



Trabajo final de Grado  
Gustavo González López-  
Casero  
I.E.S. Juan Bosco

Nombre de la aplicación: ChatsApp

Año de creación: 2023

Asignatura: Desarrollo de aplicaciones web

## *Agradecimientos especiales, a mi familia.*

© 2023 Gustavo González López-Casero. Todos los derechos reservados.

El contenido y diseño de esta aplicación web, incluyendo pero no limitado a texto, gráficos, logotipos, imágenes y código fuente, están protegidos por las leyes de derechos de autor y otras leyes de propiedad intelectual. Queda prohibida la reproducción, distribución, modificación o cualquier otro uso no autorizado del contenido sin el permiso previo por escrito del propietario del copyright.

Esta aplicación web puede contener marcas comerciales, nombres comerciales u otros nombres de productos o empresas de terceros, que son propiedad de sus respectivos propietarios y están protegidos por ley.

Se concede permiso para utilizar y acceder a esta aplicación web de acuerdo con los términos y condiciones establecidos en los documentos legales y acuerdos aplicables, siempre y cuando se respeten los derechos de autor y se cumplan las restricciones de uso.

Cualquier uso no autorizado de esta aplicación web o su contenido puede violar las leyes de derechos de autor, marcas comerciales y otras leyes aplicables, y puede resultar en acciones legales y responsabilidad.

Para obtener más información o solicitar permisos para utilizar el contenido de esta aplicación web, por favor contáctanos en [gustavobasauri1996@gmail.com](mailto:gustavobasauri1996@gmail.com)

# Índice

I.	Introducción .....	6
II.	Arquitectura .....	7
II.I.	Backend .....	8
Tecnologías utilizadas.....	8	
Estructura general .....	8	
Interacción con la base de datos.....	8	
Seguridad.....	8	
Diagrama de Clases .....	8	
Estructura de directorios del backend .....	9	
Carpeta raíz .....	9	
Carpeta controlador .....	10	
Carpeta modelo.....	10	
Carpeta vistas .....	11	
Carpeta utilidades .....	11	
II.II.	Frontend.....	12
Tecnologías utilizadas.....	12	
Estructura general .....	12	
Interacción con el backend .....	12	
Diseño y estilización .....	12	
Diagramas de flujo de proceso.....	12	
<b>Diagrama de flujo de login</b> .....	13	
<b>Diagrama de flujo de registro</b> .....	14	
<b>Diagramas de flujo de la vista de mensajes</b> .....	15	
<i>Cambiar foto de perfil</i> .....	15	
<i>Lista de amigos – Seleccionar chat</i> .....	16	
<i>Envío de mensajes</i> .....	16	
<i>Edición de cuenta de usuario</i> .....	17	
III.	Requisitos .....	18
Requisitos Funcionales .....	18	
Requisitos No Funcionales .....	18	
IV.	Diseño del Sistema .....	19
Diagrama E-R.....	19	
Resumen general.....	19	

Tabla “Usuarios” .....	20
Tabla “Contactos” .....	21
Tabla “mensajes” .....	22
Tabla “imagenes” .....	23
V. Interfaz de usuario .....	24
Paleta de diseño .....	24
Colores principales .....	24
Color principal .....	24
Color secundario .....	24
Color terciario.....	24
Colores de acento.....	25
Mensajes leídos.....	25
Mensajes de error o mensajes al usuario en general .....	25
Iconos de borrado .....	25
Estados de interacción .....	25
Hover general.....	25
Hover de eliminar usuario .....	25
Hover de añadir usuario .....	25
Vista de inicio .....	26
Caja de Login - Formulario de login.....	27
Caja Mensaje – Registro exitoso .....	27
Caja Mensaje – Error de Login.....	28
Interacción del usuario con la vista.....	28
Vista de registro .....	29
Caja de Registro.....	30
Caja Mensaje – Error de Registro.....	30
Interacción del usuario con la vista.....	31
Vista de mensajes.....	32
Estructura General .....	34
Interacción del usuario con la vista.....	34
Componente – foto perfil.....	36
Componente – Lista de Mensajes & selección de chat.....	37
Componente – Envío de Mensajes & cajas de mensajes .....	38
Componente – Agregar amigos o “contactos” .....	39
Componente – Contenido dinámico “Configuración” .....	39
Componente – “Mis contactos” .....	40
VI. Mantenimiento y evolución .....	41

VII.	Planificación .....	42
	Análisis y Diseño (10 horas) .....	42
	Backend (30 horas).....	42
	Frontend (20 horas) .....	42
VIII.	Bibliografía .....	43
	Comunidades y foros en línea .....	43
	Sitios de desarrollo web .....	43
	Documentación oficial.....	43

# I. Introducción

La aplicación web desarrollada como parte de este Trabajo de Fin de Grado tiene como objetivo principal el rediseño de la conocida plataforma de mensajería WhatsApp, con el propósito de ofrecer una interfaz más moderna, estilizada y mejorar la experiencia del usuario.

Además de buscar una renovación visual, este proyecto también tiene como finalidad servir como carta de presentación del desarrollador, demostrando sus habilidades y conocimientos en el ámbito del desarrollo de aplicaciones web.

La aplicación web clon de WhatsApp tiene como meta inicial ofrecer una experiencia de usuario optimizada, centrándose en el diseño y la usabilidad. A medida que el proyecto avance, se busca también ampliar las funcionalidades para proporcionar una plataforma más completa, manteniendo siempre la calidad y el enfoque estético.

El presente documento tiene como propósito guiar el proceso de desarrollo y documentar los elementos clave de la aplicación, desde su arquitectura hasta su implementación. A través de la descripción detallada de los componentes y funcionalidades, se busca brindar una visión integral del proyecto y facilitar su comprensión y reproducción en futuras iteraciones.

## II. Arquitectura

La aplicación web ChatsApp tiene una arquitectura que combina backend y frontend para ofrecer una experiencia de usuario completa. A continuación, se detallan los aspectos clave de cada una de estas capas.

## II.I. Backend

### Tecnologías utilizadas

El backend de la aplicación se desarrolla utilizando PHP, SQL y peticiones asíncronas con AJAX.

### Estructura general

La estructura del backend se basa en un enfoque de mapeo de acciones en el archivo **index.php**. Las diferentes acciones, como registro, inicio de sesión y mensajes, están mapeadas como rutas dentro de este archivo.

### Interacción con la base de datos

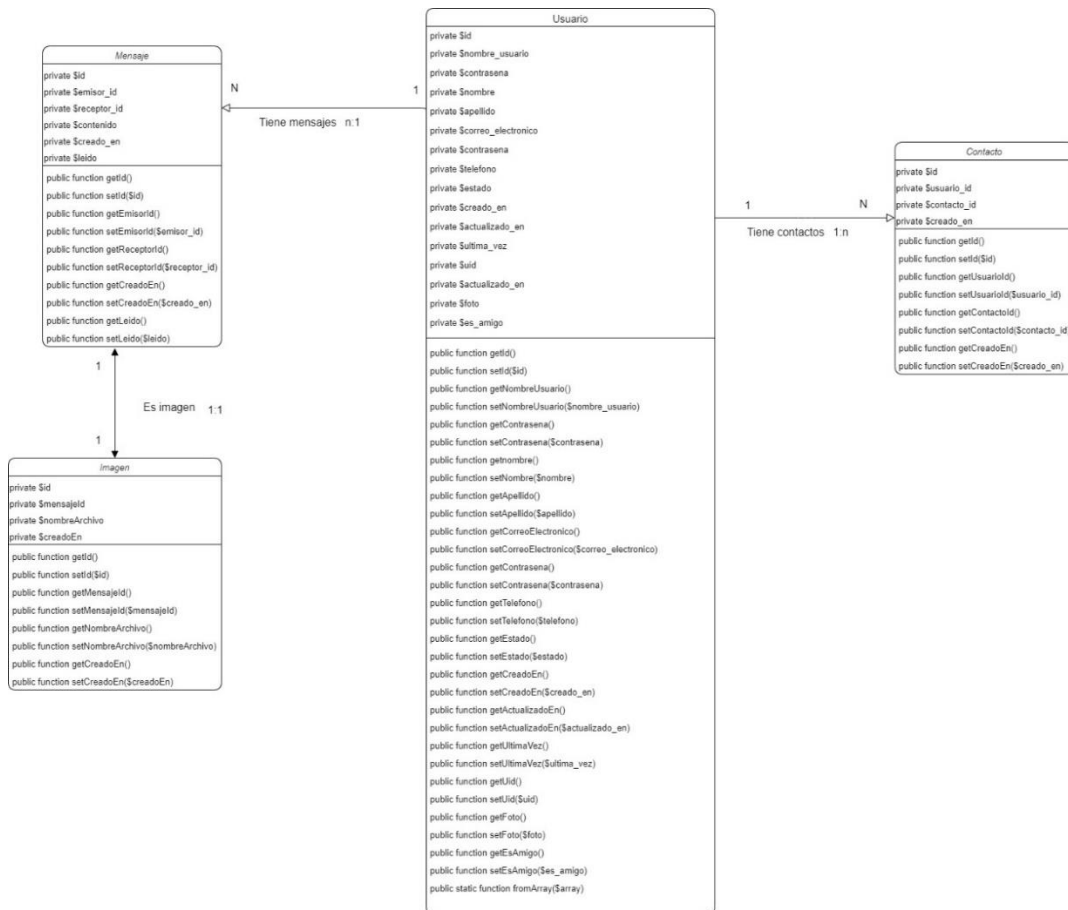
Para comunicarse con la base de datos SQL, se utilizan consultas SQL directas. Las consultas se encargan de la obtención y manipulación de datos necesarios para las funcionalidades de la aplicación.

### Seguridad

El backend implementa medidas de seguridad para proteger los datos y la integridad del sistema. Algunas de estas medidas incluyen el uso de funciones como **password\_hash** para encriptar las contraseñas de los usuarios y la validación de datos de entrada para prevenir ataques de inyección SQL.

### Diagrama de Clases











## Estructura de directorios del backend

### Carpeta raíz

Ubicación → ChatsApp\app







Archivo de configuración → **config.php**

Este archivo establece variables globales para la conexión con la base de datos y otras configuraciones relevantes para el backend.

 <b>app</b>	10/06/2023 19:07	Carpeta de archivos
 <b>controlador</b>	10/06/2023 19:07	Carpeta de archivos
 <b>modelo</b>	11/06/2023 11:52	Carpeta de archivos
 <b>utilidades</b>	10/06/2023 19:07	Carpeta de archivos
 <b>vistas</b>	10/06/2023 19:07	Carpeta de archivos
 <b>config</b>	01/05/2023 17:33	Archivo PHP













## Carpeta controlador

En esta carpeta se encuentran los archivos controladores que manejan la lógica de la aplicación y procesan las solicitudes del cliente. Aquí se implementan las funciones y métodos que interactúan con los modelos y devuelven respuestas al cliente.

 app	10/06/2023 19:07	Carpeta de archivos
 controlador	10/06/2023 19:07	Carpeta de archivos
 ContactoController	11/06/2023 16:35	Archivo de origen PHP
 EntornoController	01/06/2023 19:06	Archivo de origen PHP
 MensajesController	10/06/2023 12:35	Archivo de origen PHP
 UsuariosController	10/06/2023 12:38	Archivo de origen PHP






## Carpeta modelo

En esta carpeta se encuentran las clases objetos y las clases DAO (Data Access Object) que ejecutan sentencias a la base de datos. Las clases objetos representan entidades o conceptos de tu aplicación, mientras que las clases DAO manejan la comunicación con la base de datos, realizando consultas, inserciones, actualizaciones, etc.

 app	10/06/2023 19:07	Carpeta de archivos
 modelo	11/06/2023 11:52	Carpeta de archivos
 UsuarioDAO	08/06/2023 11:05	Archivo de origen PHP
 Usuario	07/06/2023 20:31	Archivo de origen PHP
 MensajeDAO	11/06/2023 11:51	Archivo de origen PHP
 Mensaje	10/06/2023 19:19	Archivo de origen PHP
 ImagenDAO	11/06/2023 16:43	Archivo de origen PHP
 Imagen	11/06/2023 18:01	Archivo de origen PHP
 Conversacion	10/06/2023 12:30	Archivo de origen PHP
 ContactoDAO	10/06/2023 16:10	Archivo de origen PHP
 Contacto	03/06/2023 12:02	Archivo de origen PHP
 ConexionBD	01/05/2023 15:56	Archivo de origen PHP




## Carpeta vistas

Esta carpeta puede contener archivos relacionados con las vistas en el backend, como plantillas o archivos de generación de respuestas en formato específico, si se aplica.

 app	10/06/2023 19:07	Carpeta de archivos
 vistas	10/06/2023 19:07	Carpeta de archivos
 inicio	28/05/2023 13:31	Archivo de origen PHP
 registro	30/05/2023 19:54	Archivo de origen PHP
 web_Index	13/06/2023 17:06	Archivo de origen PHP

## Carpeta utilidades

En esta carpeta puedes almacenar utilidades o funciones auxiliares que son utilizadas en diferentes partes del backend. Pueden incluir funciones de validación, funciones de formateo de datos, funciones de manejo de archivos, entre otros.

 app	10/06/2023 19:07	Carpeta de archivos
 utilidades	10/06/2023 19:07	Carpeta de archivos
 MensajeFlash	22/05/2023 14:58	Archivo de origen PHP

## II.II. Frontend

### Tecnologías utilizadas

El frontend de la aplicación se desarrolla utilizando HTML, CSS y JavaScript, con el uso de JQuery para animaciones y efectos adicionales.

### Estructura general

El frontend está organizado en diferentes directorios dentro de la ruta del proyecto. En particular, los archivos CSS se encuentran en **web/css**, divididos en subdirectorios correspondientes a las vistas de "Inicio" y "Mensajes". Los archivos JavaScript se ubican en **web/js** y se llaman en las vistas según sea necesario, para mejorar la modularidad y organización del código.

### Interacción con el backend

El frontend se comunica con el backend para obtener y enviar datos utilizando el método POST en formularios y peticiones. Las acciones del usuario, como registro, inicio de sesión y envío de mensajes, se realizan a través de estas interacciones con el backend.

### Diseño y estilización

El frontend se enfoca en brindar una experiencia de usuario moderna y estilizada. Se utilizan tecnologías como CSS y JQuery para lograr efectos visuales atractivos y animaciones que mejoren la usabilidad y la apariencia general de la aplicación.

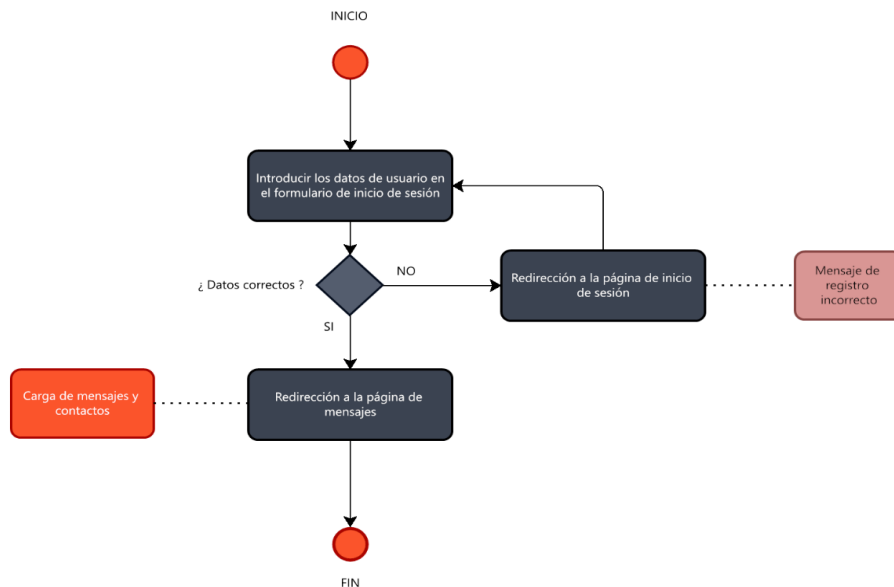
### Diagramas de flujo de proceso

En este apartado se presenta un **diagrama de flujo de alto nivel** que ilustra el proceso de registro de usuario en la aplicación. Este diagrama permite comprender de manera visual y concisa los pasos involucrados en el proceso de registro del usuario

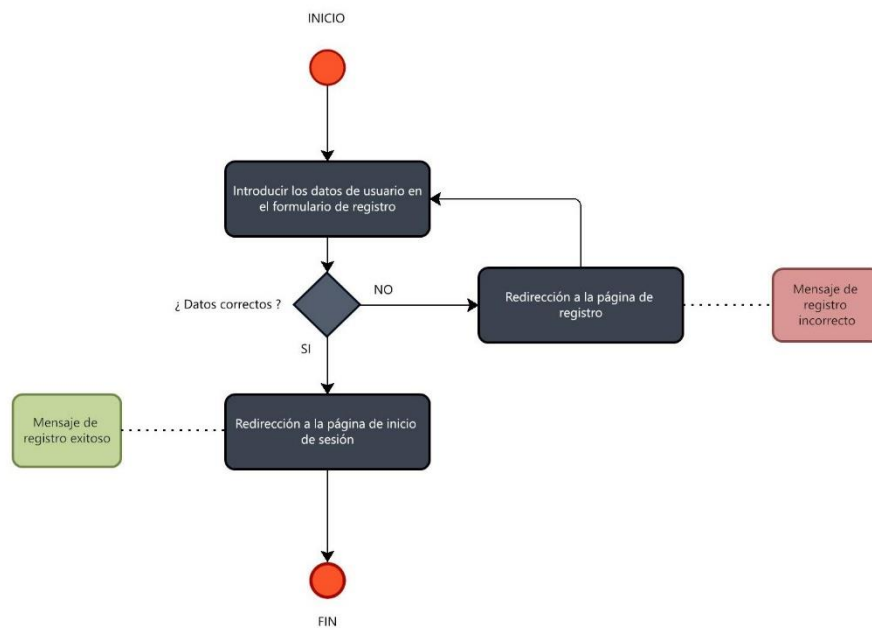


El uso de un diagrama de flujo de alto nivel facilita la comprensión visual del proceso de inicio de sesión, resaltando los pasos clave y las decisiones que se toman en cada etapa. Este enfoque simplificado ayuda a los lectores a entender rápidamente el flujo general y la secuencia de acciones involucradas en el proceso de autenticación

## Diagrama de flujo de login



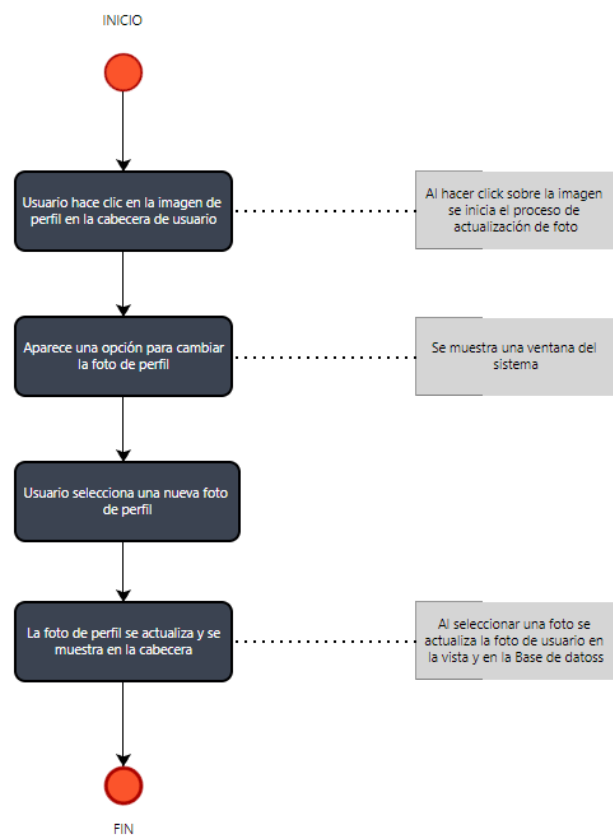
## Diagrama de flujo de registro



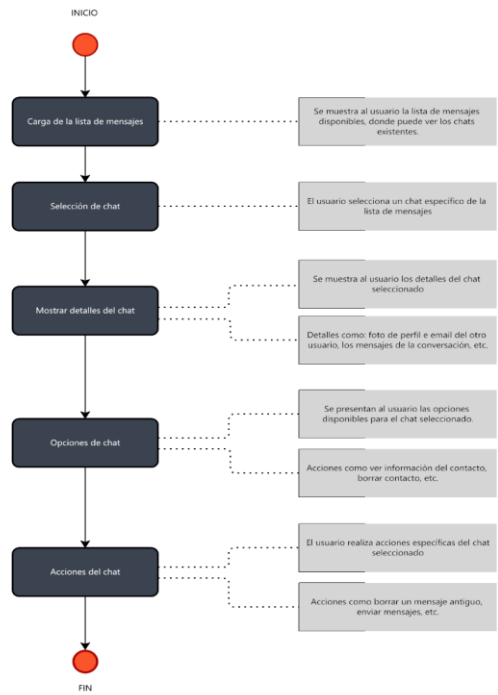
## Diagramas de flujo de la vista de mensajes

Atacaremos en este punto cada una de las principales funcionalidades con las que el usuario puede interactuar.

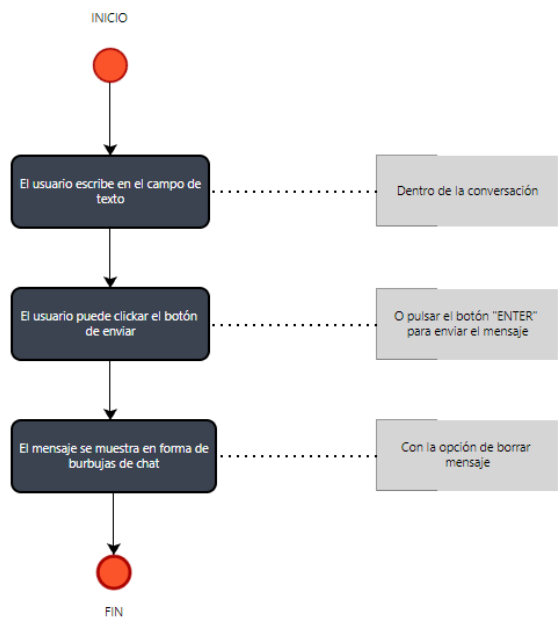
### Cambiar foto de perfil



Lista de amigos – Seleccionar chat

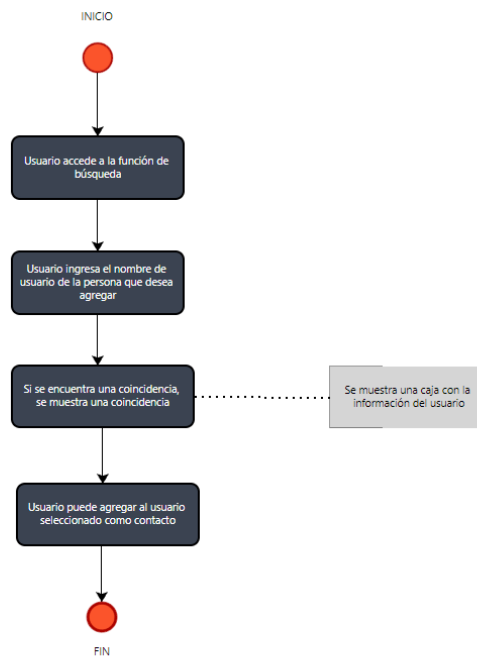


Envío de mensajes

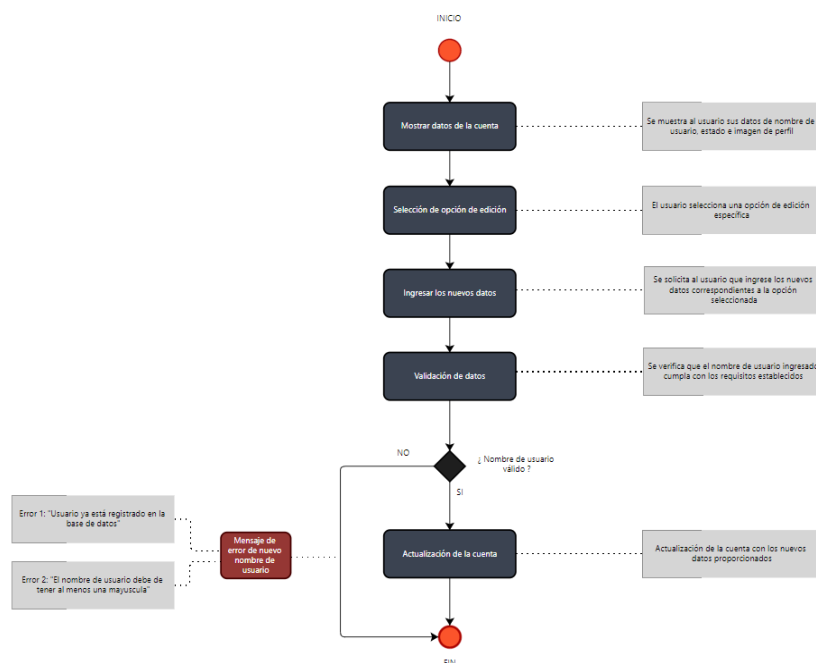




## Agregar a amigos



## Edición de cuenta de usuario



# III. Requisitos

## Requisitos Funcionales

1. **Registro de Usuarios:** Los usuarios deben poder crear una cuenta en la aplicación web proporcionando información como nombre de usuario, contraseña, dirección de correo electrónico y número de teléfono.
  2. **Inicio de Sesión:** Los usuarios registrados deben poder iniciar sesión en la aplicación web utilizando sus credenciales de cuenta.
  3. **Lista de Mensajes:** La aplicación debe mostrar una lista de mensajes de los contactos del usuario, ordenados de manera descendente por fecha y hora, de modo que los mensajes más recientes aparezcan primero.
  4. **Envío y Recepción de Mensajes:** Los usuarios deben poder enviar y recibir mensajes de otros usuarios que estén en su lista de contactos.
  5. **Administración de Contactos:** Los usuarios deben poder agregar y eliminar contactos de su lista de contactos
  6. **Actualización de Perfil:** Los usuarios deben poder cambiar su foto de perfil, estado y nombre de usuario en la aplicación.
- 

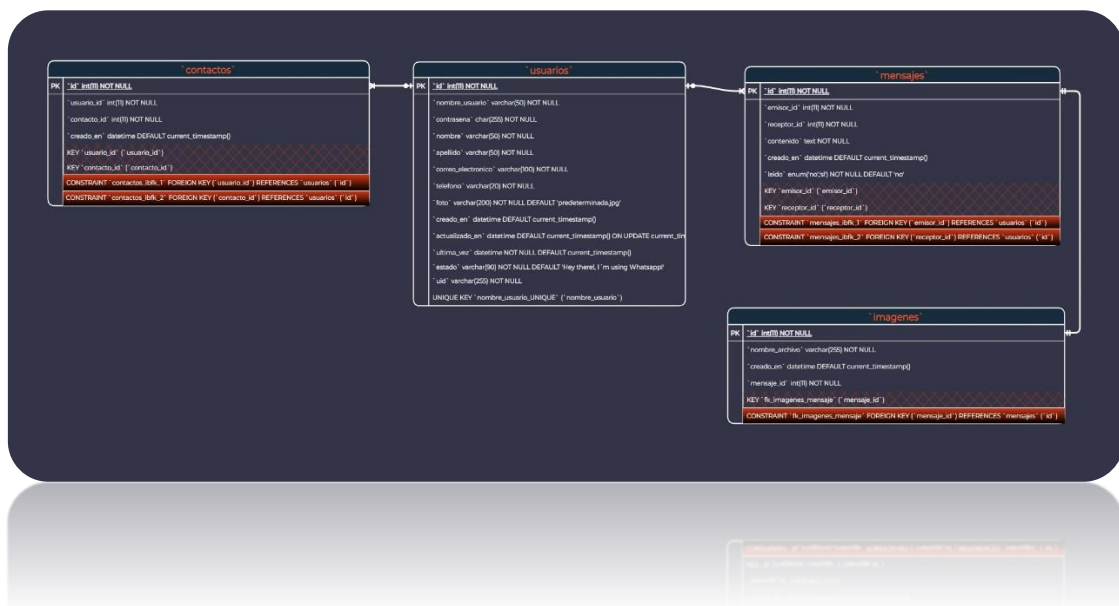
## Requisitos No Funcionales

1. **Seguridad:** La aplicación debe implementar medidas de seguridad para proteger la privacidad de los usuarios y prevenir accesos no autorizados.
  2. **Rendimiento:** La aplicación debe ser capaz de manejar un alto volumen de usuarios y proporcionar una respuesta rápida en todas las funcionalidades.
  3. **Compatibilidad:** La aplicación debe ser compatible con los navegadores web más populares.
  4. **Almacenamiento:** La aplicación debe gestionar adecuadamente el almacenamiento de los mensajes y los datos de los usuarios, cumpliendo con los límites de almacenamiento establecidos.
-

# IV. Diseño del Sistema

En este apartado se detalla la estructura de las tablas y las relaciones entre ellas:

## Diagrama E-R



## Resumen general

La base de datos del proyecto "ChatsApp" es un sistema de mensajería instantánea que permite a los usuarios enviar mensajes, compartir imágenes y gestionar su lista de contactos. La base de datos está diseñada para almacenar la información de los usuarios, sus contactos, los mensajes enviados y las imágenes compartidas en la plataforma.



El diseño de la base de datos utiliza claves foráneas para establecer las relaciones entre las tablas y garantizar la integridad referencial. Las restricciones y los índices se han definido adecuadamente para mantener la consistencia y optimizar el rendimiento de las consultas.

## Tabla “Usuarios”

La tabla "usuarios" almacena la información de los usuarios registrados en la aplicación. Contiene los siguientes campos:

- **id**: identificador único del usuario (clave primaria).
- **nombre\_usuario**: nombre de usuario único para identificar al usuario.
- **contrasena**: contraseña del usuario.
- **nombre**: nombre del usuario.
- **apellido**: apellido del usuario.
- **correo\_electronico**: dirección de correo electrónico del usuario.
- **telefono**: número de teléfono del usuario.
- **foto**: nombre del archivo de imagen asociado al usuario.
- **creado\_en**: fecha y hora de creación del usuario.
- **actualizado\_en**: fecha y hora de la última actualización del usuario.
- **ultima\_vez**: fecha y hora de la última vez que el usuario inició sesión.
- **estado**: estado o mensaje personalizado del usuario.
- **uid**: identificador único para la autenticación del usuario.



`usuarios`	
PK	`id` int(11) NOT NULL
	`nombre_usuario` varchar(50) NOT NULL
	`contrasena` char(255) NOT NULL
	`nombre` varchar(50) NOT NULL
	`apellido` varchar(50) NOT NULL
	`correo_electronico` varchar(100) NOT NULL
	`telefono` varchar(20) NOT NULL
	`foto` varchar(200) NOT NULL DEFAULT 'predeterminada.jpg'
	`creado_en` datetime DEFAULT current_timestamp()
	`actualizado_en` datetime DEFAULT current_timestamp() ON UPDATE current_timestamp()
	`ultima_vez` datetime NOT NULL DEFAULT current_timestamp()
	`estado` varchar(90) NOT NULL DEFAULT 'Hey there!, I´m using Whatsapp!'
	`uid` varchar(255) NOT NULL
	UNIQUE KEY `nombre_usuario_UNIQUE` (`nombre_usuario`)

## Tabla “Contactos”

La tabla "contactos" registra las relaciones de contacto entre los usuarios. Contiene los siguientes campos:

- **id**: identificador único del contacto (clave primaria).
- **usuario\_id**: identificador del usuario que agrega el contacto (clave externa que referencia a la tabla "usuarios").
- **contacto\_id**: identificador del usuario que es agregado como contacto (clave externa que referencia a la tabla "usuarios").
- **creado\_en**: fecha y hora en la que se creó la relación de contacto.

La relación entre la tabla "contactos" y la tabla "usuarios" es de muchos a uno (n:1), lo que significa que un usuario puede tener varios contactos y un contacto puede estar asociado a un usuario.



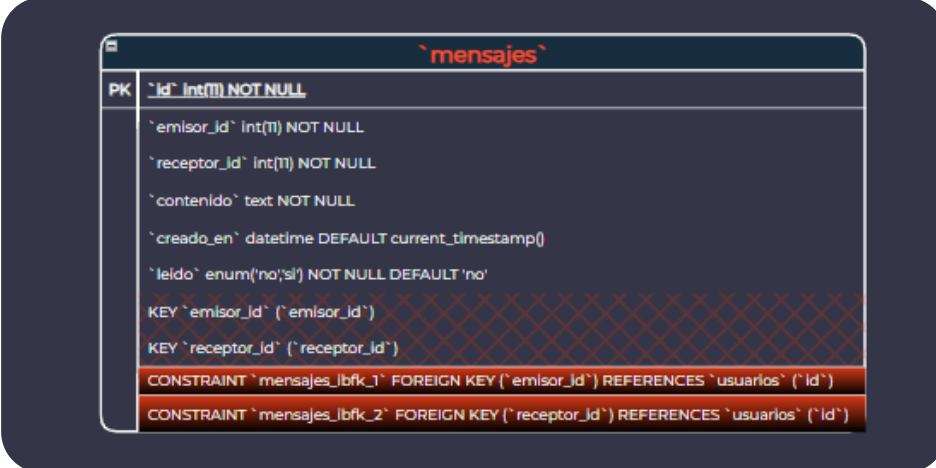
	<code>`contactos`</code>
PK	<code>`id` int(11) NOT NULL</code>
	<code>`usuario_id` int(11) NOT NULL</code>
	<code>`contacto_id` int(11) NOT NULL</code>
	<code>`creado_en` datetime DEFAULT current_timestamp()</code>
	<code>KEY `usuario_id` (`usuario_id`)</code>
	<code>KEY `contacto_id` (`contacto_id`)</code>
	<code>CONSTRAINT `contactos_ibfk_1` FOREIGN KEY (`usuario_id`) REFERENCES `usuarios` (`id`)</code>
	<code>CONSTRAINT `contactos_ibfk_2` FOREIGN KEY (`contacto_id`) REFERENCES `usuarios` (`id`)</code>

## Tabla “mensajes”

La tabla "mensajes" almacena los mensajes enviados entre los usuarios. Contiene los siguientes campos:

- **id**: identificador único del mensaje (clave primaria).
- **emisor\_id**: identificador del usuario que envía el mensaje (clave externa que referencia a la tabla "usuarios").
- **receptor\_id**: identificador del usuario que recibe el mensaje (clave externa que referencia a la tabla "usuarios").
- **contenido**: contenido del mensaje.
- **creado\_en**: fecha y hora de creación del mensaje.
- **leido**: indicador de lectura del mensaje (valor "si" o "no").

La relación entre la tabla "mensajes" y la tabla "usuarios" es de uno a muchos (1:n), ya que un usuario puede enviar varios mensajes, pero cada mensaje tiene un único emisor y receptor.



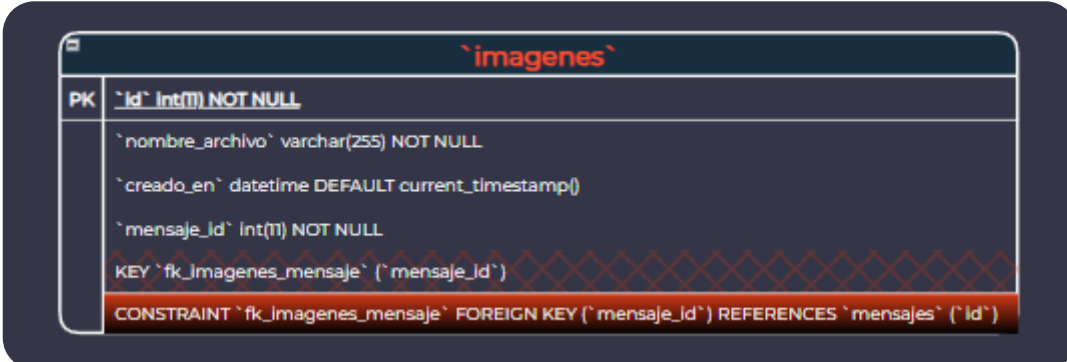
`mensajes`	
PK	<code>`id` int(11) NOT NULL</code>
	<code>`emisor_id` int(11) NOT NULL</code>
	<code>`receptor_id` int(11) NOT NULL</code>
	<code>`contenido` text NOT NULL</code>
	<code>`creado_en` datetime DEFAULT current_timestamp()</code>
	<code>`leido` enum('no','si') NOT NULL DEFAULT 'no'</code>
	<code>KEY `emisor_id` (`emisor_id`)</code>
	<code>KEY `receptor_id` (`receptor_id`)</code>
	<code>CONSTRAINT `mensajes_ibfk_1` FOREIGN KEY (`emisor_id`) REFERENCES `usuarios` (`id`)</code>
	<code>CONSTRAINT `mensajes_ibfk_2` FOREIGN KEY (`receptor_id`) REFERENCES `usuarios` (`id`)</code>

## Tabla “imagenes”

La tabla "imagenes" registra las imágenes compartidas en los mensajes. Contiene los siguientes campos:

- **id**: identificador único de la imagen (clave primaria).
- **nombre\_archivo**: nombre del archivo de imagen.
- **creado\_en**: fecha y hora de creación de la imagen.
- **mensaje\_id**: identificador del mensaje al que pertenece la imagen (clave externa que referencia a la tabla "mensajes").

La relación entre la tabla "imagenes" y la tabla "mensajes" es de uno a uno (1:1), lo que significa que una imagen está asociada a un único mensaje y cada mensaje puede tener una única imagen.



`imagenes`	
PK	<code>`id` int(11) NOT NULL</code>
	<code>`nombre_archivo` varchar(255) NOT NULL</code>
	<code>`creado_en` datetime DEFAULT current_timestamp()</code>
	<code>`mensaje_id` int(11) NOT NULL</code>
	<code>KEY `fk_imagenes_mensaje` (`mensaje_id`)</code>
	<code>CONSTRAINT `fk_imagenes_mensaje` FOREIGN KEY (`mensaje_id`) REFERENCES `mensajes` (`id`)</code>

# V. Interfaz de usuario


En esta sección, se detallarán los componentes clave de la interfaz y se proporcionarán capturas de pantalla que ilustren cómo se ven y cómo interactúa el usuario con ellos.

El objetivo principal es mostrar de manera clara y visual los distintos elementos de la interfaz, como formularios, botones, mensajes de error o éxito, barras de navegación, etc.

## Paleta de diseño


### Colores principales

#### *Color principal*

#343546 

Color utilizado como fondo principal en la interfaz de usuario, proporcionando una sensación de elegancia y sobriedad.

#### *Color secundario*

#FB542B 

Color llamativo y vibrante utilizado para resaltar elementos importantes y llamar la atención del usuario.

#### *Color terciario*


#A0A1B2 

Color utilizado para elementos secundarios y para proporcionar un contraste sutil con el color principal.



## Colores de acento

### *Mensajes leídos*

#00d99a 


Color verde brillante utilizado para resaltar mensajes leídos y proporcionar una indicación visual de su estado.

### *Mensajes de error o mensajes al usuario en general*

#FF0000 

Color rojo intenso utilizado para resaltar mensajes de error o mensajes importantes dirigidos al usuario.


### *Iconos de borrado*

#FF0000 

Color rojo utilizado para resaltar los iconos de borrado y transmitir visualmente su función de eliminación.

## Estados de interacción

### *Hover general*

#FB542B 


Color utilizado para resaltar elementos cuando el usuario pasa el cursor sobre ellos, brindando una retroalimentación visual clara y enfatizando la interactividad.

### *Hover de eliminar usuario*

darkred 

Color rojo oscuro utilizado para resaltar la opción de eliminar un usuario, transmitiendo visualmente una acción de alto impacto y irreversible.

### *Hover de añadir usuario*

rgb(6, 173, 6) 

(descripción: Color verde intenso utilizado para resaltar la opción de añadir un usuario, transmitiendo visualmente una acción positiva y de éxito).

## Vista de inicio

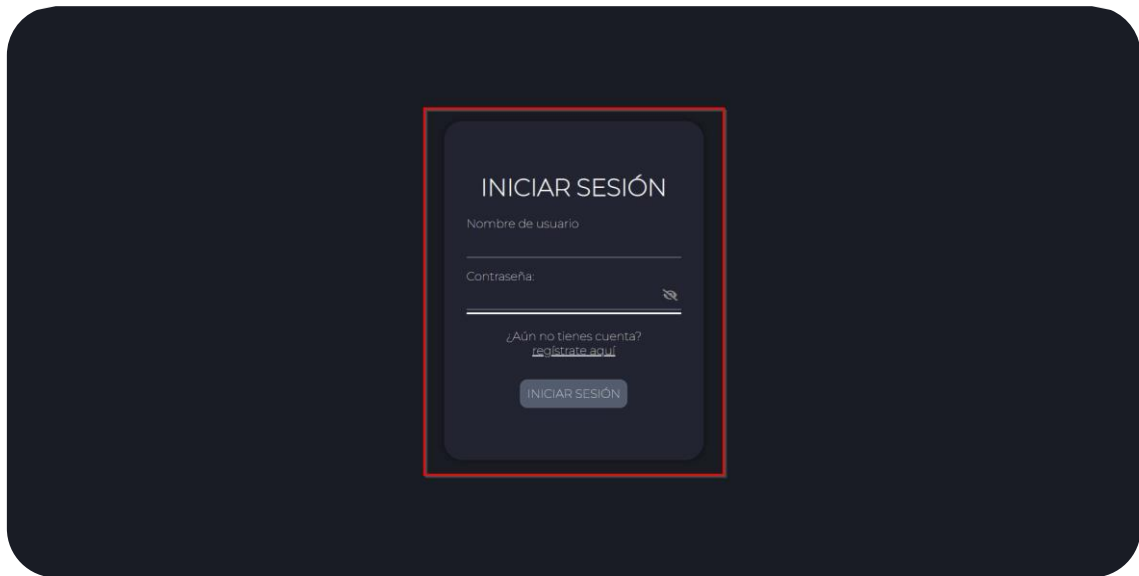
En la vista de Login, podemos identificar dos componentes principales: la caja del formulario de login y la caja de mensajes.

1. La **caja del formulario** de login es donde los usuarios ingresan su información de inicio de sesión, usando su nombre de usuario y contraseña. Aquí se encuentran los campos correspondientes y los botones para enviar el formulario.
2. La **caja de mensajes** es un componente que se utiliza para mostrar mensajes de error o éxito relacionados con el proceso de login. Por ejemplo, si se ingresan credenciales incorrectas, se mostrará un mensaje de error. Por otro lado, si el usuario se registra exitosamente, se mostrará un mensaje de éxito.

Estos componentes son importantes para brindar una experiencia de usuario clara y significativa, ya que permiten al usuario interactuar con la aplicación de manera efectiva y recibir retroalimentación sobre sus acciones.

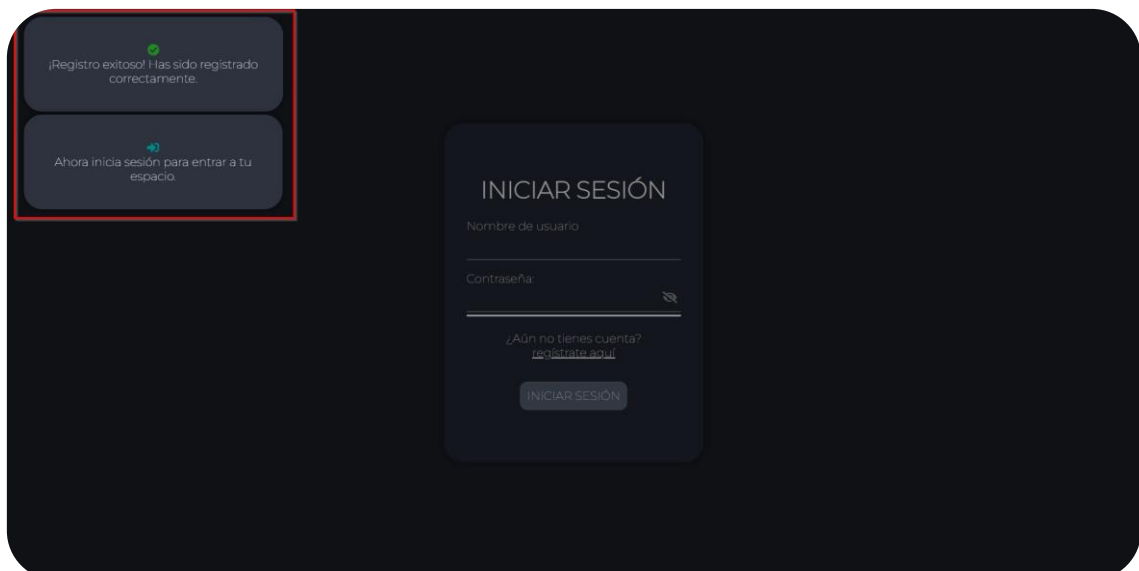
---

## Caja de Login - Formulario de login



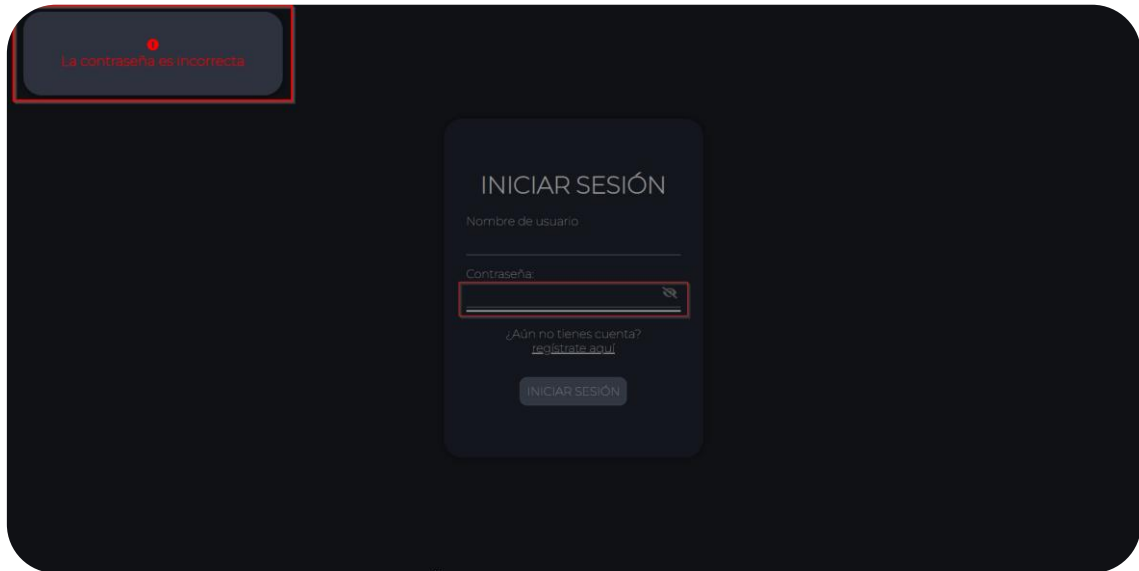
A dark-themed login form titled "INICIAR SESIÓN" is centered on a dark blue background. The form is enclosed in a red rectangular border. It contains two input fields: "Nombre de usuario" and "Contraseña". The "Contraseña" field has a small eye icon to its right. Below the fields, there is a link that says "¿Aún no tienes cuenta? [regístrate aquí](#)". At the bottom of the form is a button labeled "INICIAR SESIÓN".

## Caja Mensaje Login – Registro exitoso



The same dark-themed login form is shown, but with an additional message box on the left side, highlighted by a red rectangular border. The message box contains two lines of text: "¡Registro exitoso! Has sido registrado correctamente." with a green checkmark icon, and "Ahora inicia sesión para entrar a tu espacio." with a blue double-headed arrow icon. The login form itself remains unchanged, with the "INICIAR SESIÓN" button and the "¿Aún no tienes cuenta? [regístrate aquí](#)" link.

## Caja Mensaje – Error de Login



## Interacción del usuario con la vista

El usuario podrá acceder a la aplicación a través de la funcionalidad de inicio de sesión. Para ello, se le proporcionarán dos campos en la vista: "Nombre de usuario" y "Contraseña". Estos campos permitirán al usuario ingresar la información correspondiente a su nombre de usuario y contraseña para iniciar sesión en la aplicación.

Cuando el usuario complete los campos y envíe la solicitud de inicio de sesión, la aplicación realizará los siguientes pasos para validar los datos:

1. Recibirá el nombre de usuario y la contraseña ingresados por el usuario.
2. Utilizará el nombre de usuario para buscar en la base de datos el registro asociado a ese usuario.
3. Extraerá la contraseña almacenada en la base de datos para ese usuario.
4. Utilizará la función **password\_verify()** para comparar la contraseña ingresada por el usuario con el hash almacenado en la base de datos.
5. Si la contraseña coincide con el hash almacenado, significa que la contraseña es válida y se permitirá al usuario acceder al sistema.

6. En caso contrario, se mostrará un mensaje de error indicando que las credenciales ingresadas no son correctas.



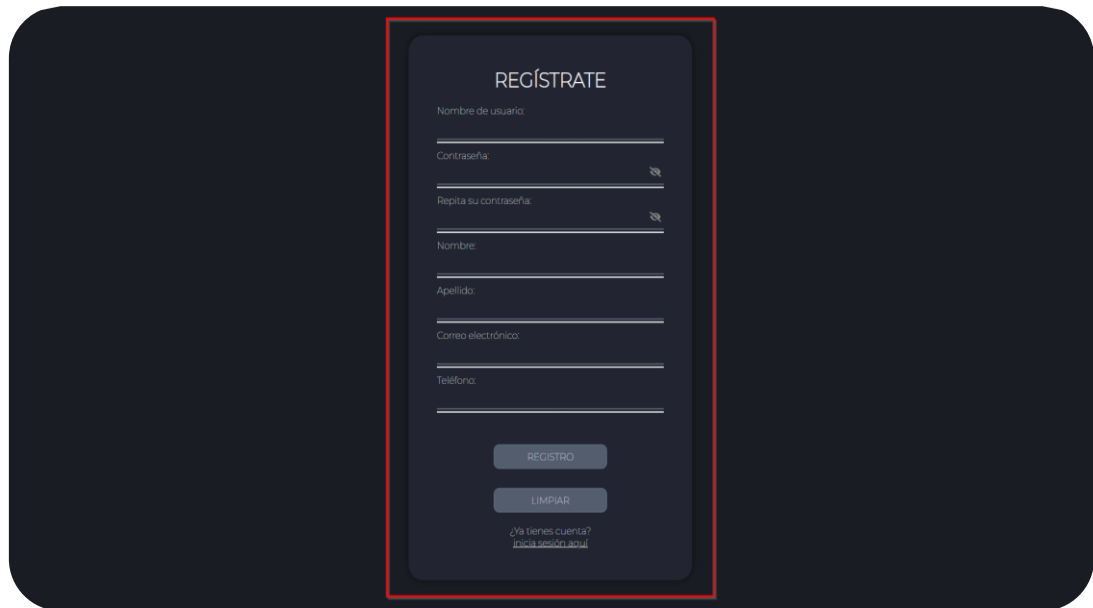
El uso de **password\_verify()** es fundamental para garantizar la seguridad de las contraseñas de los usuarios. Esta función realiza una comparación segura de la contraseña ingresada con el hash almacenado, sin revelar la información original.

## Vista de registro

En la vista de registro, los usuarios tienen la oportunidad de crear una cuenta en nuestro sistema. La interfaz se presenta en dos cajas distintas: una para el formulario de registro y otra para mostrar mensajes de error en caso de que se produzcan.

En el formulario de registro, los usuarios deben completar los campos requeridos, como el nombre de usuario, contraseña, nombre, apellido, correo electrónico y teléfono. Cada campo tiene sus propias restricciones y validaciones para garantizar la integridad de los datos ingresados.

## Caja de Registro



REGÍSTRATE

Nombre de usuario

Contraseña

Repita su contraseña

Nombre

Apellido

Correo electrónico

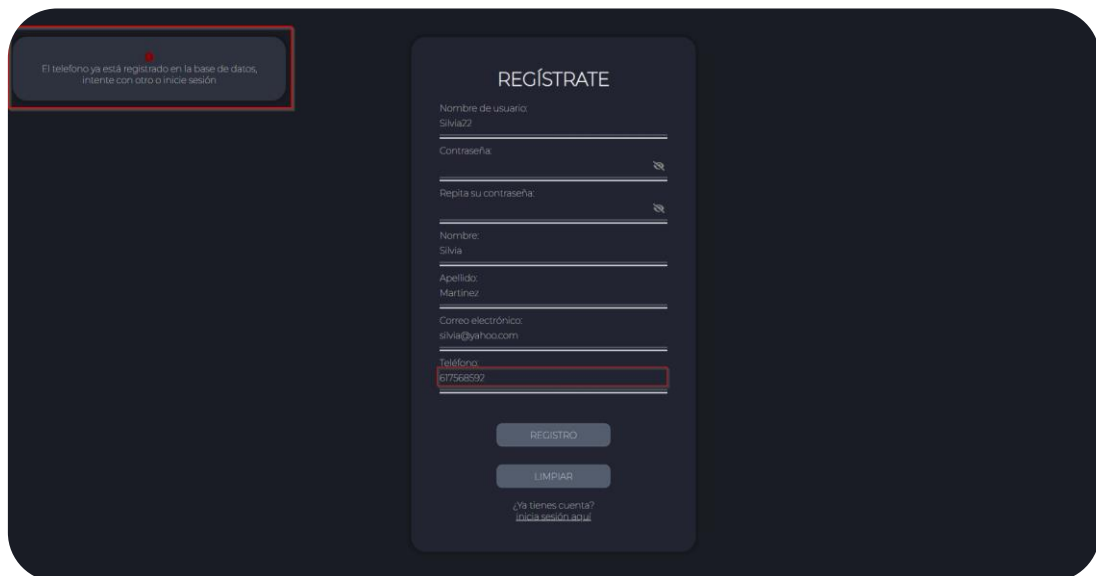
Teléfono

REGISTRO

LIMPIAR

¿Ya tienes cuenta?  
[inicia sesión aquí](#)

## Caja Mensaje – Error de Registro



El teléfono ya está registrado en la base de datos, intente con otro o inicie sesión

REGÍSTRATE

Nombre de usuario  
Silvia22

Contraseña

Repita su contraseña

Nombre  
Silvia

Apellido  
Martínez

Correo electrónico  
silvia@yahoo.com

Teléfono  
67528592

REGISTRO

LIMPIAR

¿Ya tienes cuenta?  
[inicia sesión aquí](#)

## Interacción del usuario con la vista

En este paso se explica el proceso utilizado para que los usuarios puedan crear una cuenta en la aplicación. A continuación, se muestra una descripción detallada de las verificaciones y restricciones realizadas en el controlador (**registrar()**) para cada campo del formulario:

1. Nombre de usuario:
  - Se verifica si ya existe otro usuario con el mismo nombre de usuario en la base de datos. En caso afirmativo, se muestra un mensaje de error indicando que el nombre de usuario ya está registrado.
2. Contraseña:
  - Se verifica que la contraseña cumpla con el siguiente formato:
    - Al menos una letra mayúscula.
    - Al menos un número.
    - Una longitud mínima de 8 caracteres.
  - Si la contraseña no cumple con alguno de estos requisitos, se muestran mensajes de error específicos.
3. Repetir contraseña:
  - Se verifica que la contraseña ingresada en este campo sea igual a la contraseña ingresada en el campo de contraseña anterior. Si no coinciden, se muestra un mensaje de error indicando que las contraseñas deben coincidir.
4. Nombre:
  - No se realizan verificaciones adicionales para este campo.
5. Apellido:
  - No se realizan verificaciones adicionales para este campo.
6. Correo electrónico:
  - Se verifica que se proporcione una dirección de correo electrónico válida utilizando una expresión regular.
  - Si el correo electrónico no tiene un formato válido, se muestra un mensaje de error.
7. Teléfono:
  - Se verifica que el número de teléfono cumpla con los siguientes requisitos:
    - Debe ser una cadena numérica.

- Debe tener una longitud de 9 dígitos.
- Si el número de teléfono no cumple con alguno de estos requisitos, se muestra un mensaje de error.

Además de las verificaciones específicas de cada campo, también se realizan las siguientes comprobaciones generales:

- Se verifica si hay campos obligatorios que no se hayan completado. Si alguno de los campos obligatorios está vacío, se muestra un mensaje de error.
- Si no se produce ningún error en las verificaciones anteriores, se comprueba si la contraseña y la contraseña repetida coinciden. Si no coinciden, se muestra un mensaje de error.

En caso de que no se produzcan errores, se crea un objeto **Usuario**, se encripta la contraseña y se inserta el nuevo usuario en la base de datos.



La importancia del **password hash** en un sistema de autenticación reside en la seguridad de las contraseñas de los usuarios. El uso de password hash es una práctica fundamental para proteger la información confidencial y evitar posibles brechas de seguridad.

## Vista de mensajes

La vista de mensajes tiene un diseño elegante y depurado, con bordes en cajas y componentes. Se ha buscado una estructura en el diseño similar a la aplicación de mensajería WhatsApp. Se utiliza JavaScript para añadir animaciones y efectos interactivos, como respuestas visuales al hacer hover en los componentes. El diseño se centra en la legibilidad y la experiencia de usuario.

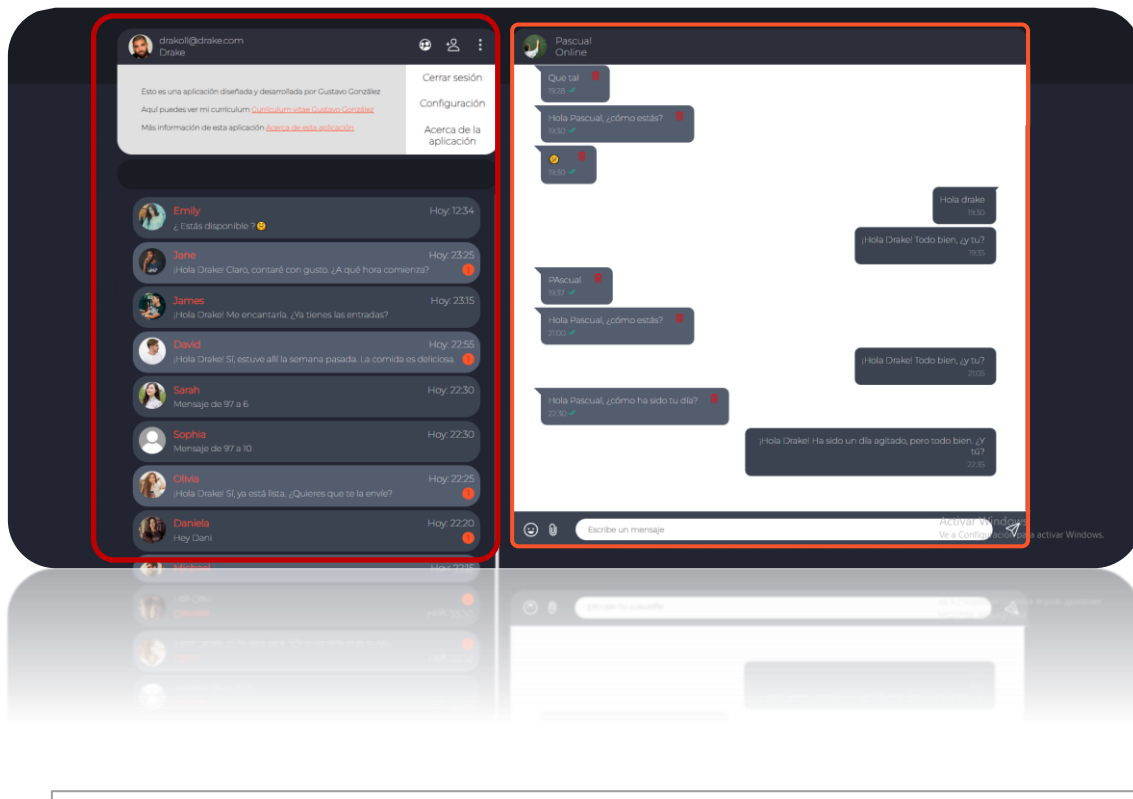
La vista de mensajes presenta los siguientes componentes principales:

1. **Estructura general:** La vista se divide en dos secciones principales, que son representadas por dos contenedores. Esto permite mostrar la lista de mensajes y el área de chat de manera organizada.



2. **Cabeceras:** Se incluyen cabeceras para identificar los usuarios o grupos de chat. Estas cabeceras pueden contener la foto de perfil, el nombre de usuario, el estado actual del usuario o menús de navegación.
  3. **Lista de mensajes:** Se muestra una lista de mensajes que permite visualizar la conversación. Cada mensaje incluye el contenido del mensaje y la hora de envío.
  4. **Contenedor de usuario dinámico:** Al hacer clic en el icono de menú (Elipsis de tres puntos), se muestra un contenedor oculto con tres enlaces: "**Acerca de**", "**Configuración**" y "**Cerrar sesión**". Al hacer clic en cada enlace, se muestra contenido diferente en el contenedor dinámico.
    - "**Acerca de**": Muestra información personal y detalles sobre la aplicación, como el currículum y otros datos relevantes.
    - "**Configuración**": Permite al usuario cambiar su nombre de usuario y estado actual.
    - "**Cerrar sesión**": Ofrece la opción de cerrar la sesión actual y salir de la aplicación.
  5. **Caja de chat:** Es el área donde se ingresan los mensajes y se visualizan los mensajes enviados y recibidos. Puede incluir opciones para enviar mensajes, adjuntar archivos u otros elementos interactivos.
-

## Estructura General



## Interacción del usuario con la vista

En esta sección, exploraremos cómo los usuarios interactúan con nuestra aplicación, un clon de WhatsApp. Nuestra aplicación permite a los usuarios enviar mensajes, añadir amigos, abrir conversaciones, enviar imágenes, seleccionar la caja de un usuario en concreto y editar su cuenta de usuario. A continuación, describiremos el flujo de interacción general de los usuarios con la aplicación:

1. **Cambio de foto de perfil:** El usuario puede cambiar su foto de perfil haciendo click sobre la imagen ubicada en la cabecera de usuario.
2. **Lista de amigos:** Una vez en la pantalla principal, el usuario puede ver una lista de amigos disponibles. Cada amigo se muestra en una caja separada, que contiene su imagen de perfil, el resumen del último mensaje de la conversación, la fecha de envío del mensaje y el número mensajes no leídos de la conversación. El usuario puede hacer clic en una de las cajas para abrir una conversación con ese amigo.
3. **Envío de mensajes de texto:** Dentro de una conversación, el usuario puede escribir mensajes en un campo de texto y enviarlos al amigo seleccionado. Los mensajes se muestran en forma de burbujas de chat, donde el nombre del remitente y la hora del mensaje también

están presentes. Los mensajes se pueden enviar presionando el botón de enviar o utilizando la tecla Enter.

4. **Agregar amigos:** El usuario tiene la opción de agregar nuevos amigos a su lista. Esto se puede hacer utilizando una función de búsqueda, donde el usuario ingresa el nombre de usuario de la persona que desea agregar. Si se encuentra una coincidencia, se muestra una caja con la información del amigo y se puede enviar una solicitud de amistad

Esta funcionalidad se hace de manera asíncrona con AJAX, cada letra que busca el usuario manda una petición a la base de datos, para mostrar usuarios registrados en la aplicación que tengan nombres de usuario coincidentes.

5. **Envío de imágenes:** Además de los mensajes de texto, los usuarios también pueden compartir imágenes con sus amigos. Hay un botón específico que permite al usuario seleccionar una imagen de su dispositivo y enviarla en la conversación activa.
6. **Gestión de conversaciones:** Los usuarios pueden administrar sus conversaciones abiertas. Pueden volver a la lista de amigos para seleccionar otra conversación o cerrar la aplicación. Además, la aplicación puede proporcionar funciones adicionales, como archivar o eliminar conversaciones antiguas.
7. **Edición de cuenta de usuario:** La aplicación permite a los usuarios editar su cuenta de usuario. Esto incluye la capacidad de cambiar la foto de perfil, actualizar la información personal como el nombre y el apellido, y modificar la contraseña.

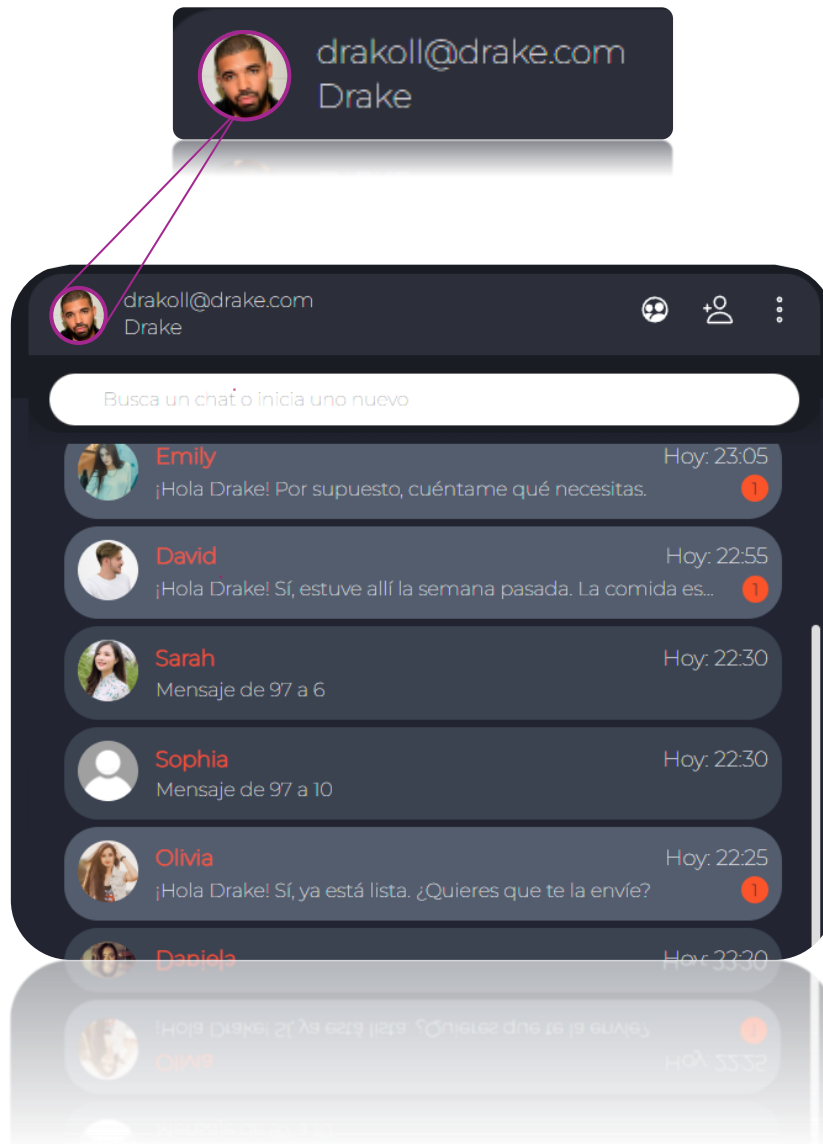
En resumen, nuestra aplicación de clon de WhatsApp brinda a los usuarios una experiencia similar a la aplicación original. Los usuarios pueden interactuar fácilmente con sus amigos, enviar mensajes y compartir imágenes. Además, tienen la opción de editar su cuenta de usuario para personalizar su perfil y mantener su información actualizada. La interfaz de usuario intuitiva y las funciones disponibles permiten una comunicación fluida y agradable entre los usuarios



La importancia del **password hash** en un sistema de autenticación reside en la seguridad de las contraseñas de los usuarios. El uso de password hash es una práctica fundamental para proteger la información confidencial y evitar posibles brechas de seguridad.

## Componente – foto perfil

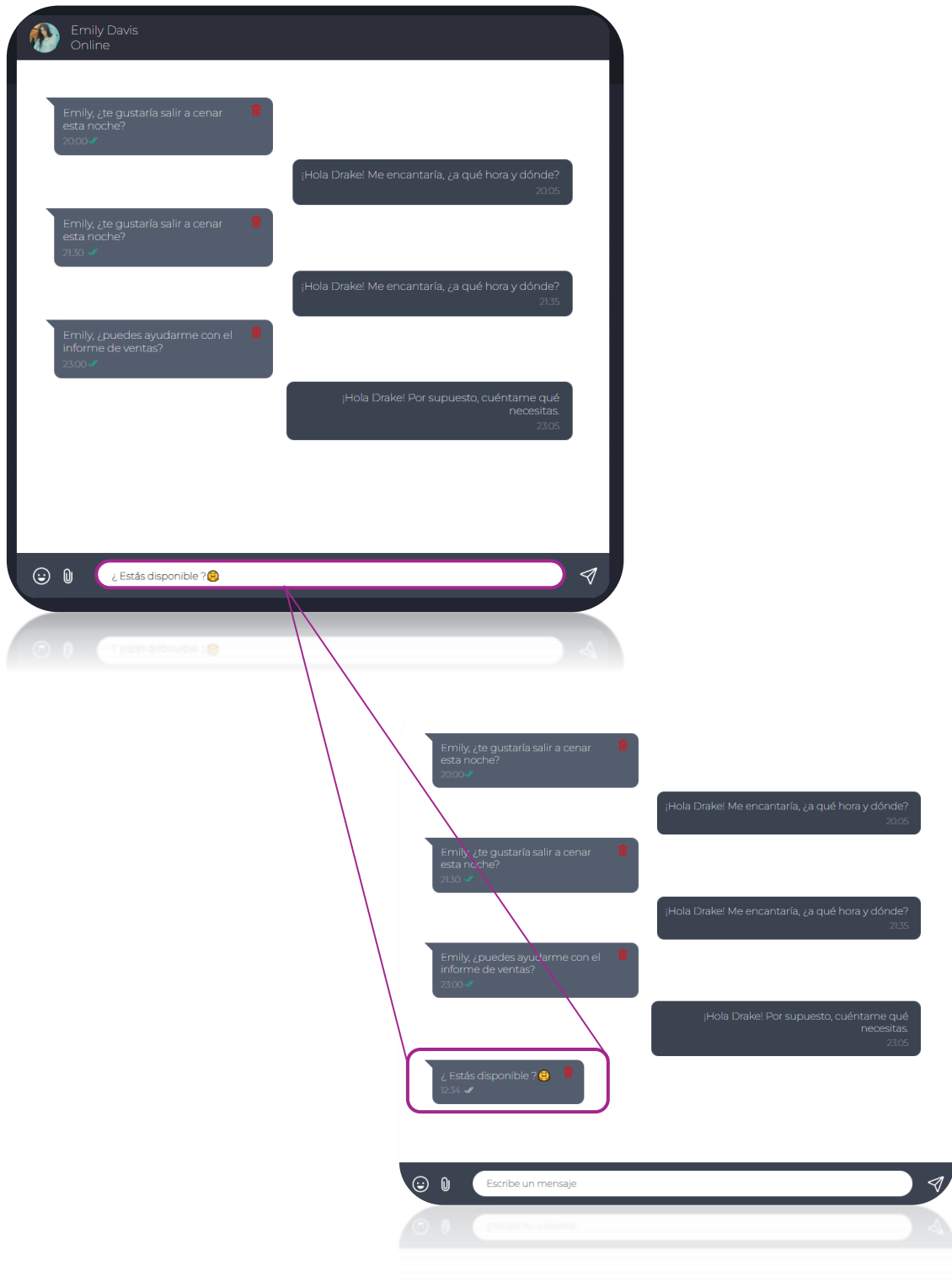
---



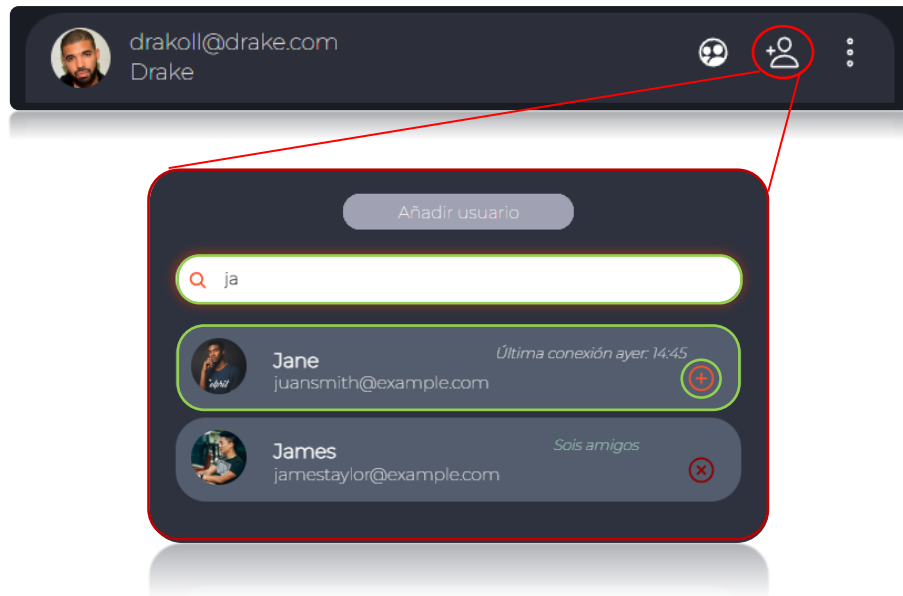
## Componente – Lista de Mensajes & selección de chat



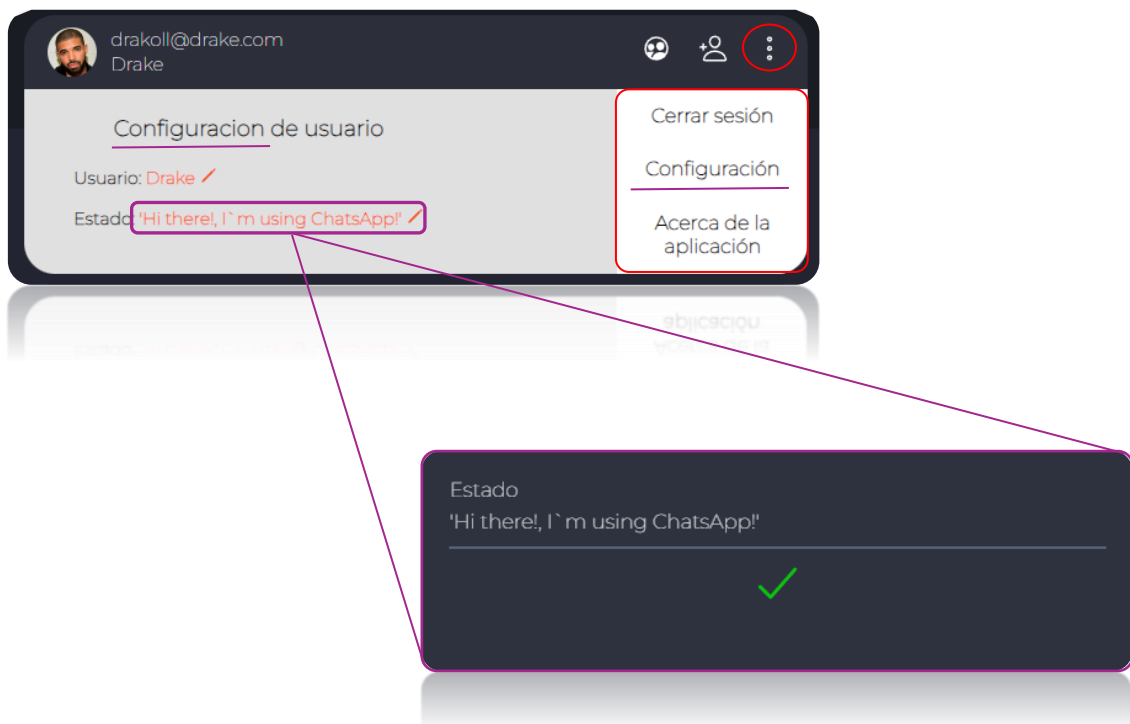
## Componente – Envío de Mensajes & cajas de mensajes



## Componente – Agregar amigos o “contactos”

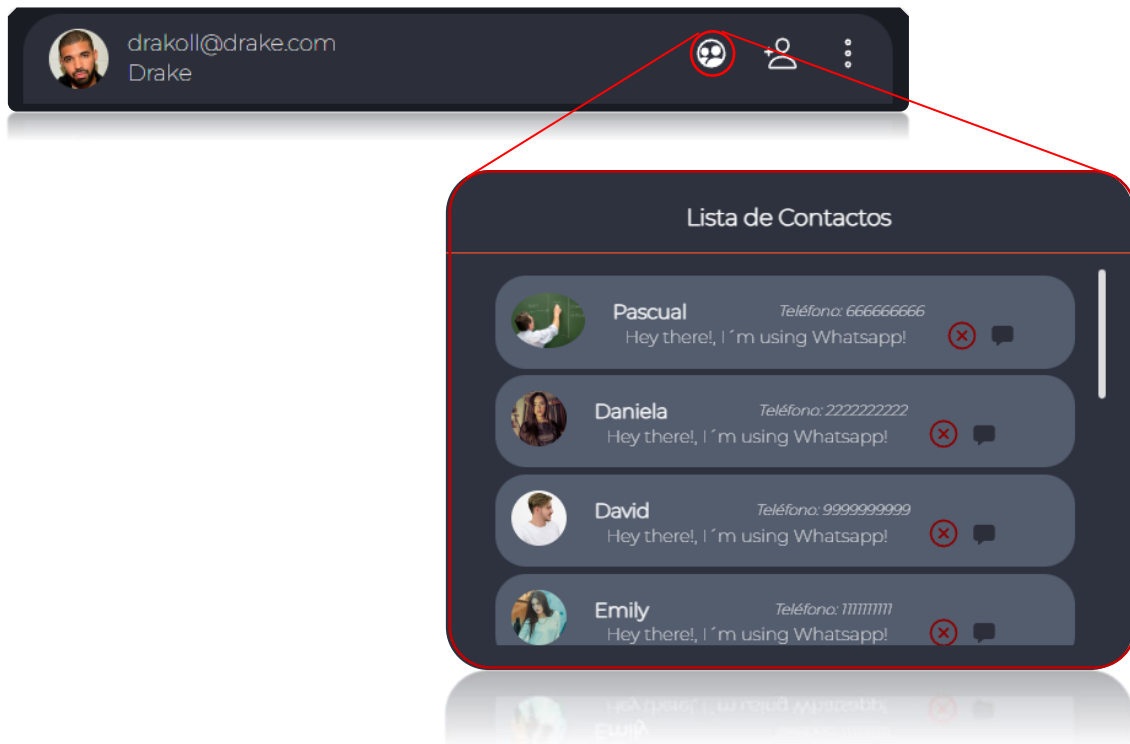


## Componente – Contenido dinámico “Configuración”



## Componente – “Mis contactos”

---





## VI. Mantenimiento y evolución

Como mejora de futuro, se propone traspasar las funcionalidades de redirección de la aplicación por tecnología asíncrona para brindar una mejor experiencia al usuario. Esto implica utilizar técnicas como AJAX (Asynchronous JavaScript and XML) o frameworks modernos de JavaScript para realizar las solicitudes y actualizaciones en segundo plano, sin necesidad de recargar completamente la página.

El objetivo de esta mejora es agilizar la interacción del usuario con la aplicación, evitando esperas innecesarias y proporcionando una experiencia más fluida. Al implementar la tecnología asíncrona, se podrían realizar las siguientes tareas:

1. Carga de contenido dinámico: En lugar de cargar páginas completas al navegar por diferentes secciones de la aplicación, se podría cargar únicamente el contenido necesario utilizando solicitudes asíncronas. Esto permitiría una transición suave y rápida entre diferentes vistas.
2. Actualización en tiempo real: Mediante el uso de eventos y actualizaciones asincrónicas, se podría proporcionar información actualizada en tiempo real a los usuarios. Por ejemplo, notificaciones de nuevos mensajes recibidos o cambios en el estado de conexión de amigos.
3. Envío y recepción de mensajes instantáneos: Al utilizar tecnología asíncrona, se podrían enviar y recibir mensajes en tiempo real sin la necesidad de recargar la página. Esto permitiría una comunicación más fluida y rápida entre los usuarios.
4. Interacciones sin interrupciones: Al realizar acciones como agregar o eliminar amigos, cambiar la configuración de la cuenta u otras interacciones, se podría utilizar tecnología asíncrona para realizar las operaciones sin interrumpir la experiencia de navegación del usuario.

Es importante tener en cuenta que la implementación de tecnología asíncrona requerirá cambios en el backend y en el frontend de la aplicación. Se deberá refactorizar el código existente y utilizar las herramientas y técnicas adecuadas para lograr una transición suave y eficiente hacia la tecnología asíncrona.

Esta mejora puede llevar tiempo y esfuerzo, por lo que se sugiere incluirla en una etapa de mantenimiento y evolución posterior al desarrollo inicial de la aplicación

# VII. Planificación

## Análisis y Diseño (10 horas)

- Revisión de requisitos y especificaciones.
- Diseño de la arquitectura y estructura de la aplicación.
- Diseño de la interfaz de usuario y experiencia de usuario.

Configuración del entorno (5 horas)

- Instalación y configuración del servidor web.
- Configuración de la base de datos.
- Configuración de las herramientas de desarrollo (por ejemplo, IDE, control de versiones).

## Backend (30 horas)

- Desarrollo del sistema de autenticación y gestión de usuarios.
- Implementación de la lógica del chat y gestión de conversaciones.
- Desarrollo de la funcionalidad de envío y recepción de mensajes.
- Implementación de la funcionalidad de cambio de foto de perfil.
- Implementación de la funcionalidad de agregar y gestionar amigos.
- Desarrollo de la funcionalidad de envío de imágenes.
- Implementación de la gestión de conversaciones abiertas.
- Desarrollo de la funcionalidad de edición de cuenta de usuario.

## Frontend (20 horas)

- Diseño y desarrollo de la interfaz de usuario de la vista principal.
- Implementación de la lista de amigos y visualización de perfiles.
- Desarrollo de la funcionalidad de envío y visualización de mensajes.
- Implementación de la funcionalidad de cambio de foto de perfil en la interfaz de usuario.
- Diseño y desarrollo de la interfaz de búsqueda y agregado de amigos.

- Implementación de la funcionalidad de envío de imágenes en la interfaz de usuario.
- Desarrollo de la interfaz de gestión de conversaciones.
- Implementación de la interfaz de edición de cuenta de usuario.

Documentación y Entrega (10 horas):

- Elaboración de la documentación del proyecto, incluyendo manuales de usuario y técnicos.
- Preparación de los archivos finales y entrega del proyecto.

## VIII. Bibliografía

### Comunidades y foros en línea

- Stack Overflow  
<https://stackoverflow.com/>
- Foro del sitio oficial de PHP  
<https://www.php.net/manual/es/maillinglists.php>
- Foro del sitio oficial de JavaScript  
<https://developer.mozilla.org/es/docs/Web/Foros>

### Sitios de desarrollo web

- MDN Web Docs  
<https://developer.mozilla.org/es/>
- W3Schools  
<https://www.w3schools.com/>
- PHP.net  
<https://www.php.net/docs.php>

### Documentación oficial

- Documentación oficial de JavaScript  
<https://developer.mozilla.org/es/docs/Web/JavaScript>
- Documentación oficial de PHP

<https://www.php.net/docs.php>

- Documentación oficial de AJAX:  
<https://developer.mozilla.org/es/docs/Web/Guide/AJAX>
- Documentación oficial de CSS  
<https://developer.mozilla.org/es/docs/Web/CSS>