

# **TECNOLÓGICO NACIONAL DE MEXICO**

## **CAMPUS ENSENADA**

### **Ingeniería de Software.**

RESPONDITE

DISEÑO DE SOFTWARE

#### **ALUMNOS:**

Alonso Hernandez Devorah 22760237

Angulo Martinez Angel Gabriel 22760239

Castillo Escareño Coral 22760231

Gaynor Zurita Cindy Geraldine 22760927

Zerega Navarro Eduardo Isidro 22760601

6SF

#### **DOCENTE:**

Luis Armando Cárdenas Florido

Ensenada, Baja California a 10 de Marzo de 2025.

# ÍNDICE

<b>ÍNDICE</b>	<b>2</b>
<b>1.- DISEÑO ARQUITECTÓNICO</b>	<b>3</b>
• 1.1 PATRON ARQUITECTÓNICO	3
• 1.2.- DIAGRAMAS DE ARQUITECTURA DEL SISTEMA QUE INTEGRE Y MUESTRE A TODOS LOS MÓDULOS DEL SISTEMA A DESARROLLAR	4
• 1.3.- JUSTIFICACIÓN DE LAS DECISIONES ARQUITECTÓNICAS TOMADAS	5
<b>2.- DISEÑO DE PROCESOS</b>	<b>6</b>
• 2.1.- DIAGRAMA DE FLUJO DE PROCESOS	6
• 2.2.- DESCRIPCIÓN DE LA LÓGICA DE LOS PROCESOS CLAVE EN EL SISTEMA	7
<b>3.- DISEÑO DE DATOS</b>	<b>8</b>
▪ 3.1.- DIAGRAMA DE CLASES	8
▪ 3.2.- MODELO ENTIDAD-RELACIÓN (ER):	9
▪ 3.3.-Normalización y justificación de la estructura de datos elegida.	10
<b>4.- DISEÑO DE INTERFAZ DE USUARIO</b>	<b>13</b>
4.1.- Mockups	13
▪ 4.2.- Diseño de navegación entre pantallas.	19
4.3.- Consideraciones de usabilidad y experiencia de usuario (UX/UI).	23
<b>REFERENCIAS:</b>	<b>24</b>

## 1.- DISEÑO ARQUITECTÓNICO

- 1.1 PATRON ARQUITECTÓNICO

El sistema sigue el patrón Cliente-Servidor, con una arquitectura basada en Capas:

- **CLIENTE-SERVIDOR**

El frontend (cliente) realiza solicitudes al backend (servidor) a través de una API REST.

El backend procesa las solicitudes y se comunica con la base de datos.

La base de datos en la nube (PostgreSQL en Supabase) almacena la información necesaria para la operación de la IA.

- **CAPAS**

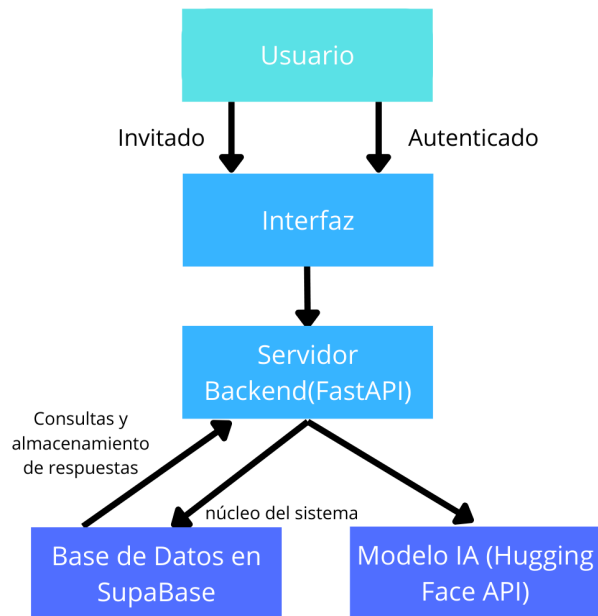
Capa de Presentación: Interfaz gráfica web donde los usuarios interactúan con el chat.

Capa de Lógica de Negocio: Backend que maneja las peticiones del usuario y la comunicación con la IA.

Capa de Datos: Base de datos en la nube que almacena información sobre usuarios, chats y preguntas frecuentes.

- 1.2.- DIAGRAMAS DE ARQUITECTURA DEL SISTEMA QUE INTEGRE Y MUESTRE A TODOS LOS MÓDULOS DEL SISTEMA A DESARROLLAR

### Diagrama de Bloques



*Figura 1. Diagrama de Bloques*

El diagrama de bloques muestra la arquitectura general del sistema y cómo interactúan sus componentes:

**Usuario:** Puede acceder como invitado o como usuario autenticado.

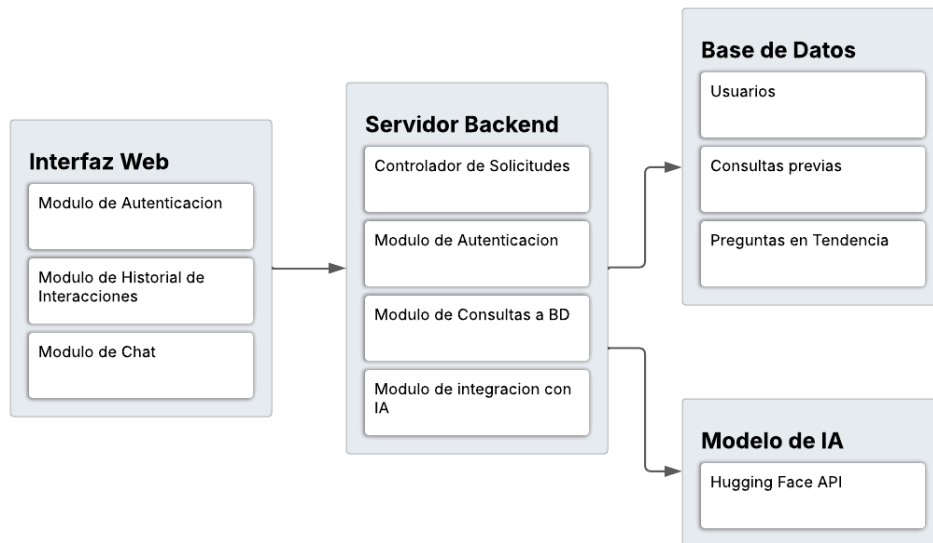
**Interfaz:** Es el puente visual que conecta al usuario con el sistema, permitiéndole realizar funciones como ingresar mensajes y visualizar respuestas.

**Servidor Backend (FastAPI):** Se encarga de recibir y procesar las solicitudes desde la interfaz. Es el núcleo lógico del sistema.

**Modelo IA (Hugging Face API):** Se encarga de interpretar y generar respuestas a los mensajes del usuario.

**Base de Datos en Supabase:** Almacena las respuestas, registros de usuarios y demás datos necesarios para el funcionamiento del sistema.

## Diagrama de Componentes



*Figura 2. Diagrama de Componentes*

Este diagrama ilustra la arquitectura modular del sistema, mostrando cómo se distribuyen las responsabilidades entre la interfaz web, el servidor backend, la base de datos y el modelo de inteligencia artificial. La interfaz permite la autenticación, el manejo de chats y la visualización del historial. El backend coordina las solicitudes, autentica usuarios, consulta datos y se comunica con el modelo de IA (Hugging Face API). La base de datos almacena información sobre usuarios y consultas.

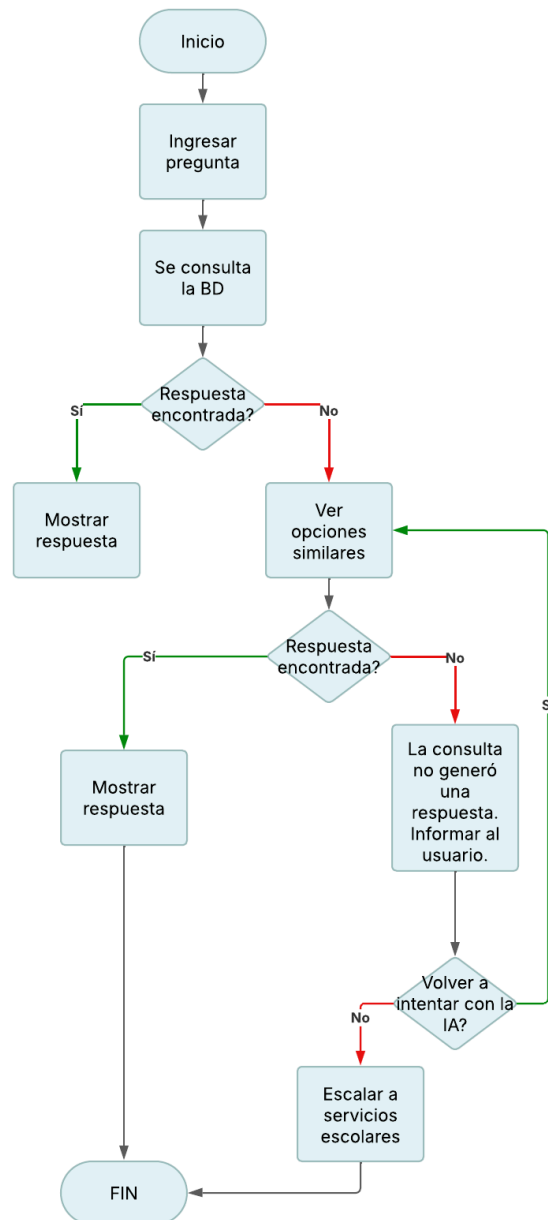
### ● 1.3.- JUSTIFICACIÓN DE LAS DECISIONES ARQUITECTÓNICAS TOMADAS

Se eligió el patrón Cliente-Servidor porque al poder separar el proyecto en partes esto facilita la escalabilidad y el mantenimiento del sistema.

Además, se implementa el patrón por capas, dividiéndolo en partes similares a las anteriores, pero con este patrón ayudando en la mejora de la organización y reutilización del código.

## 2.- DISEÑO DE PROCESOS

### ● 2.1.- DIAGRAMA DE FLUJO DE PROCESOS



*Figura 3. Diagrama de flujo de procesos*

Este diagrama representa la lógica que sigue una consulta realizada por el usuario. Comienza con el ingreso de una pregunta, seguida de una búsqueda en la base de datos. Si se encuentra una respuesta, se muestra directamente. Si no, se ofrecen opciones similares. Si aún no se obtiene una respuesta, se decide si intentar con la IA. En caso de no éxito, la consulta es escalada a servicios escolares. El flujo permite visualizar la toma de decisiones dentro del sistema.

- **2.2.- DESCRIPCIÓN DE LA LÓGICA DE LOS PROCESOS CLAVE EN EL SISTEMA**

- **MANEJO DE CONSULTAS**

Un usuario ingresa su pregunta en la interfaz.

El backend revisa si la pregunta está en la base de datos.

Si la respuesta está en la base de datos se envía directamente.

Si no, se busca el problema del usuario y se mira si hay algo similar.

Si eso tampoco funciona se proporciona una opción para contactar al estudiante con servicios escolares por medio de un correo electrónico.

- **GESTIÓN DE USUARIOS Y AUTENTICACIÓN**

Los usuarios pueden autenticarse creando una cuenta, o pueden acceder como invitados.

Para crear cuentas de administrador se deberán de ingresar directamente en la base de datos.

Los administradores tienen acceso a funciones adicionales como la gestión de la información sobre los procesos, guardada en la base de datos.

### 3.- DISEÑO DE DATOS

#### ▪ 3.1.- DIAGRAMA DE CLASES

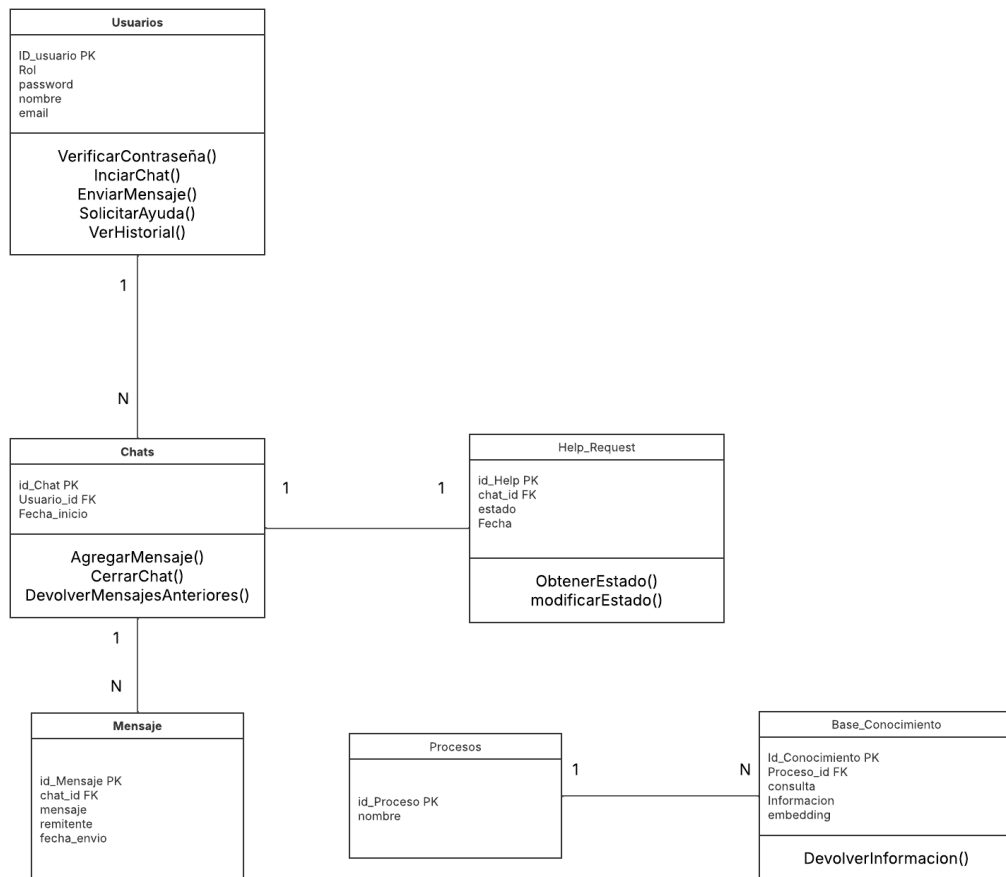


Figura 4. Diagrama de clases

Este diagrama de clases representa el diseño de la base de datos la cual estará compuesta por las siguientes entidades:

- **Usuarios:**  
Cada usuario tiene un identificador único, rol (estudiante o administrador), contraseña, nombre y correo electrónico
- **Chats:**  
Cada chat está relacionado con un usuario, cuentan con un id, y fecha de inicio. Un usuario puede tener uno o muchos chats. Desde esta entidad se pueden agregar mensajes y consultar mensajes anteriores.
- **Mensaje:**  
Cada mensaje pertenece a un chat y contiene información sobre el remitente, el contenido y la



fecha de envío. Un chat puede contener muchos mensajes, pero un mensaje solamente puede pertenecer a un solo chat (1:N).

- **Help\_Request:**  
Relacionada uno a uno con Chats, esta entidad permite gestionar solicitudes de ayuda generadas por los usuarios durante el chat. Incluye estado y fecha de la solicitud.
- **Procesos:**  
Representa los procesos (servicios) disponibles en el sistema. Está relacionado con la base de conocimiento. Tiene un identificador y un nombre.
- **Base\_Conocimiento:**  
Contiene consultas, información y sus representaciones en forma de embeddings, asociadas a un proceso específico. De esta tabla el asistente obtiene información para sus respuestas.

### 3.2.- MODELO ENTIDAD-RELACIÓN (ER):

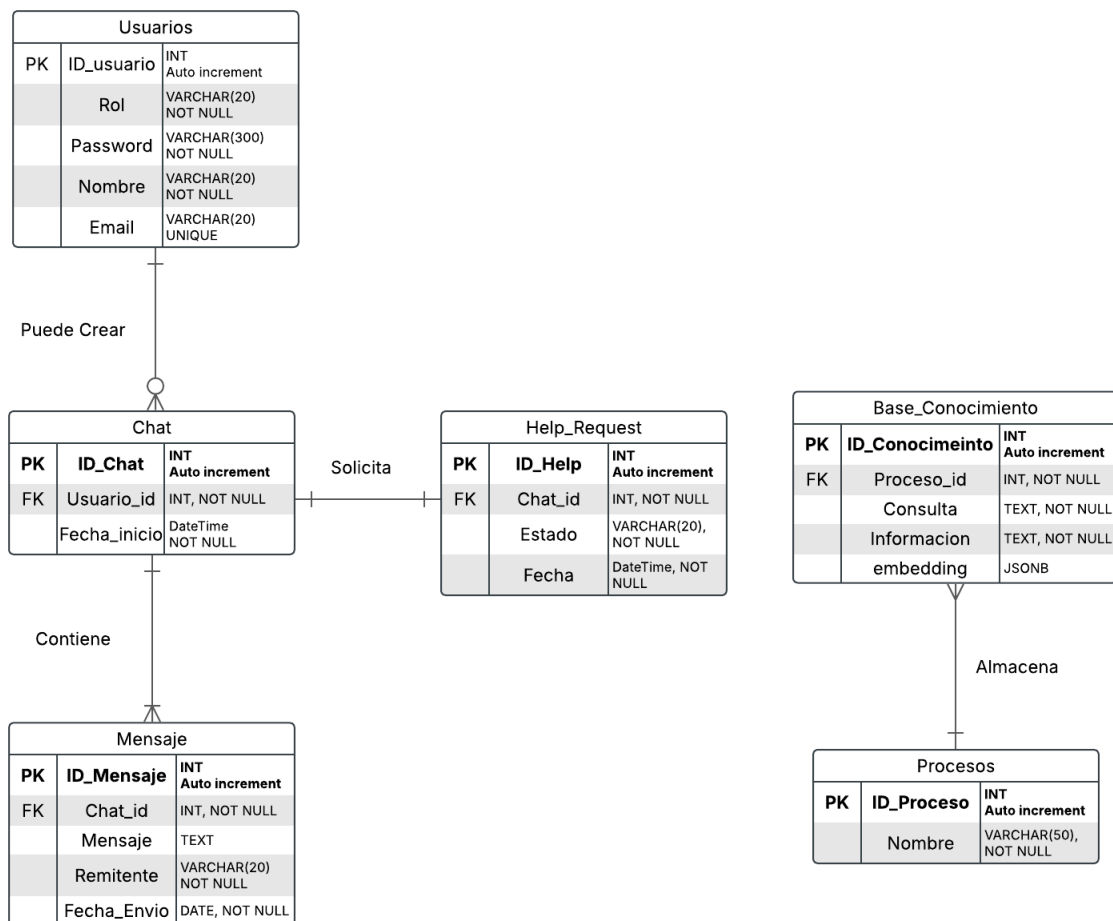


Figura 5. Diagrama de Entidad-Relación

#### Entidades Principales:

- **Usuarios:** Contiene información de los usuarios registrados.

- **Chat:** Cada usuario puede iniciar un chat que contiene mensajes y puede estar asociado a una solicitud de ayuda.
- **Mensaje:** Representa cada mensaje enviado dentro de un chat. Incluye información del remitente, el contenido del mensaje y la fecha de envío.
- **Help\_Request:** Contiene las solicitudes de ayuda generadas por los estudiantes.
- **Procesos:** Contiene los nombres de los diferentes procesos escolares (por ejemplo, credencialización, reinscripción, etc.) para los cuales se registran las consultas.
- **Base\_Conocimiento:** Almacena consultas y sus respuestas (Información). Aunque por el momento no está en uso, cuenta con soporte para embeddings semánticos (embedding). Esta tabla está asociada con la tabla Procesos, que clasifica las consultas por proceso administrativo.

### ▪ 3.3.-Normalización y justificación de la estructura de datos elegida.

Nuestra base de datos ha sido diseñada cumpliendo con las tres primeras formas normales (1NF, 2NF y 3NF), asegurando eficiencia, integridad y escalabilidad.

A continuación se explica la estructura de cada una de las tablas y sus atributos.

**Usuarios:** La tabla Usuarios registra a los estudiantes y al administrador de la página.

Cuenta con los siguientes atributos:

- ID\_Usuarios: Identifica de forma única a cada usuario.
- Rol: Determina si el usuario es un estudiante o administrador.
- Correo: Se usa UNIQUE para evitar registros duplicados.
- Nombre: Almacena el nombre del usuario.
- password: Se guarda la contraseña, esta se utiliza para la autenticación del usuario, se guarda en formato hash para mayor seguridad.

Ejemplo:

ID_Usuario	Rol	Correo	Nombre	password
1	Estudiante	estudiante@gmail.com	Pedro	contraseña123

*Tabla 1. Usuarios*

### ***Chats:***

La tabla Chats almacena cada conversación entre el estudiante y la IA o el administrador.

Los estudiantes pueden crear múltiples chats.

Cuenta con los siguientes atributos:

- **id\_chat:** Identifica de forma única cada chat.
- **usuario\_id:** Relaciona cada chat con un usuario.
- **fecha\_inicio:** Registra la fecha y hora en que se inició el chat. Esto para una mayor organización en el historial.

Ejemplo:

id_chat	usuario_id	fecha_inicio
1	1	2025-03-10 11:31:10
2	1	2025-03-10 12:05:09

*Tabla 2. Chats*

### ***Mensaje:***

La tabla Mensajes almacena cada mensaje enviado dentro de un chat.

- **Id\_Mensaje:** Identifica cada mensaje.
- **Chat\_id:** Relaciona cada mensaje con su respectivo chat.
- **Remitente:** Indica si el mensaje fue enviado por un estudiante, administrador o IA.
- **mensaje:** Contenido del mensaje.
- **fecha\_envio:** Almacena la fecha y hora del mensaje.

id_Mensaje	Chat_id	Remitente	mensaje	fecha_envio
1	1	usuario	Holaaa	2025-03-10 11:31:10
2	1	IA	Hola	2025-03-10 11:31:14

*Tabla 3. Mensaje*

### ***Procesos:***

La tabla procesos almacena los nombres de los distintos trámites dentro del sistema.

- **ID\_Procesos:** Identifica de forma única cada proceso.
- **nombre:** Se usa UNIQUE para evitar procesos duplicados.

ID_Procesos	nombre
1	Credenciales
2	Constancias

*Tabla 4. Procesos*

### ***Base\_Conocimiento:***

Almacena la información para las respuestas de la IA:

- **ID\_conocimiento:** Identifica de forma única cada entrada de conocimiento.
- **proceso\_id:** Relaciona la información con un proceso.
- **consulta:** Un título corto para identificar la información almacenada.
- **informacion:** Información para la respuesta de la IA.
- **embedding:** Representación numérica para búsquedas semánticas eficientes.

ID_conocimiento	proceso_id	consulta	informacion	embedding
1	1	Vencimiento	Info	
2	1	Extravio	Info	

*Tabla 6. Base de Conocimientos*

### ***Help\_Request:***

Registra solicitudes de ayuda cuando la IA no puede responder.

- **Id\_help:** Identifica de forma única cada solicitud.
- **chat\_id:** Relaciona la solicitud con un chat.

- **estado:** Indica si la solicitud está "pendiente" o "resuelta"
- **Fecha:** Fecha y hora de la solicitud

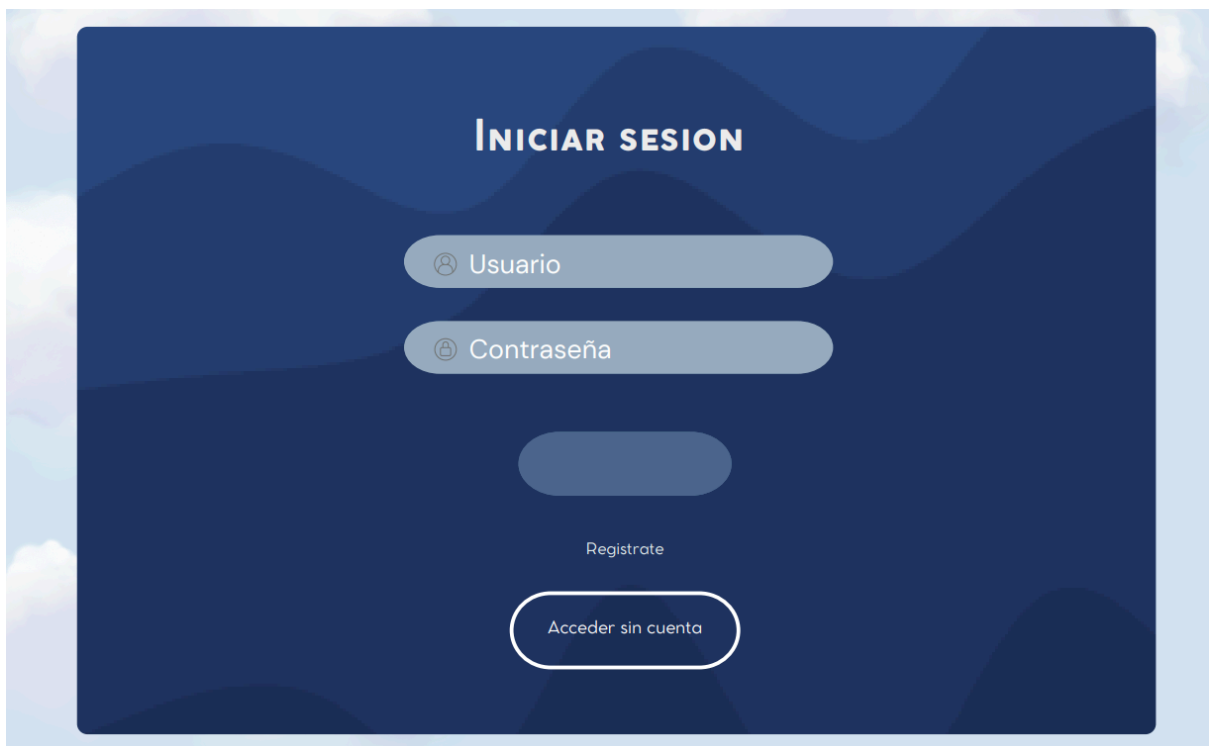
Id_help	chat_id	estado	Fecha
1	1	Pendiente	2025-03-10 11:39:10

*Tabla 7. Help Request*


## 4.- DISEÑO DE INTERFAZ DE USUARIO


### 4.1.- Mockups

Vista en web






# REGISTRATE


 Usuario

 Contraseña


Iniciar Sesión




08 MARZO 2025




Hola! En que puedo ayudarte?



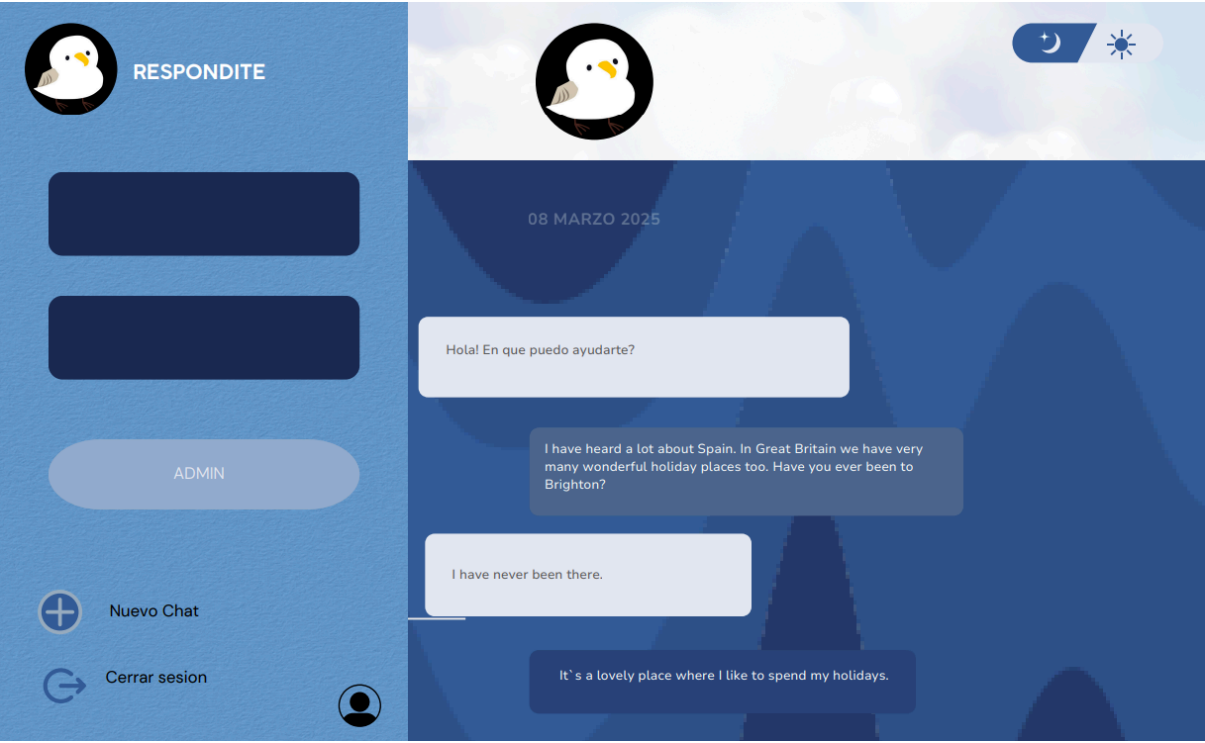
I have never been there.



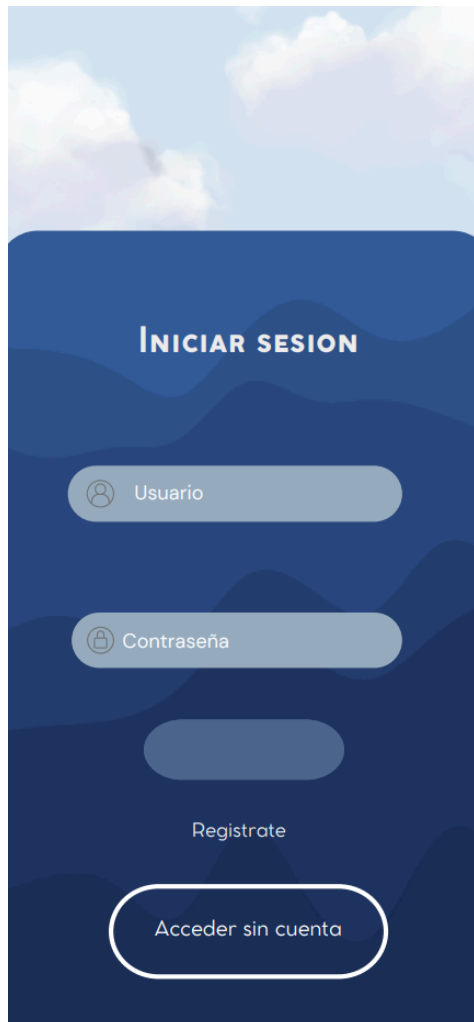
I have heard a lot about Spain. In Great Britain we have very many wonderful holiday places too. Have you ever been to Brighton?



It's a lovely place where I like to spend my holidays.



## Vista móvil



Mobile login screen with a dark blue background and a light blue sky header. The title "INICIAR SESION" is centered. Below it are two input fields: "Usuario" and "Contraseña", each with a user icon. A "Registrate" button is below the password field, and an "Acceder sin cuenta" button is at the bottom.

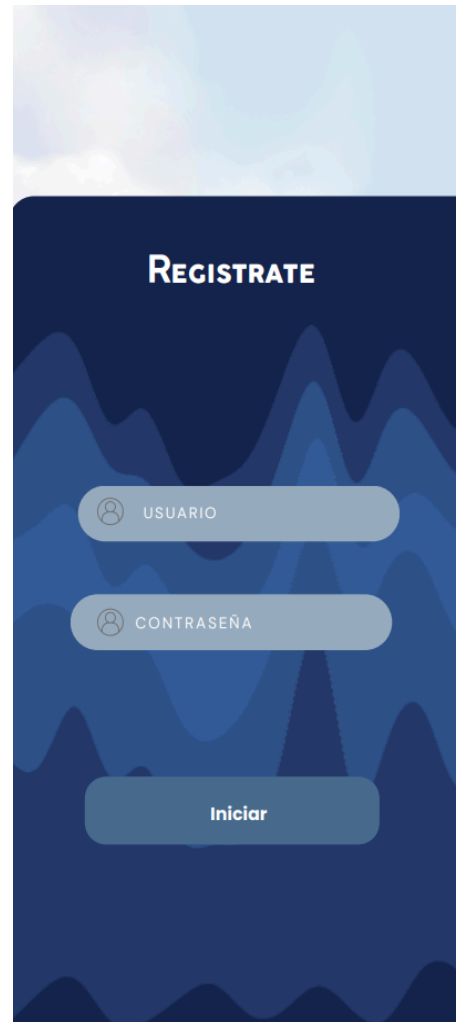
**INICIAR SESION**

Usuario

Contraseña

Registrate

Acceder sin cuenta



Mobile registration screen with a dark blue background and a light blue sky header. The title "REGISTRATE" is centered. Below it are two input fields: "USUARIO" and "CONTRASEÑA", each with a user icon. An "Iniciar" button is below the password field.

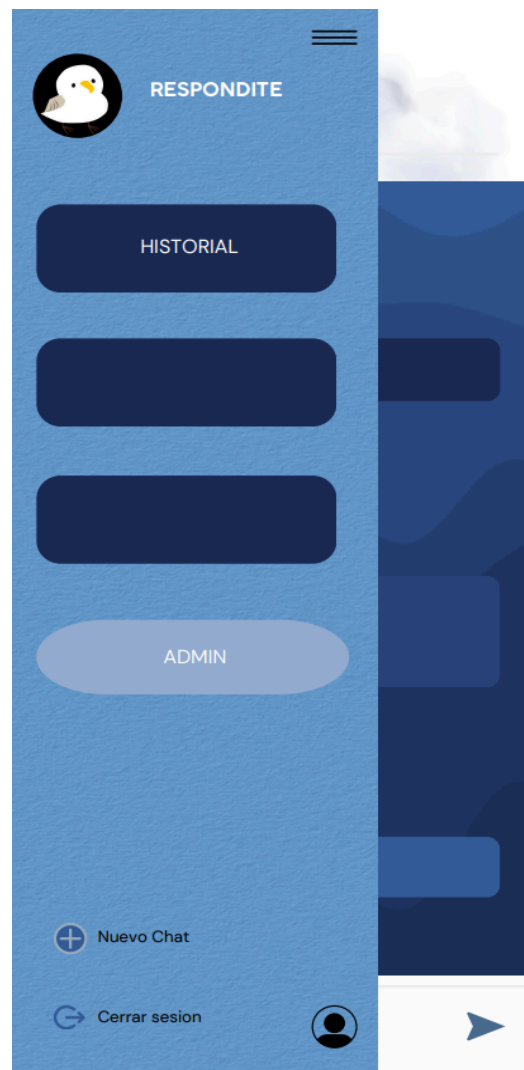
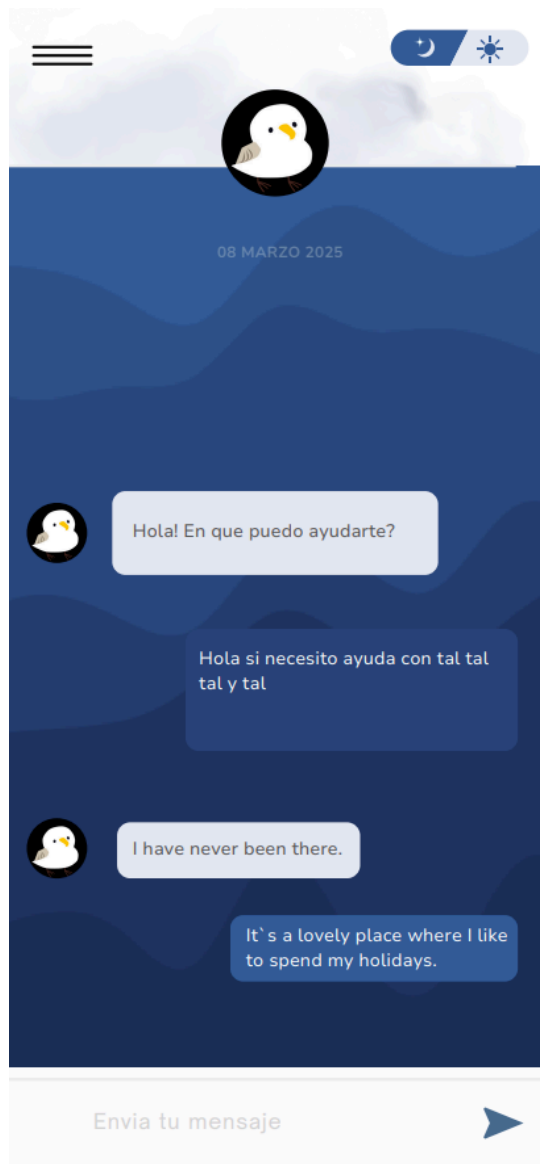
**REGISTRATE**

USUARIO

CONTRASEÑA

Iniciar





INFORMACIÓN  
PARA LA BASE DE  
DATOS

Añadir  
proceso

Eliminar  
proceso

Credenciales

Consulta	Respuesta

Modificar  
Información

Modificar  
Información

Constancias

Consulta	Respuesta

Modificar  
Información

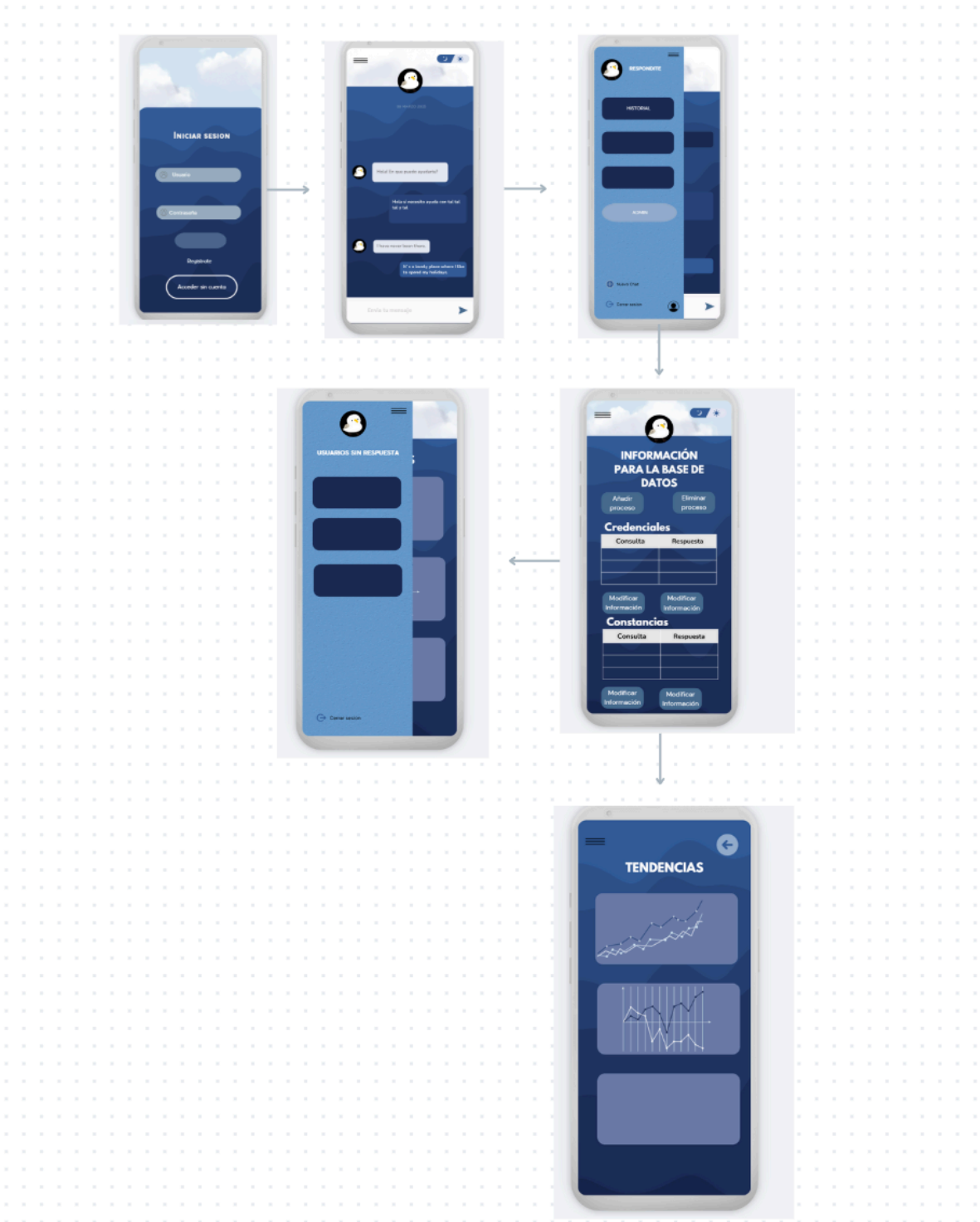
Modificar  
Información

- 4.2.- Diseño de navegación entre pantallas.

### Estudiante(Cliente)



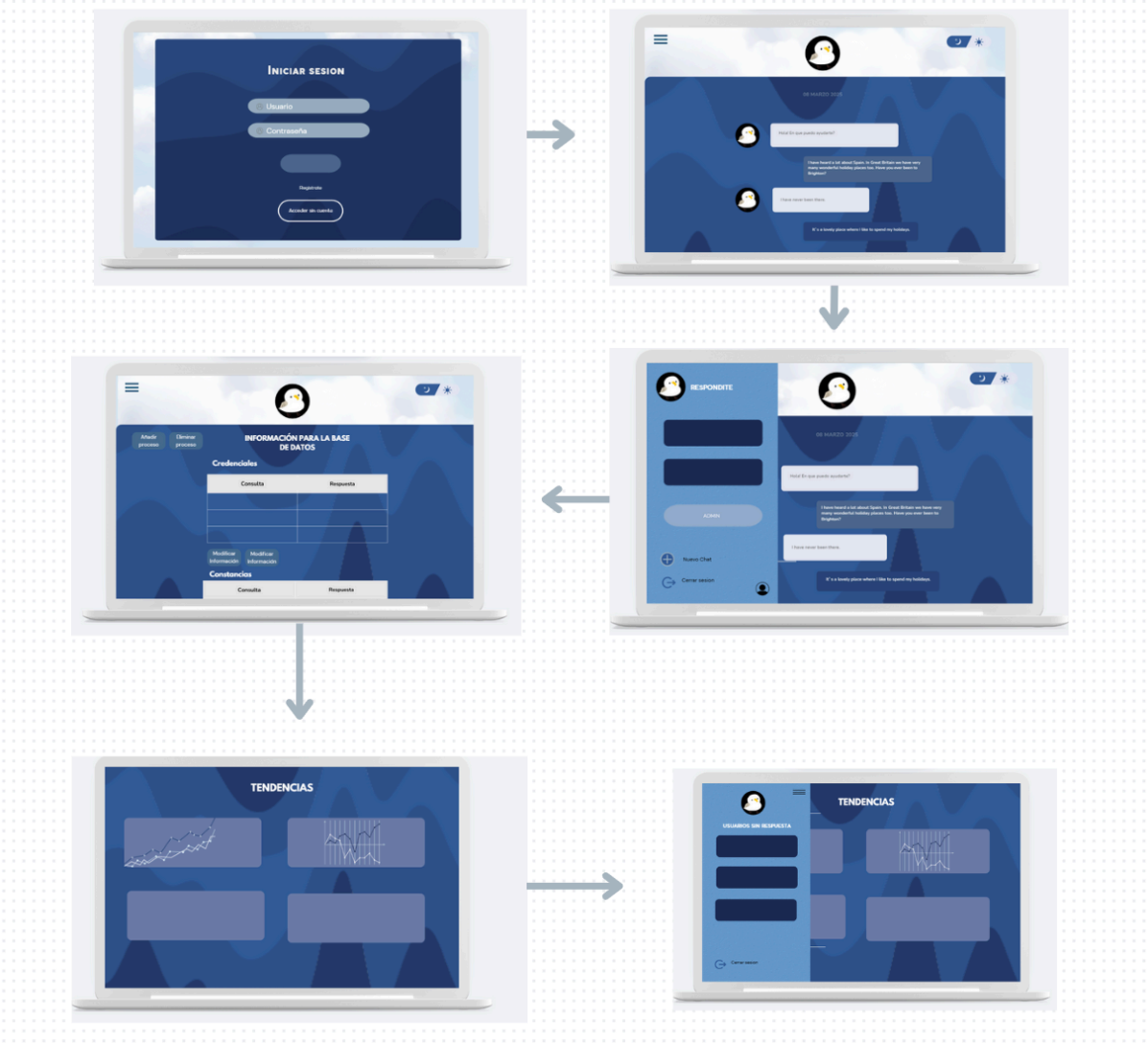
Administrador



## Cliente



Administrador



#### 4.3.- Consideraciones de usabilidad y experiencia de usuario (UX/UI).

##### Simplicidad y Claridad en la Interfaz (UX)

##### UX

- Evitar elementos innecesarios
- Inicio de sesión clara
- Acceso sin cuenta
- Menú intuitivo (nuevo chat, cerrar sesión)
- Botones rápidos (agregar, modificar, eliminar información de la sección de administrador)
- Botones de navegación
- interfaz responsiva
- Compatible en web y móvil
- Compatible en teclado y navegaciones sin mouse
- El usuario puede identificar quién está enviando el mensaje por el avatar del Chat IA
- Menú lateral desplegable

##### UI

- Secciones diferenciadas en la interfaz de administración
- Distribución de texto y botones
- Diseño minimalista
- Opción de modo oscuro
- Colores de la institución
- indicadores de carga (para representar que la IA está buscando la respuesta)
- Diseño responsivo
- Botón de enviar mensaje
- Historial visible

## REFERENCIAS:

<https://rockcontent.com/es/blog/ui-ux/>

<https://www.lucidchart.com/pages/es/que-es-el-lenguaje-unificado-de-modelado-uml>

<https://www.lucidchart.com/pages/es/que-es-el-lenguaje-unificado-de-modelado-uml>

<https://www.gluo.mx/blog/arquitectura-de-software-que-es-y-que-tipos-hay>