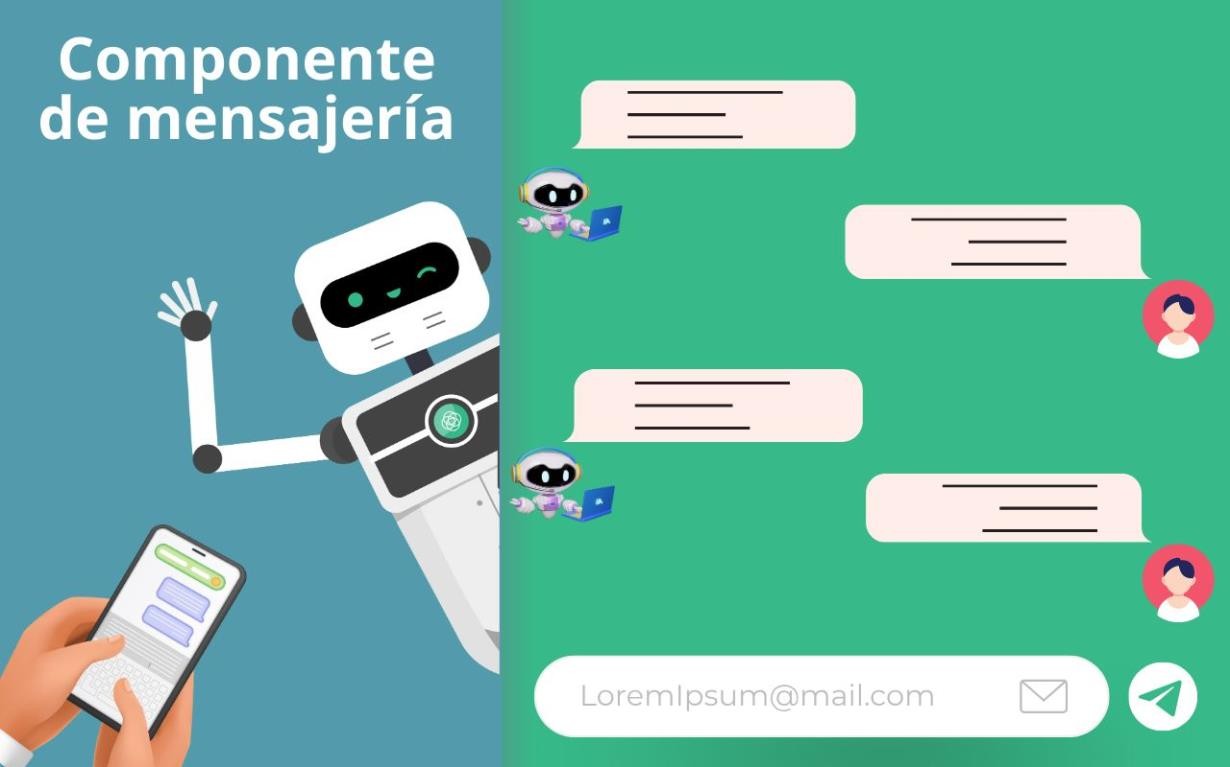
# Documento de Propuesta de Diseño de

Software I, II y II

**Componente de mensajería**

**Autores**: Jaider Martínez Paternina - María Camila Salgado Montiel - Andrés Felipe López León- Sergio Andrés Pineda Ruiz



##### 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](#_bookmark0)

[1.](#_bookmark1) 6

[PROPÓSITO DEL DOCUMENTO 6](#_bookmark2)

[ALCANCE DEL PROYECTO MÓDULO DE PIZARRA COMPARTIDA 8](#_bookmark3)

[DEFINICIONES Y ACRÓNIMOS 8](#_TOC_250000)

[2.](#_bookmark4) 8

[OBJETIVOS DEL SISTEMA 10](#_bookmark5)

[FUNCIONALIDAD GENERAL 10](#_bookmark6)

[USUARIOS DEL SISTEMA 11](#_bookmark7)

RESTRICCIONES 12

[3.](#_bookmark8) 10

[CASOS DE USO 13](#_bookmark9)

[DIAGRAMAS DE FLUJO DE CASOS DE USO 14](#_bookmark10)

[DESCRIPCIÓN DETALLADA DE CADA CASO DE USO 14](#_bookmark11)

[PRIORIDAD DE REQUERIMIENTOS 16](#_bookmark12)

[4.](#_bookmark13) 26

[REQUISITOS DE DESEMPEÑO 18](#_bookmark14)

[REQUISITOS DE SEGURIDAD 19](#_bookmark15)

[REQUISITOS DE USABILIDAD 20](#_bookmark16)

[REQUISITOS DE ESCALABILIDAD 20](#_bookmark17)

[5.](#_bookmark18) 30

[DIAGRAMA DE ENTIDAD-RELACIÓN 21](#_bookmark19)

[DIAGRAMA RELACIONAL 22](#_bookmark20)

[SCRIPT DE MODELO RELACIONAL 23](#_bookmark21)

[DESCRIPCIÓN DE ENTIDADES Y RELACIONES 24](#_bookmark22)

[REGLAS DE INTEGRIDAD REFERENCIAL 25](#_bookmark23)

[COLECCIONES (NOSLQ) 28](#_bookmark24)

[6.](#_bookmark25) 40

[DIAGRAMAS ADICIONALES 29](#_bookmark26)

[REFERENCIAS 29](#_bookmark27)

[ETAPA 2: PERSISTENCIA DE DATOS CON BACKEND 30](#_bookmark28)

[7.](#_bookmark29) 41

[PROPÓSITO DE LA ETAPA 30](#_bookmark30)

[ALCANCE DE LA ETAPA 30](#_bookmark31)

[DEFINICIONES Y ACRÓNIMOS 30](#_bookmark32)

[8.](#_bookmark33) 41

[DESCRIPCIÓN DE LA ARQUITECTURA PROPUESTA 30](#_bookmark34)

[COMPONENTES DEL BACKEND 30](#_bookmark35)

[DIAGRAMAS DE ARQUITECTURA 30](#_bookmark36)

[9.](#_bookmark37) 41

[EVALUACIÓN DE OPCIONES (SQL O NOSQL) 31](#_bookmark38)

[JUSTIFICACIÓN DE LA ELECCIÓN 31](#_bookmark39)

[DISEÑO DE ESQUEMA DE BASE DE DATOS 31](#_bookmark40)

[10.](#_bookmark41) 42

[ELECCIÓN DEL LENGUAJE DE PROGRAMACIÓN 31](#_bookmark42)

[CREACIÓN DE LA LÓGICA DE NEGOCIO 31](#_bookmark43)

[DESARROLLO DE ENDPOINTS Y APIS 31](#_bookmark44)

[AUTENTICACIÓN Y AUTORIZACIÓN 31](#_bookmark45)

[11.](#_bookmark46) 42

[CONFIGURACIÓN DE LA CONEXIÓN 32](#_bookmark47)

[DESARROLLO DE OPERACIONES CRUD 32](#_bookmark48)

[MANEJO DE TRANSACCIONES 32](#_bookmark49)

[12.](#_bookmark50) 43

[DISEÑO DE CASOS DE PRUEBA 32](#_bookmark51)

[EJECUCIÓN DE PRUEBAS UNITARIAS Y DE INTEGRACIÓN 32](#_bookmark52)

[MANEJO DE ERRORES Y EXCEPCIONES 32](#_bookmark53)

[ETAPA 3: CONSUMO DE DATOS Y DESARROLLO FRONTEND 33](#_bookmark54)

[13.](#_bookmark55) 44

[PROPÓSITO DE LA ETAPA 33](#_bookmark56)

[ALCANCE DE LA ETAPA 33](#_bookmark57)

[DEFINICIONES Y ACRÓNIMOS 33](#_bookmark58)

[14.](#_bookmark59) 44

[DISEÑO DE LA INTERFAZ DE USUARIO (UI) CON HTML Y CSS 33](#_bookmark60)

[CONSIDERACIONES DE USABILIDAD 33](#_bookmark61)

[MAQUETACIÓN RESPONSIVA 33](#_bookmark62)

[15.](#_bookmark63) 45

[DESARROLLO DE LA LÓGICA DEL FRONTEND 34](#_bookmark64)

[MANEJO DE EVENTOS Y COMPORTAMIENTOS DINÁMICOS 34](#_bookmark65)

[USO DE BIBLIOTECAS Y FRAMEWORKS (SI APLICABLE) 34](#_bookmark66)

[16.](#_bookmark67) 45

[CONFIGURACIÓN DE CONEXIONES AL BACKEND 34](#_bookmark68)

[OBTENCIÓN Y PRESENTACIÓN DE DATOS 34](#_bookmark69)

[ACTUALIZACIÓN EN TIEMPO REAL (SI APLICABLE) 34](#_bookmark70)

[17.](#_bookmark71) 45

[MANEJO DE FORMULARIOS Y VALIDACIÓN DE DATOS 35](#_bookmark72)

[IMPLEMENTACIÓN DE FUNCIONALIDADES INTERACTIVAS 35](#_bookmark73)

[MEJORAS EN LA EXPERIENCIA DEL USUARIO 35](#_bookmark74)

[18.](#_bookmark75) 46

[DISEÑO DE CASOS DE PRUEBA DE FRONTEND 35](#_bookmark76)

[PRUEBAS DE USABILIDAD 35](#_bookmark77)

[DEPURACIÓN DE ERRORES Y OPTIMIZACIÓN DEL CÓDIGO 35](#_bookmark78)

[19.](#_bookmark79) 46

[MIGRACIÓN DE LA LÓGICA DE NEGOCIO DESDE EL BACKEND (SI NECESARIO) 36](#_bookmark80)

[VALIDACIÓN DE DATOS Y REGLAS DE NEGOCIO EN EL FRONTEND 36](#_bookmark81)

[20.](#_bookmark82) 47

[VERIFICACIÓN DE LA COMUNICACIÓN EFECTIVA CON EL BACKEND 36](#_bookmark83)

[PRUEBAS DE INTEGRACIÓN FRONTEND-BACKEND 36](#_bookmark84)

[ANEXOS 36](#_bookmark85)

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:

###### Chat Bot de Respuestas Rápidas:

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

###### 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.

###### 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.

1. **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.
2. **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.
3. **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.
4. **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.

1. 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.

###### 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.

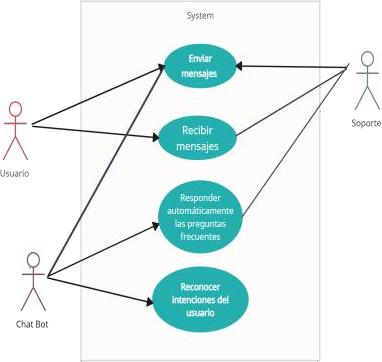
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Funcionalidad** | **Sistema** | **Usuario** | **Chat-bot** | **Soporte** |
| Enviar mensajes | ✓ | ✓ | ✓ | ✓ |
| Recibir mensajes | ✓ | ✓ | ✓ | ✓ |
| Responder automáticamente a las preguntas frecuentes | ✓ |  | ✓ |  |
| Reconocer intención del usuario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

## 3. Requisitos Funcionales

### Casos de Uso

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

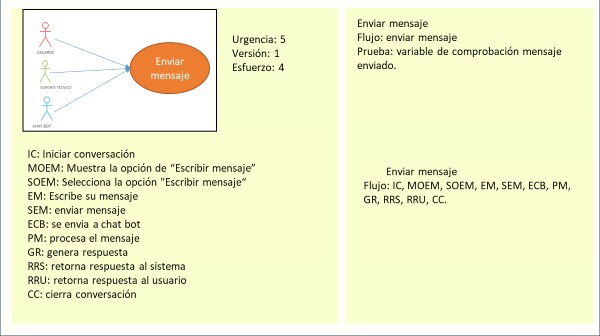
##### Diagrama de caso de uso



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

### Diagramas de Flujo de Casos de Uso

**CU-1**

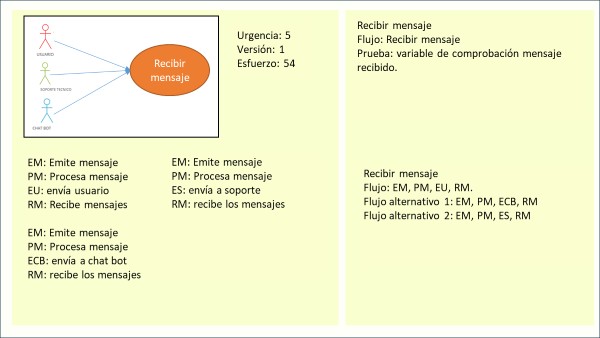


**Urgencia: 4**

**Versión: 1**

**Esfuerzo: 5**

**CU-2**

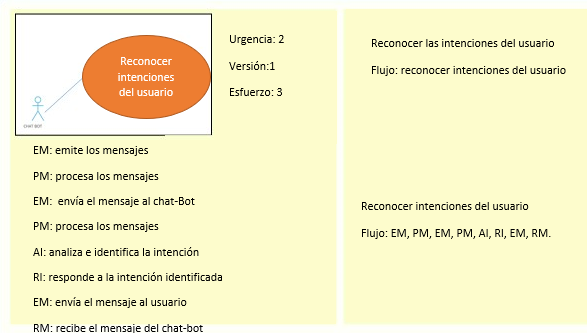


**Urgencia: 4**

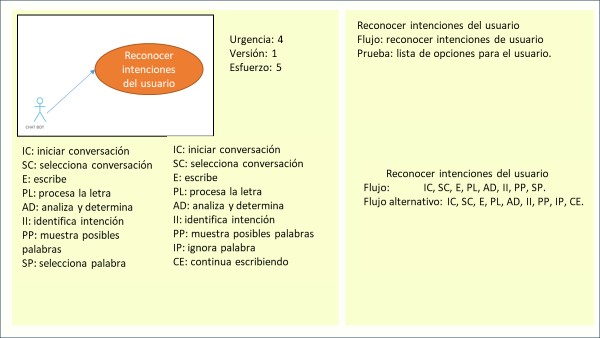
**Versión: 1**

**Esfuerzo: 5**

**CU-3**



**CU-4**



**Textos asociados**

**Textos asociados**

**Textos asociados**

**Flujo: Reconocer la intención en el texto**

**Urgencia: 3**

**Versión: 1**

**Esfuerzo: 4**

Descripción detallada de cada caso de uso

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

**CASO No. 1 Enviar mensajes**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | 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 | 5 | | | |
| 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 Reconocer intenciones del usuario**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID: | CU-3 | | | |
| Nombre | Reconocer intenciones del usuario | | | |
| Actores | Chat-bot | | | |
| Objetivo | Permitir responder preguntas | | | |
| Urgencia | 3 | | | |
| Esfuerzo | 4 | | | |
| Pre- condiciones | Estar autenticado | | |  |
| **Flujo** | **USUARIO** | **SISTEMA** | **CHAT-BOT** | **SOPORTE** |
| **Normal** |  |  |  |  |
|  |  |  |  |
|  | Emite un mensaje |
|  |  | Procesa los mensajes |  |  |
|  |  | Envía al chat bot |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  | Procesa las palabras |  |
|  |  | Analiza e identifica la intención |  |
|  |  | Responde a la intención identificada |  |
|  | Envía mensaje al usuario |  |  |
| Recibe el mensaje del chat bot |  |  |  |

}

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  | automáticamen te |  |
|  | Despliega respuesta |  |  |
| Visualiza la respuesta |  |  |  |

**CASO No.4 Textos asociados**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID: | CU-4 | | | |
| Nombre | Textos asociados | | | |
| Actores | Chat-bot | | | |
| Objetivo | Permitir reconocer las intenciones del usuario | | | |
| Urgencia | 2 | | | |
| Esfuerzo | 3 | | | |
| Pre- condiciones | Estar autenticado | | |  |
|  | **USUARIO** | **SISTEMA** | **CHAT-BOT** | **SOPORTE** |
| Iniciar conversación |  |  |  |
| Selecciona conversación |  |  |  |
| Escribe |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | Procesa las letras |  |  |
|  |  | Envía letras a chat-bot |  |  |
| **Flujo Normal** |  |  | 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 |  | 1-Baja | 2-Menor | 3- | 4-Alta | 5- |
| pa |  |  | Moderada |  | Obligatoria |
| ct |  |  |  |  |  |
|  |  |  |  |  |  |
| o | 5-Muy | 5 | 10 | 15 | 20 | 25 |
|  | alto |  |  |  |  |  |
|  |  |  |  |  | **CU-1** |  |
|  |  | **CU-2** |
|  | 4-Alto | 4 | 8 | 12 | 16 | 20 |
|  |  |  |  | **CU-3** |  |  |
|  | 3-Medio | 3 | 6 | 9 | 12 | 15 |
|  |  |  | CU-4 |  |  |  |
|  | 2-Bajo | 2 | 4 | 6 | 8 | 10 |
|  |  |  |  |  |  |  |
|  | 1-Muy | 1 | 2 | 3 | 4 | 5 |
|  | bajo |  |  |  |  |  |
|  |  |  |  |  |  |  |

1. 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.

## Modelado E/R

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

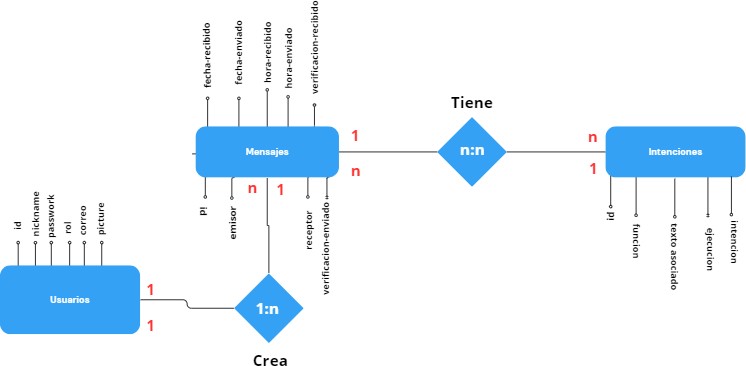
|  |  |  |
| --- | --- | --- |
| **CU-1 Enviar mensaje** | Variables | Datos |
| **Mensaje Enviado** | Contenido: Hola, necesito ayuda para encontrar los contenidos. |
| **Verificación de enviado** | True |
| **Verificación de recibido** | True |
| **Hora de enviado** | 8:23 AM |
| **Fecha de enviado**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

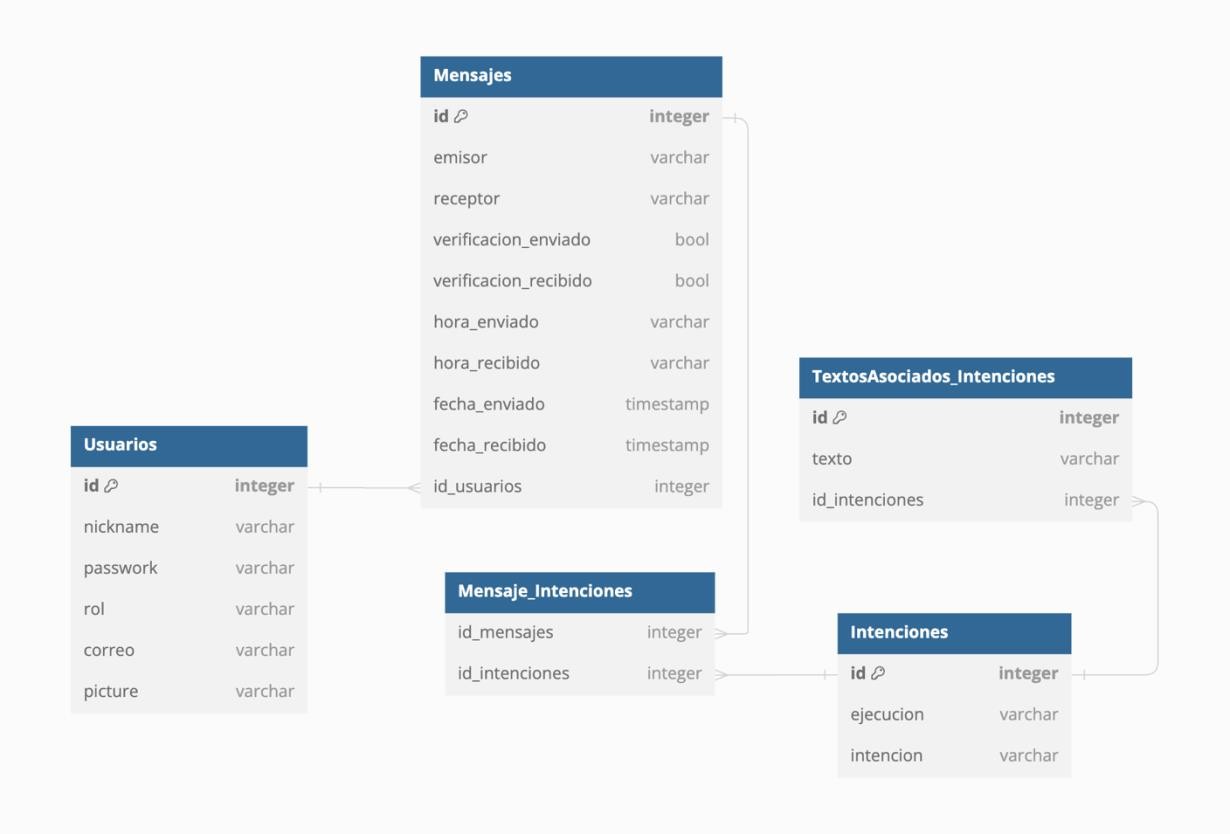
|  |  |  |
| --- | --- | --- |
| **CU-3 Reconocer intenciones del usuario** |  |  |
| **Mensaje de usuario** | ¿Dónde puedo encontrar los contenidos? |
| **Preguntas Frecuentes** | Lista de preguntas.  ¿Cómo buscar contenidos? |

|  |  |  |
| --- | --- | --- |
| **CU-4 Textos asociados** |  |  |
| **texto asociado** | Hola. Necesito ayuda para resolver un problema podrias  contactarme con soporte |
| **función** | 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 passwork 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:

###### User (Usuario):

Almacena información sobre los usuarios que pueden acceder al módulo de mensajes

**Atributos:** ID (identificador único), nickname(), passwork(),rol(usuario, administrador), correo(dirección de correo electrónico para comunicación), picture(imagen de perfil del usuario)

**Relaciones:** Cada usuario puede estar asociado con varios mensajes

###### Messages (Mensajes):

Almacena información de los mensajes de los usuarios en el módulo.

**Atributos:** ID (identificador único), emisor (ente que envía un mensaje), receptor (ente que recibe un mensaje), verificación\_enviado(marca de enviado), verificacio\_recibido(marca de recibido), hora\_enviado(números), hora recibido(números), fecha\_recibido(letras y números), fecha\_enviado(letras y números), id\_usuarios()

**Relaciones:** Un mensaje puede ser escrito por un solo usuario, pero puede tener muchas intenciones.

###### Intentions (intenciones):

Almacena contenido en el modulo de lo que quiere un usuario a través de su mensaje.

**Atributos:** ID (identificador único), funcion(texto variado), texto\_asociado(texto variado que incluye la intención ), ejecicion(), intención()

**Relaciones:** Las intenciones se asocian con un usuario a través de la relación con los mensajes.

###### Associated texts (Textos asociados):

Se asocia a la intención del usuario relacionado con los mensajes.

**Atributos:** ID (identificador único), texto, id\_intenciones

**Relaciones:** Cada mensaje puede tener varias intenciones y una intención puede tener varios mensajes

#### Relaciones:

* "User" se relaciona con "Messages" para indicar que un usuario puede tener varios mensajes.
* "Messages" se relaciona con "Intentions" para registrar que los usuarios que pueden tener intenciones e n los mensajes.
* "Messages" se relaciona con "User" para registrar que los usuarios que pueden tener varios mensajes.
* "intenciones" se relaciona con "Textos asociados" para registrar que las intenciones se derivan de los textos.

### Reglas de Integridad Referencial

1. **Integridad Referencial entre "Usert" y "menssages"**: Cada registro en el módulo "Messages" debe estar asociado con un usuario existente en la tabla "Users" a través de la clave foránea "user\_id". Esto asegura que cada mensaje enviado o recibido esté asociado a un usuario válido en el sistema.
2. **Integridad Referencial entre "Messages" y "Conversations":** Cada registro en el módulo "Messages" debe estar asociado con una conversación existente en "Conversations" a través de la clave foránea "conversation\_id".
3. **Integridad Referencial entre "Intents" y "Messages":** Cada registro en la tabla "Intents" debe estar asociado con un mensaje existente en la tabla "Messages" a través de la clave foránea "message\_id". Descripción: Esto asegura que cada intención reconocida esté vinculada a un mensaje específico que la generó.
4. Integridad Referencial entre "Intents" y "Associated\_Texts": Cada registro en la tabla "Associated\_Texts" debe estar asociado con una intención existente en la tabla "Intents”. Esto asegura que cada texto asociado para la identificación de intenciones esté vinculado a una intención válida en el sistema.

### Colecciones (NoSLQ)

###### 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

}

### Anexos Diagramas Adicionales Referencias

Etapa 2: Persistencia de Datos con Backend

## Introducción

La persistencia de la base de datos se refiere a los datos o metadatos guardados durante la ejecución de una integración, En esta etapa, detallaremos cómo se gestionan, almacenan y recuperan los datos a través del backend, garantizando la integridad y disponibilidad de la información en el sistema. Utilizamos MongoDB Atlas como base de datos y Postman para pruebas y conexiones.

### Propósito de la Etapa

Esta etapa tiene como objetivo crear un sistema sólido y eficiente para la persistencia de datos que apoye las funciones del módulo de mensajes. Esto incluye procesar las intenciones del usuario, enviar y recibir mensajes de texto y garantizar que todos los datos estén correctamente almacenados y recuperados de manera confiable.

Alcance de la Etapa

 **Diseño de la Base de Datos:**

* Definición de las colecciones necesarias ( Users, Messages, Intents, Associated\_Texts).
* Establecimiento de las relaciones entre las colecciones usando claves foráneas y referencias.

 **CRUD Operaciones:**

* Implementación de operaciones de creación, lectura, actualización y eliminación (CRUD) para cada colección.

 **Conexión y Configuración del Backend:**

* Configuración de la conexión con MongoDB Atlas.
* Desarrollo de endpoints en el backend para gestionar las operaciones CRUD.

 **Integración y Pruebas:**

* Uso de Postman para probar y validar los endpoints del backend.

### Definiciones y Acrónimos

#### Definiciones

* **Backend:** Parte del sistema que se encarga de la lógica del servidor, gestión de bases de datos, y procesamiento de solicitudes del cliente. Es invisible para el usuario final.
* **CRUD:** Acrónimo de Create, Read, Update, Delete. Se refiere a las operaciones básicas que se pueden realizar en una base de datos.
* **Intención (Intent):** Representa la intención del usuario detrás de una entrada de texto. Por ejemplo, una intención podría ser "saludo" o "consulta".
* **Mensaje (Message):** Unidad de comunicación que contiene texto, multimedia o ambos, enviado entre usuarios o desde el sistema al usuario.
* **Texto asociado (Associated Text):** Fragmentos de texto que se utilizan para identificar intenciones específicas en los mensajes del usuario.

 **Usuario (User):** Persona que interactúa con el sistema, enviando y recibiendo mensajes.

* **MongoDB Atlas:** Servicio de base de datos en la nube para MongoDB, que ofrece una plataforma escalable y segura para almacenar y gestionar datos.

#### Acrónimos

* **API:** Application Programming Interface (Interfaz de Programación de Aplicaciones)
* **CRUD:** Create, Read, Update, Delete
* **DB:** Database (Base de Datos)
* **ID:** Identifier (Identificador)
* **JSON:** JavaScript Object Notation (Notación de Objetos de JavaScript)
* **REST:** Representational State Transfer (Transferencia de Estado Representacional)
* **HTTPS:** Hypertext Transfer Protocol Secure (Protocolo Seguro de Transferencia de Hipertexto)

## Diseño de la Arquitectura de Backend

La arquitectura propuesta para el backend del módulo de mensajes está diseñada para ser escalable, segura y eficiente. Utiliza una estructura basada en microservicios para gestionar las distintas funcionalidades del sistema, permitiendo una fácil implementación y mantenimiento.

### CRUD Operaciones

**Ejemplos de Endpoints CRUD:**

1. **Users**

* **Crear Usuario:** POST /users
* **Buscar Usuario:** GET /users/:id
* **Actualizar Usuario:** PUT /users/:id
* **Eliminar Usuario:** DELETE /users/:id

1. **Messages**

* **Crear Mensajes:** POST /messages
* **Buscar Mensajes:** GET /messages/:id
* **Actualizar Mensajes:** PUT /messages/:id
* **Actualizar Mensajes:** PATCH /messages/:id
* **Eliminar Mensajes:** DELETE /messages/:id

1. **Intents**

* **Crear Intención:** POST /intents
* **Buscar Intención:** GET /intents/:id
* **Actualizar Intención:** PUT /intents/:id
* **Actualizar Intención:** PACH /intents/:id
* **Eliminar Intención:** DELETE /intents/:id

1. **Associated\_Texts**

* **Crear Texto Asociado:** POST /associated\_texts
* **Leer Texto Asociado:** GET /associated\_texts/:id
* **Actualizar Texto Asociado:** PUT /associated\_texts/:id
* **Eliminar Texto Asociado:** DELETE /associated\_texts/:id



### Descripción de la Arquitectura Propuesta

Componentes del Backend

Diagramas de Arquitectura

1. Elección de la Base de Datos

* Evaluación de Opciones (SQL o NoSQL)
* Justificación de la Elección
* Diseño de Esquema de Base de Datos

1. Implementación del Backend

* Elección del Lenguaje de Programación
* Creación de la Lógica de Negocio
* Desarrollo de Endpoints y APIs
* Autenticación y Autorización

1. Conexión a la Base de Datos

* Configuración de la Conexión
* Desarrollo de Operaciones CRUD
* Manejo de Transacciones

1. Pruebas del Backend

* Diseño de Casos de Prueba
* Ejecución de Pruebas Unitarias y de Integración
* Manejo de Errores y Excepciones

## Elección de la Base de Datos

### Evaluación de Opciones (SQL o NoSQL)

### Para el proyecto de mensajería detallado en este documento, donde se mencionan características como el manejo de mensajes, intenciones y texto asociado, la flexibilidad y escalabilidad proporcionada por una base de datos **NoSQL** como MongoDB Atlas puede ser más adecuada. Esto se debe a la naturaleza dinámica y potencialmente voluminosa de los datos manejados.

Justificación de la Elección

La elección de MongoDB Atlas como la base de datos NoSQL para el proyecto de mensajería se justifica por su flexibilidad de esquemas, capacidad de escalabilidad horizontal, alto rendimiento en operaciones de lectura y escritura, y eficiencia en el manejo de datos desestructurados. Estas características son esenciales para satisfacer los requisitos dinámicos y de alto rendimiento del sistema de mensajería, asegurando que pueda crecer y adaptarse según sea necesario.

Diseño de Esquema de Base de Datos

Aquí se presenta un esquema propuesto para la aplicación de mensajería, considerando los requisitos funcionales y las mejores prácticas para MongoDB.

**Entidades Principales**

**1. Usuarios**

**2. Mensajes**

**3. Conversaciones**

**4. Intenciones**

**Esquema**

**1. Usuarios**

Cada usuario tendrá un documento que almacena información relevante sobre el usuario.

{

"\_id": ObjectId("60d5ec49d6e01e1d8c8b4567"),

"nombre": "Juan Pérez",

"email": "juan.perez@example.com",

"fechaRegistro": ISODate("2023-06-21T00:00:00Z"),

"estado": "activo",

"conversaciones": [

{

"conversacionId": ObjectId("60d5ec49d6e01e1d8c8b1234"),

"ultimoMensaje": ISODate("2024-06-21T12:00:00Z")

}

]

}

**2. Mensajes**

Cada mensaje estará representado como un documento, permitiendo almacenar los metadatos y el contenido del mensaje.

{

"\_id": ObjectId("60d5ec49d6e01e1d8c8b7890"),

"conversacionId": ObjectId("60d5ec49d6e01e1d8c8b1234"),

"remitenteId": ObjectId("60d5ec49d6e01e1d8c8b4567"),

"contenido": "Hola, ¿cómo estás?",

"timestamp": ISODate("2024-06-21T12:00:00Z"),

"tipo": "texto",

"leido": false

}

**3. Conversaciones**

Cada conversación almacenará la información de los participantes y los mensajes.

{

"\_id": ObjectId("60d5ec49d6e01e1d8c8b1234"),

"participantes": [

ObjectId("60d5ec49d6e01e1d8c8b4567"),

ObjectId("60d5ec49d6e01e1d8c8b8901")

],

"mensajes": [

{

"mensajeId": ObjectId("60d5ec49d6e01e1d8c8b7890"),

"timestamp": ISODate("2024-06-21T12:00:00Z")

}

]

}

```

**4. Intenciones**

Las intenciones de los mensajes, como comandos o solicitudes específicas, también estarán almacenadas.

{

"\_id": ObjectId("60d5ec49d6e01e1d8c8b4568"),

"mensajeId": ObjectId("60d5ec49d6e01e1d8c8b7890"),

"intencion": "consulta\_clima",

"parametros": {

"ciudad": "Madrid",

"fecha": "2024-06-21"

}

}

**Relación entre las Entidades**

**- Usuarios y Conversaciones:** Un usuario puede participar en múltiples conversaciones.

**- Conversaciones y Mensajes:** Cada conversación contiene múltiples mensajes.

**- Mensajes y Intenciones:** Cada mensaje puede tener una intención asociada, dependiendo del contenido del mensaje.

**Conclusión**

El diseño del esquema de base de datos en MongoDB Atlas permite aprovechar la flexibilidad y escalabilidad de NoSQL, adaptándose a los requisitos dinámicos de la aplicación de mensajería. Este esquema no solo facilita el manejo de grandes volúmenes de datos y su rápida consulta, sino que también asegura que la estructura de datos pueda evolucionar con el tiempo sin necesidad de cambios disruptivos en el modelo.

## Implementación del Backend

### Elección del Lenguaje de Programación

La elección del lenguaje de programación es un paso fundamental en el desarrollo de cualquier sistema de software. Esta decisión debe basarse en varios factores, como los requisitos del proyecto, la experiencia del equipo de desarrollo, las características del lenguaje y el ecosistema de herramientas disponibles. En este contexto, analizaremos las razones para seleccionar un lenguaje de programación adecuado para el sistema de mensajería descrito en este documento.

#### **Requisitos del Proyecto**

1. **Rendimiento**: El sistema de mensajería debe ser capaz de manejar un alto volumen de mensajes en tiempo real.
2. **Escalabilidad**: Debe ser fácil de escalar para soportar un número creciente de usuarios y mensajes.
3. **Integración con MongoDB**: Debe integrarse eficientemente con la base de datos NoSQL seleccionada (MongoDB Atlas).
4. **Desarrollo Rápido**: El lenguaje debe permitir un desarrollo rápido y ágil para iterar rápidamente sobre nuevas funcionalidades.
5. **Comunidad y Soporte**: Una comunidad activa y un buen soporte de herramientas y librerías son esenciales para resolver problemas y mejorar la productividad del equipo.

#### **Candidatos Principales**

1. **JavaScript (Node.js)**
2. **Python**
3. **Java**
4. **C#**

Creación de la Lógica de Negocio

Para el sistema de mensajería, la lógica de negocio se encargará de manejar las operaciones relacionadas con usuarios, mensajes, conversaciones e intenciones.

#### **Arquitectura**

La arquitectura de la lógica de negocio se compondrá de varios servicios y controladores que interactúan con la base de datos y otros componentes de la aplicación. Aquí se presenta un diseño general utilizando JavaScript (Node.js) y el framework Express.js, que es adecuado para la creación de aplicaciones web rápidas y escalables.

#### Componentes de la Lógica de Negocio

1. **Servicio de Usuarios**
2. **Servicio de Mensajes**
3. **Servicio de Conversaciones**
4. **Servicio de Intenciones**

### Desarrollo de Endpoints y APIs

La implementación de los endpoints y APIs en Node.js con Express.js permite crear un sistema de mensajería robusto y escalable. Los servicios están modularizados para facilitar el mantenimiento y la escalabilidad. Este diseño asegura que el sistema puede manejar eficientemente las operaciones CRUD para usuarios, mensajes, conversaciones e intenciones, proporcionando una base sólida para el desarrollo de nuevas funcionalidades en el futuro.

## Etapa 3: Consumo de Datos y Desarrollo

Frontend

## Introducción

### Propósito de la Etapa

El objetivo de esta etapa es crear una interfaz intuitiva y funcional que permita a los usuarios utilizar toda la transparencia del componente de mensajería. El objetivo principal es desarrollar una mejor experiencia de usuario para que los usuarios puedan enviar y recibir mensajes en tiempo real, interactuar con el chatbot y solicitar un servicio de ayuda eficiente. Se establecerá una conexión sólida entre la interfaz gráfica y la lógica trasera, garantizando que el mensaje y la manipulación de los mensajes del chatbot y las respuestas automatizadas sean precisos y rápidos.

### Alcance de la Etapa

En esta sección, nos centramos en la implementación frontal de la aplicación de mensajería. Estos incluyen:

• Diseño y desarrollo de pantallas: Se desarrollarán interfaces para conversaciones con chatbots, mensajería entre usuarios y soporte técnico.

• Lógica de interacción: Los eventos dinámicos y el comportamiento de la interfaz de usuario permitirán enviar mensajes, recibir respuestas en tiempo real y gestionar notificaciones.

• Integración con el backend: se debe crear una lógica de interacción con el backend para recibir y enviar datos, garantizar mensajes consistentes e identificar los objetivos del chatbot.

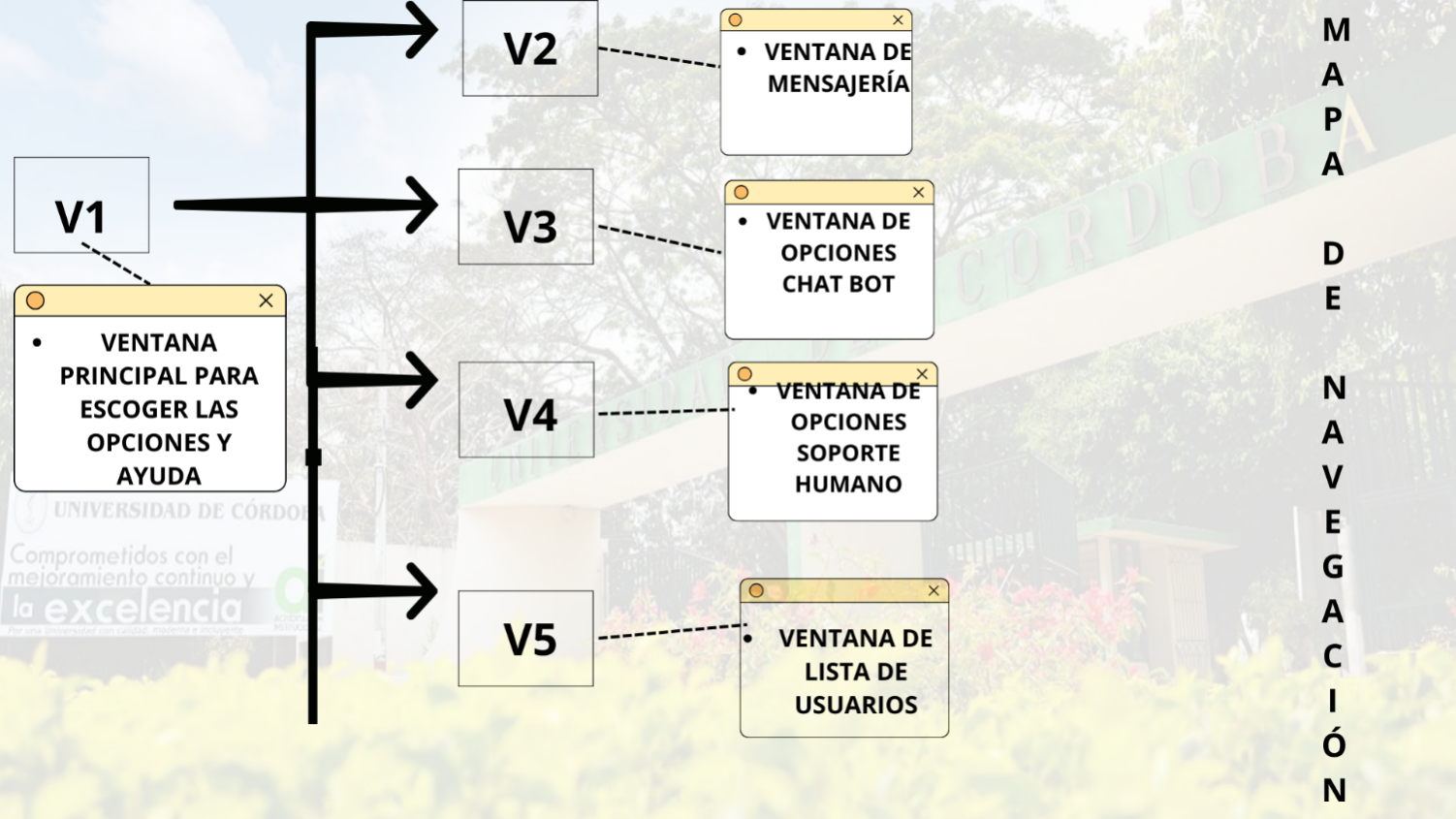
• Pruebas exhaustivas: se deben realizar pruebas funcionales y de usabilidad para garantizar que el sistema funcione correctamente en todos los dispositivos y escenarios, permita una comunicación fluida y que las notificaciones en tiempo real puedan manejarlas adecuadamente.

### Definiciones y Acrónimos

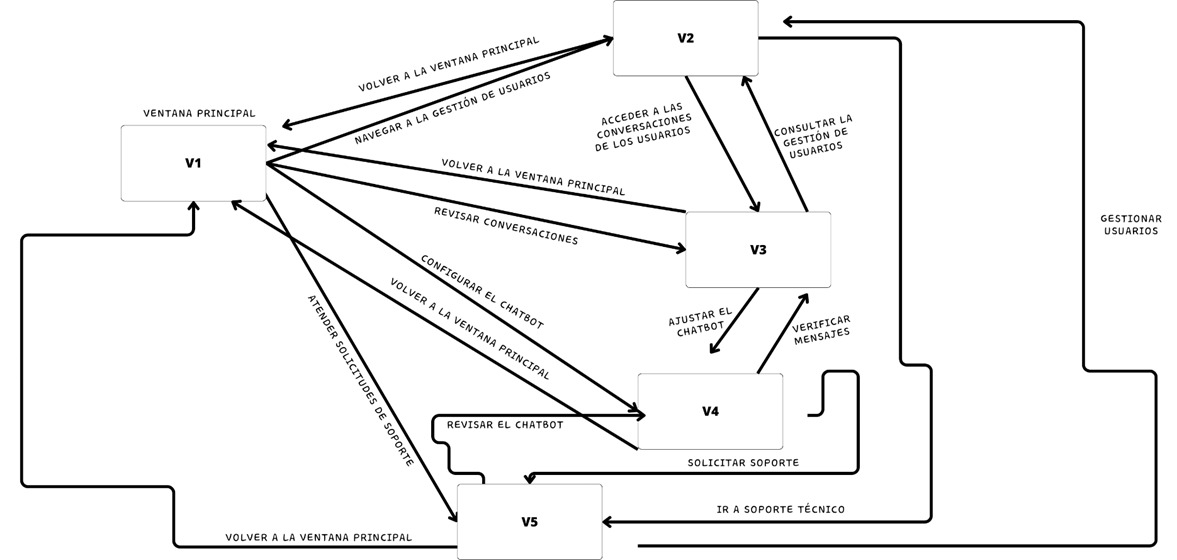
* **Interfaz de Usuario (UI):** La interfaz visual que permite a los usuarios interactuar con la aplicación. Esto incluye botones, menús, formularios y cualquier otro elemento que los usuarios manipulen dentro del sistema de mensajería​.
* **Experiencia de Usuario (UX):** Hace referencia a la satisfacción y facilidad que experimenta un usuario al interactuar con la aplicación, asegurando que la interfaz de mensajería sea clara y fácil de usar​.
* **Maquetación Responsiva:** Proceso de diseño que asegura que la interfaz gráfica del sistema de mensajería se vea bien en cualquier dispositivo, desde móviles hasta pantallas grandes​.
* **CSS (Cascading Style Sheets):** Lenguaje utilizado para definir el estilo visual de la interfaz de usuario, incluyendo colores, disposición y formato de los elementos del chat​.
* **HTML (Hypertext Markup Language):** Lenguaje utilizado para estructurar la interfaz de la mensajería, definiendo los elementos visuales como los campos para enviar y recibir mensajes​.
* **JavaScript:** Lenguaje de programación empleado para gestionar la lógica del frontend, como la actualización en tiempo real de los mensajes o la interacción con el chatbot​.
* **API (Application Programming Interface):** Conjunto de reglas y protocolos que permiten la comunicación entre el frontend y el backend, esencial para enviar y recibir datos de los mensajes y la interacción con el chatbot​.

## Creación de la Interfaz de Usuario (UI)

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



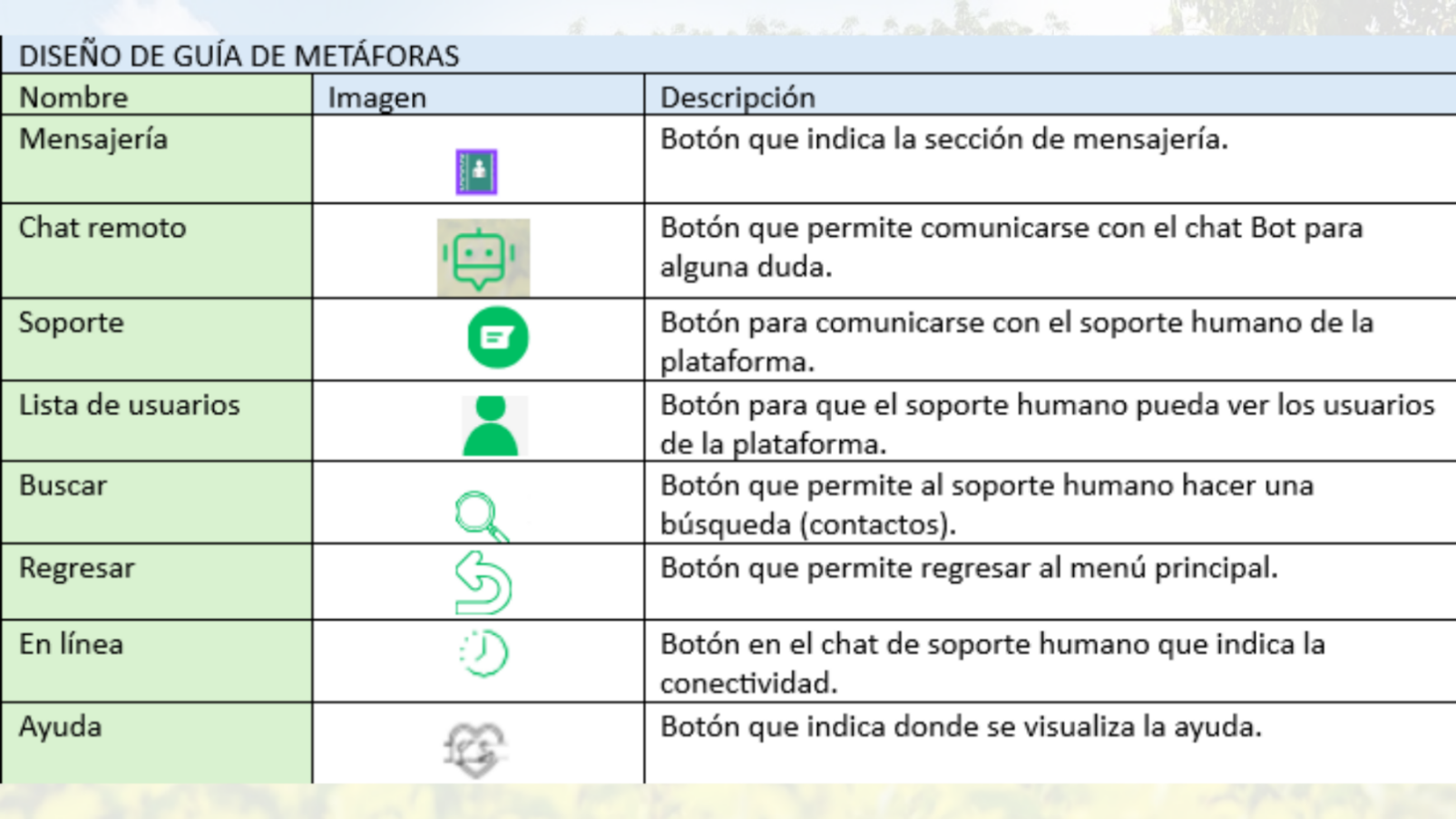
Versión 2 mejorada



**Diseño de las ventanas**



**Diseño de guía de metáforas**

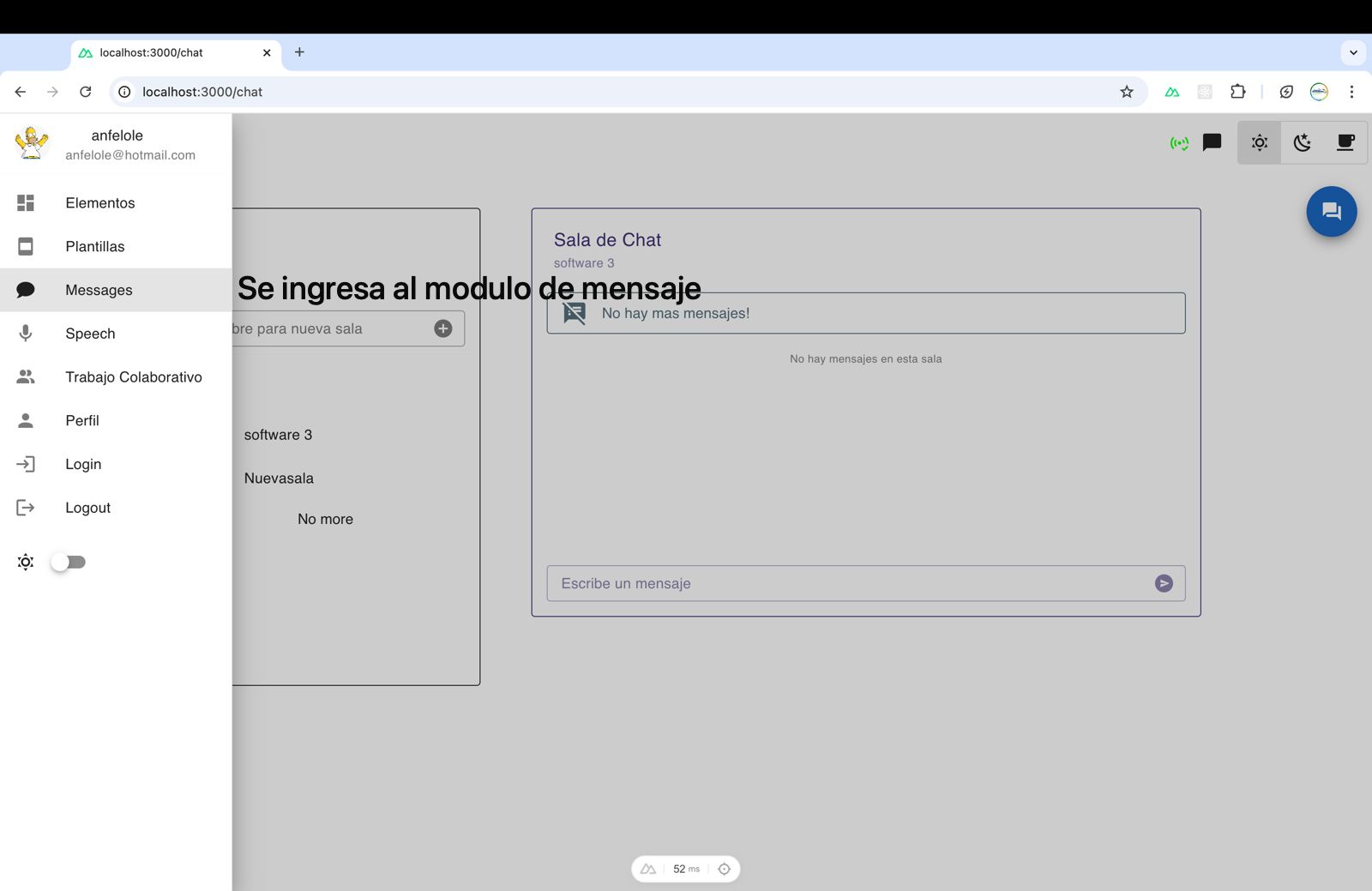


# Diagramas UML

* **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.

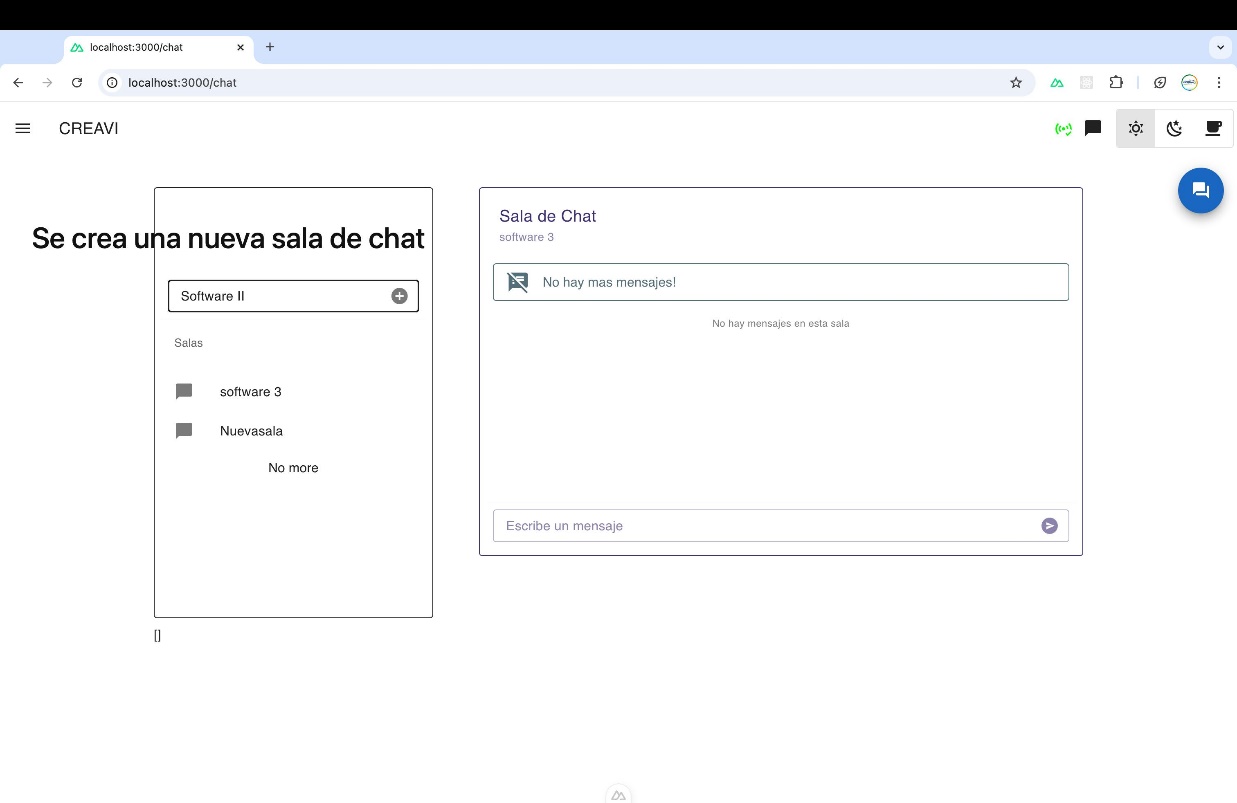
Anexos.

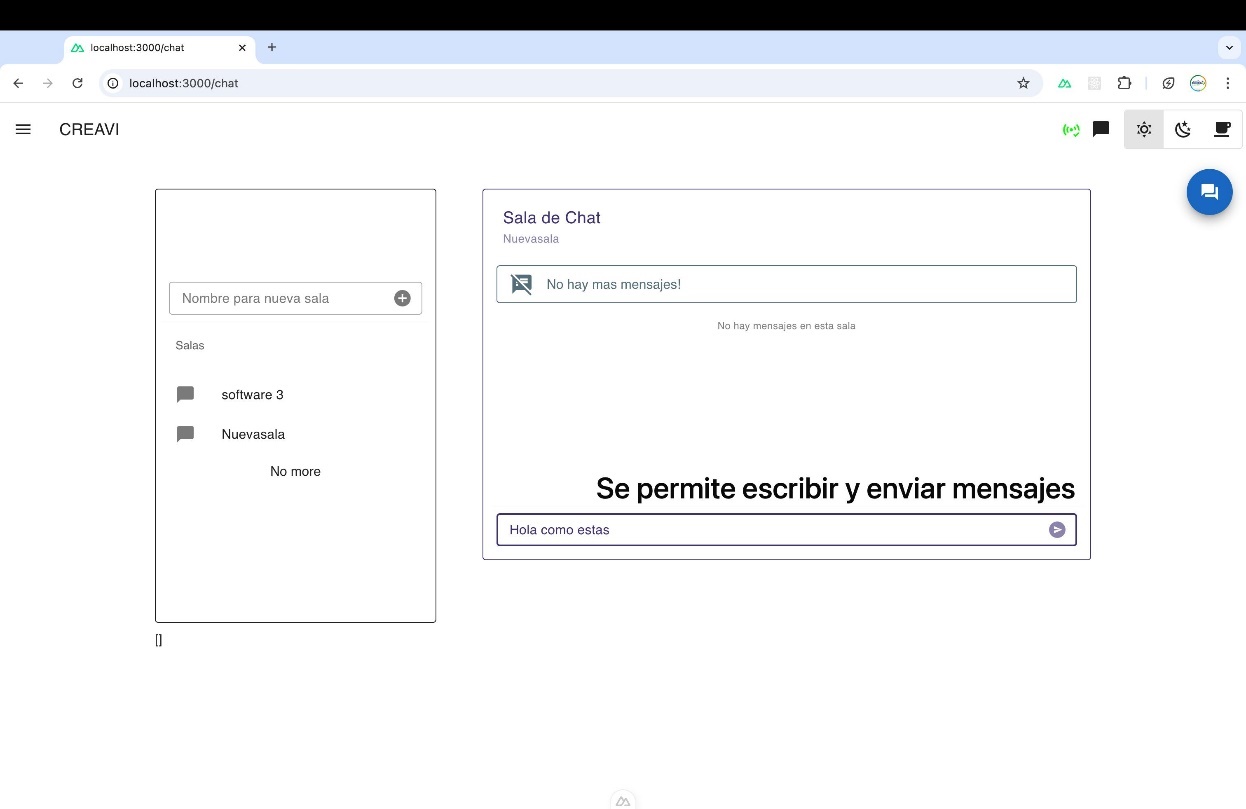
Ventana principal



En esta ventana podemos observar la interfaz de nuestro producto en el que se observa en la parte izquierda con diferentes secciones de la plataforma Creavi, como lo son la de usuario, elementos, plantillas, y la de mensajería. A su vez se muestra que al fondo esta abierta la sala de mensajería, y que se puede navegar de manera fluida en otras secciones de Creavi. Se observa un usuario, el estado de conexión del mismo, la opción para cambiar a modo oscuro entre otras, este componente de mensajería posee un diseño minimalista y funcional.

Ventana de mensajería





En esta ventana se encuentra la opción para elegir el nombre de la sala que se desea crear, que junto con la opción + se puede crear dicha sala. Al ingresar a la sala creada se pueden escribir mensajes los cueles se van a asociar a las salas que estén ya creadas, los mensajes contienen hora de envió totalmente visible, el contenido en forma de texto y el nombre del usuario que envía el mensaje. Cumpliéndose el objetivo de crear un chat Bot en el que se pudiera establecer comunicación entre usuarios y soporte técnico.

Se puede observar que del lado izquierdo de la plataforma están almacenadas las salas que han sido creadas. De lado derecho esta la ventana de mensajería y a su vez un botón azul flotante que permite acceder a sección de mensajería sin interrumpir la navegación.

Conclusión

A lo largo del desarrollo de este componente se han logrado adquirir habilidades como la resolución de problemas, el trabajo en equipo, se adquirieron conocimientos teórico prácticos en el campo de la programación.

En cuento al componente se logra cumplir en su mayoría con los casos de usos establecidos, de acuerdo al tiempo y espacio de trabajo, se avanzó de manera constante en el desarrollo del mismo para lograr aportar a la mejora de la plataforma Creavi, con este componente se mejora la navegación de los usuarios permitiendo que algo tan importante como la comunicación funciones de manera correcta y eficaz.