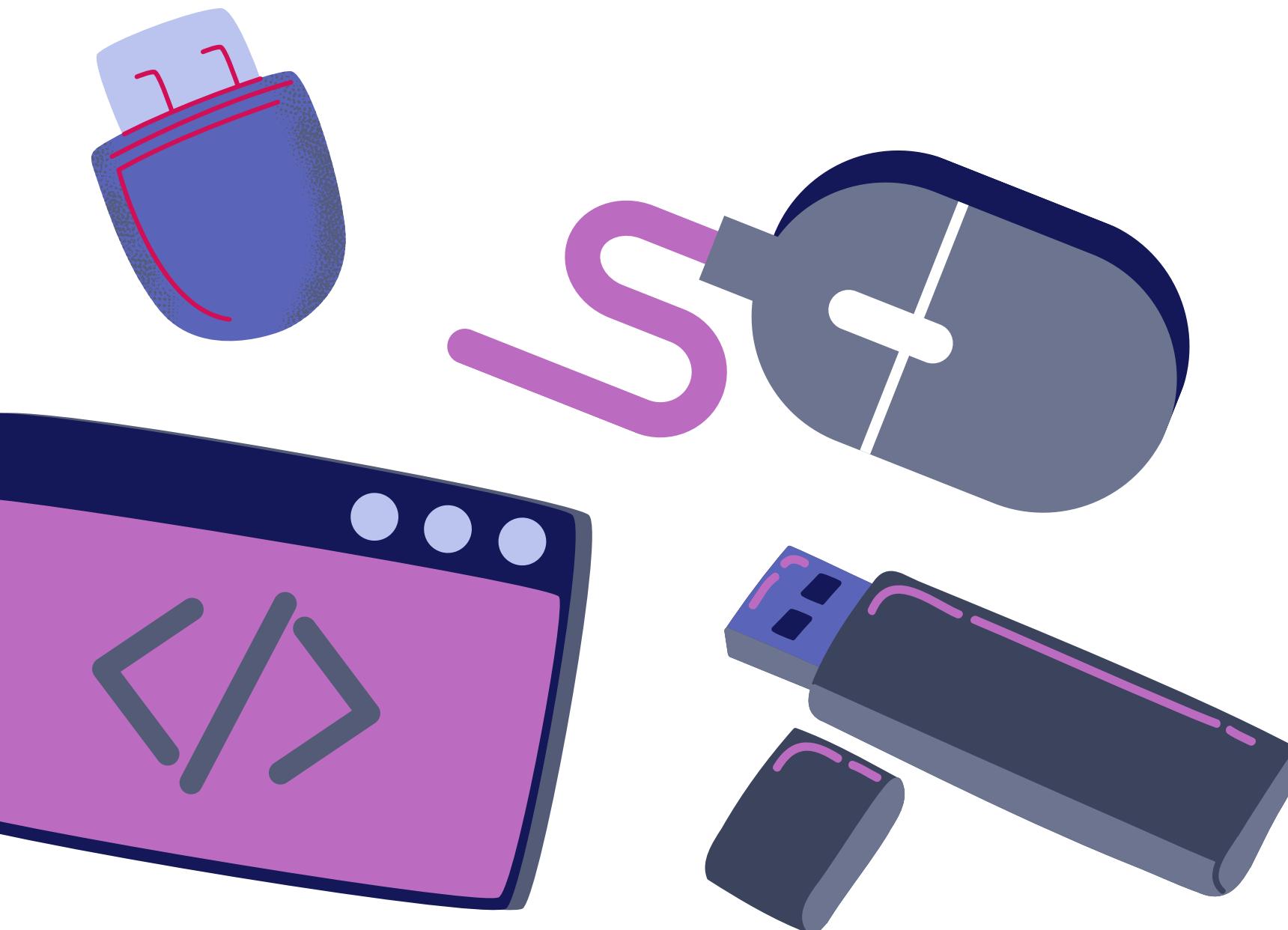


Proyecto Realizado por Carlos Edgardo Lopez Barrera

# PROYECTO 01

## *xmpp-connect*

# ÍNDICE



---

**01. Introducción**

---

**02. Tecnologías**

---

**03. Características**

---

**04. Dificultades**

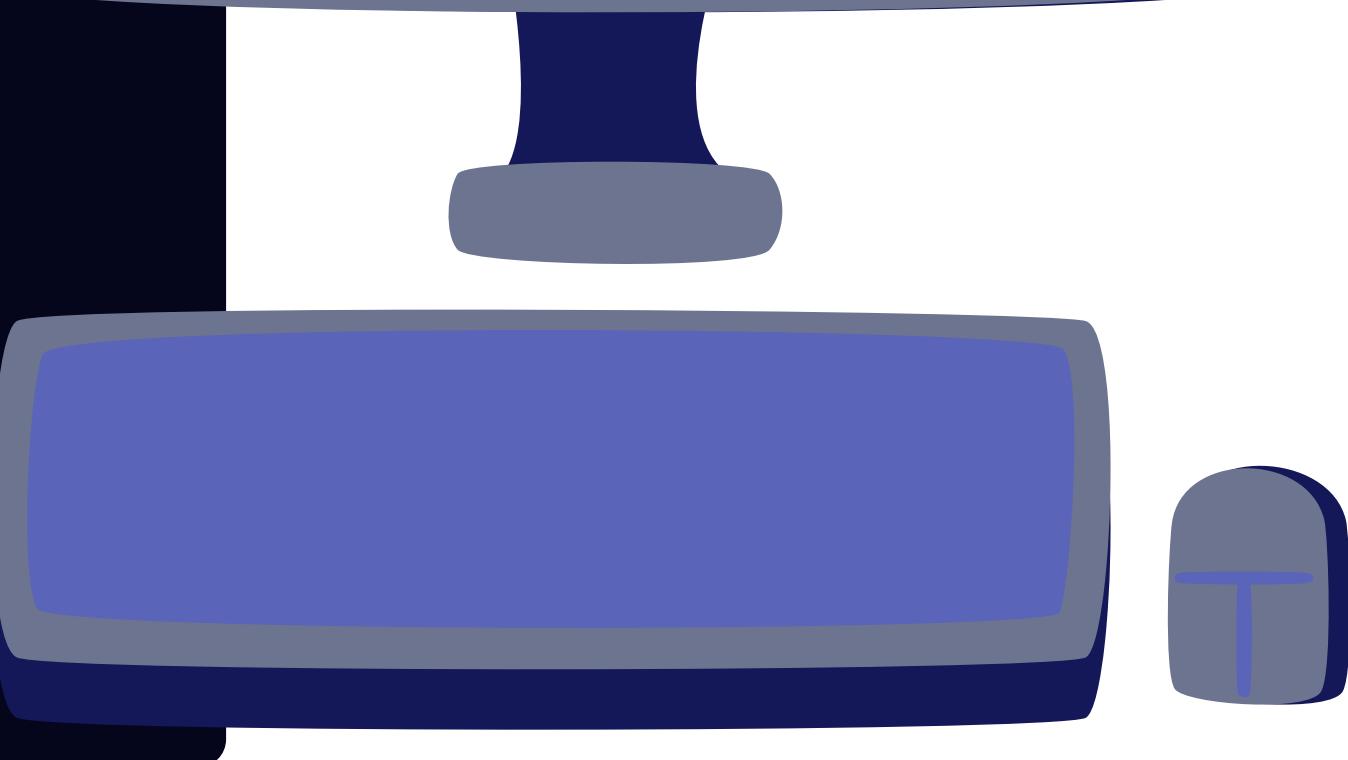
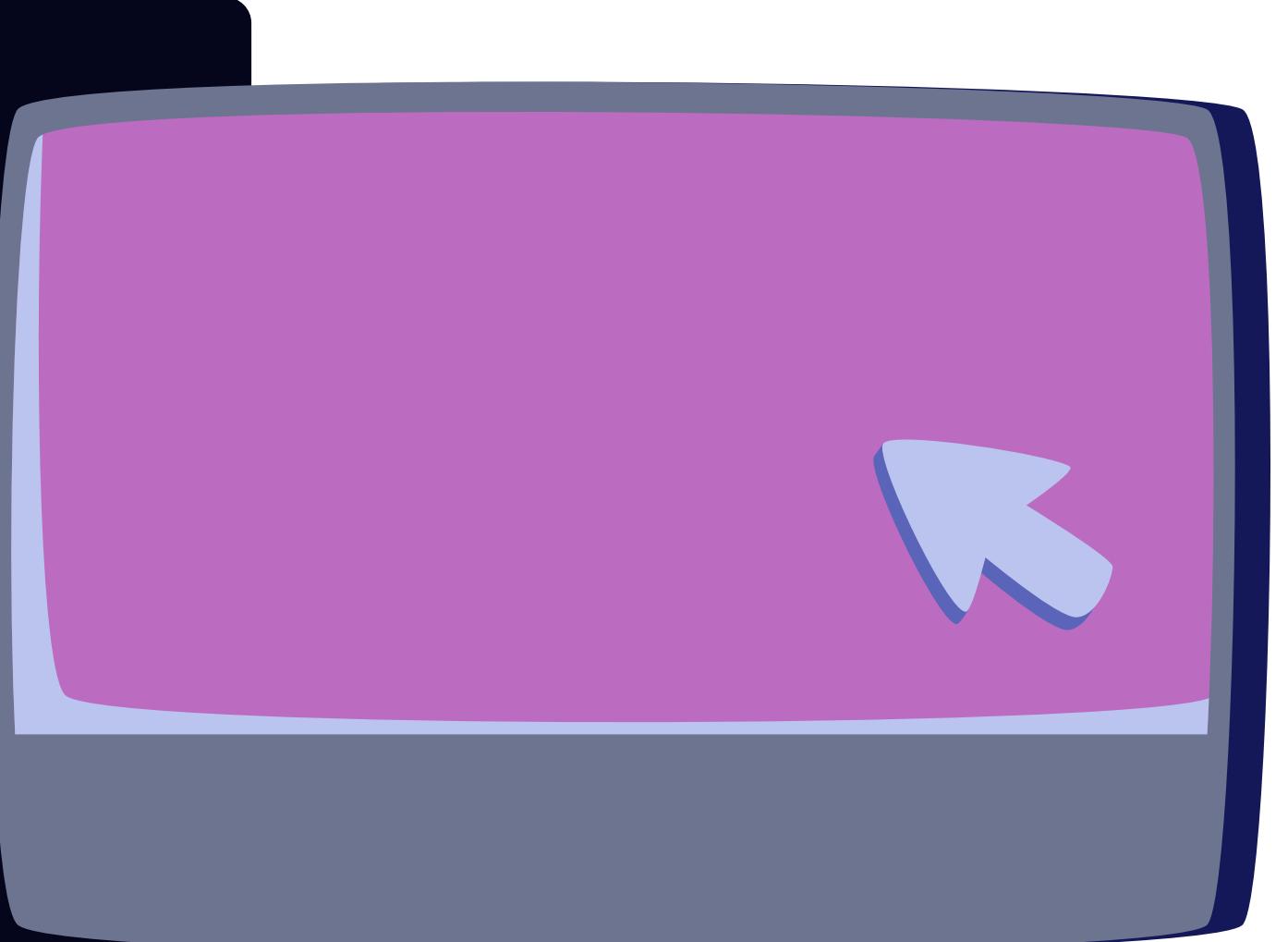
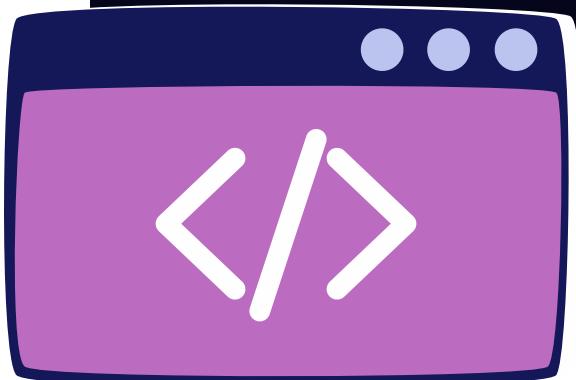
---

**05. Lecciones aprendidas**

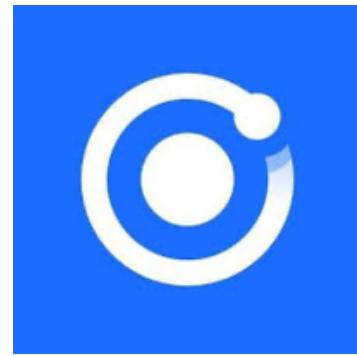
---

# Introducción

Este proyecto consiste en el desarrollo de una aplicación de chat multiplataforma utilizando Ionic Angular y Strophe.js para implementar el protocolo XMPP. La aplicación permite a los usuarios conectarse a un servidor XMPP, gestionar contactos, participar en chats individuales, y manejar su estado de presencia en tiempo real.



# Tecnologías utilizadas



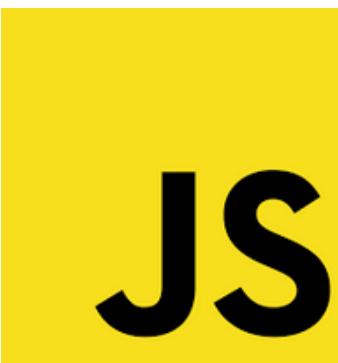
## Ionic Framework

framework de desarrollo de aplicaciones móviles híbridas



## Angular

framework estructural para desarrollar aplicaciones web dinámicas.



## Strophe.js

biblioteca JavaScript diseñada para facilitar la conexión y comunicación con servidores XMPP (Extensible Messaging and Presence Protocol).



## XMPP

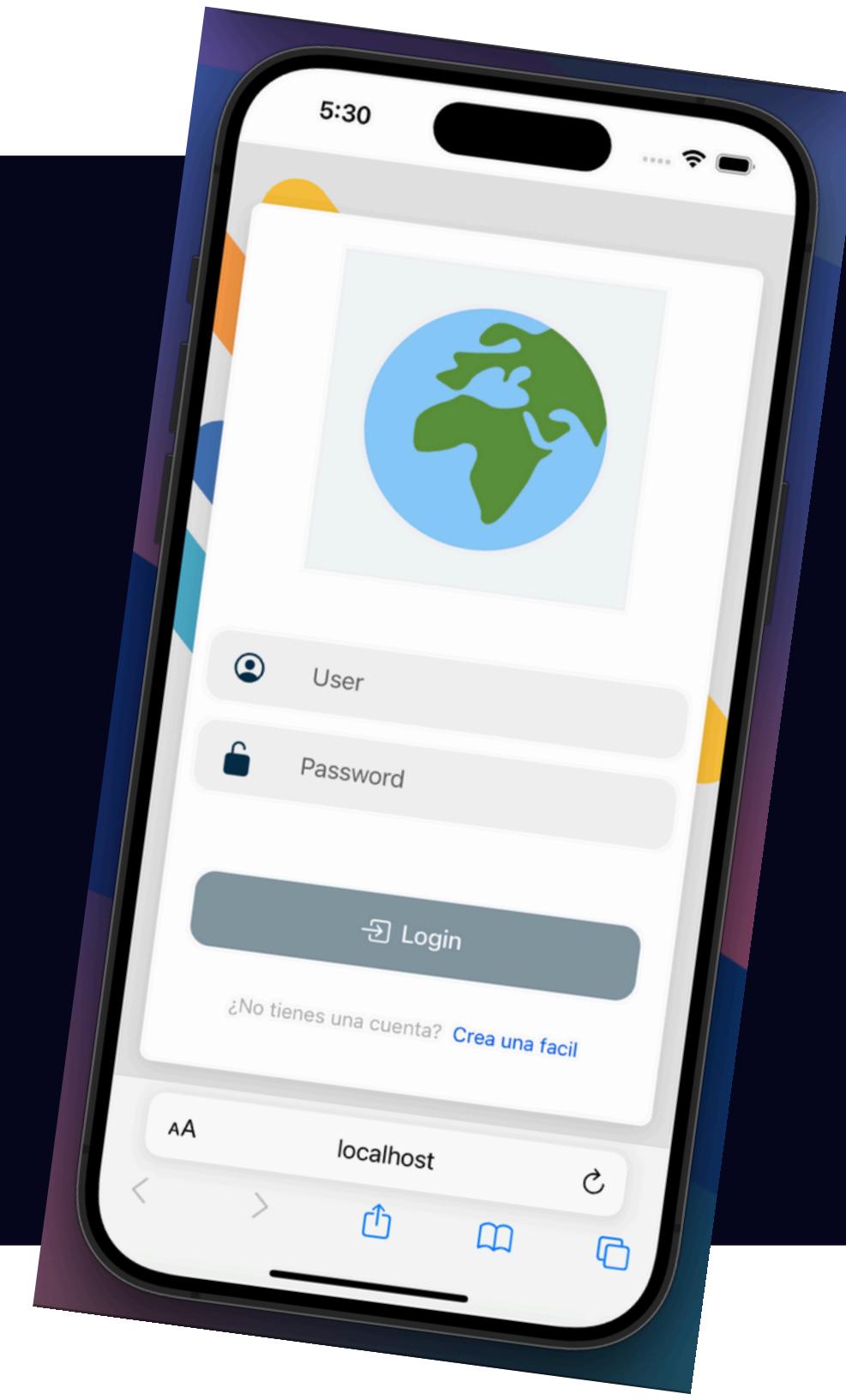
protocolo abierto basado en XML que se utiliza para mensajería instantánea y comunicación en tiempo real.

# CARACTERISTICAS

## AUTENTICACION

### Login

- Los usuarios pueden iniciar sesión en la aplicación utilizando su JID (Jabber ID) y contraseña. Esto permite autenticar su identidad y acceder a sus contactos y mensajes.

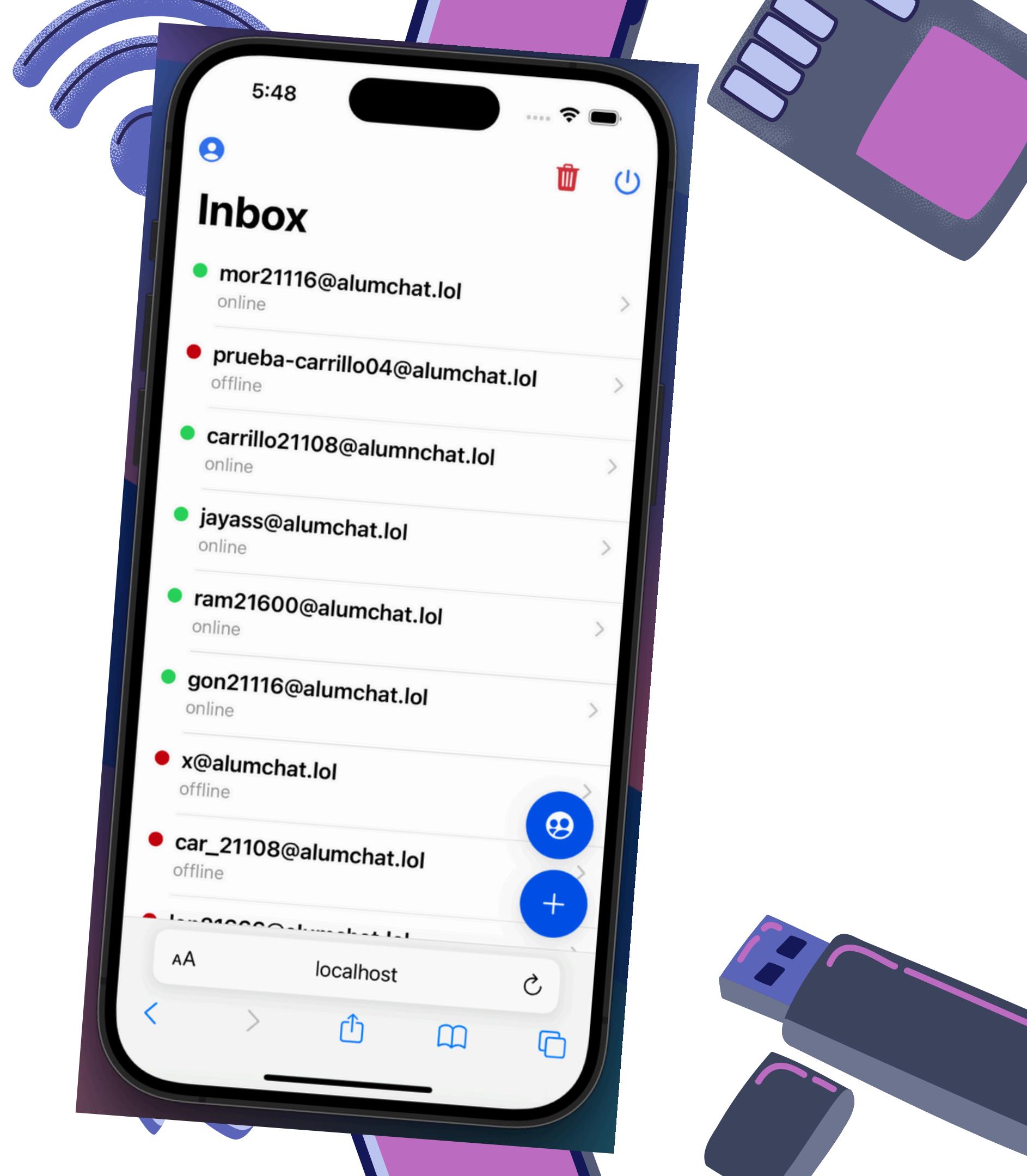


### Registro

- Los usuarios pueden crear una nueva cuenta que se registra en el servidor, y posteriormente iniciar sesión,

# PRESENCIA

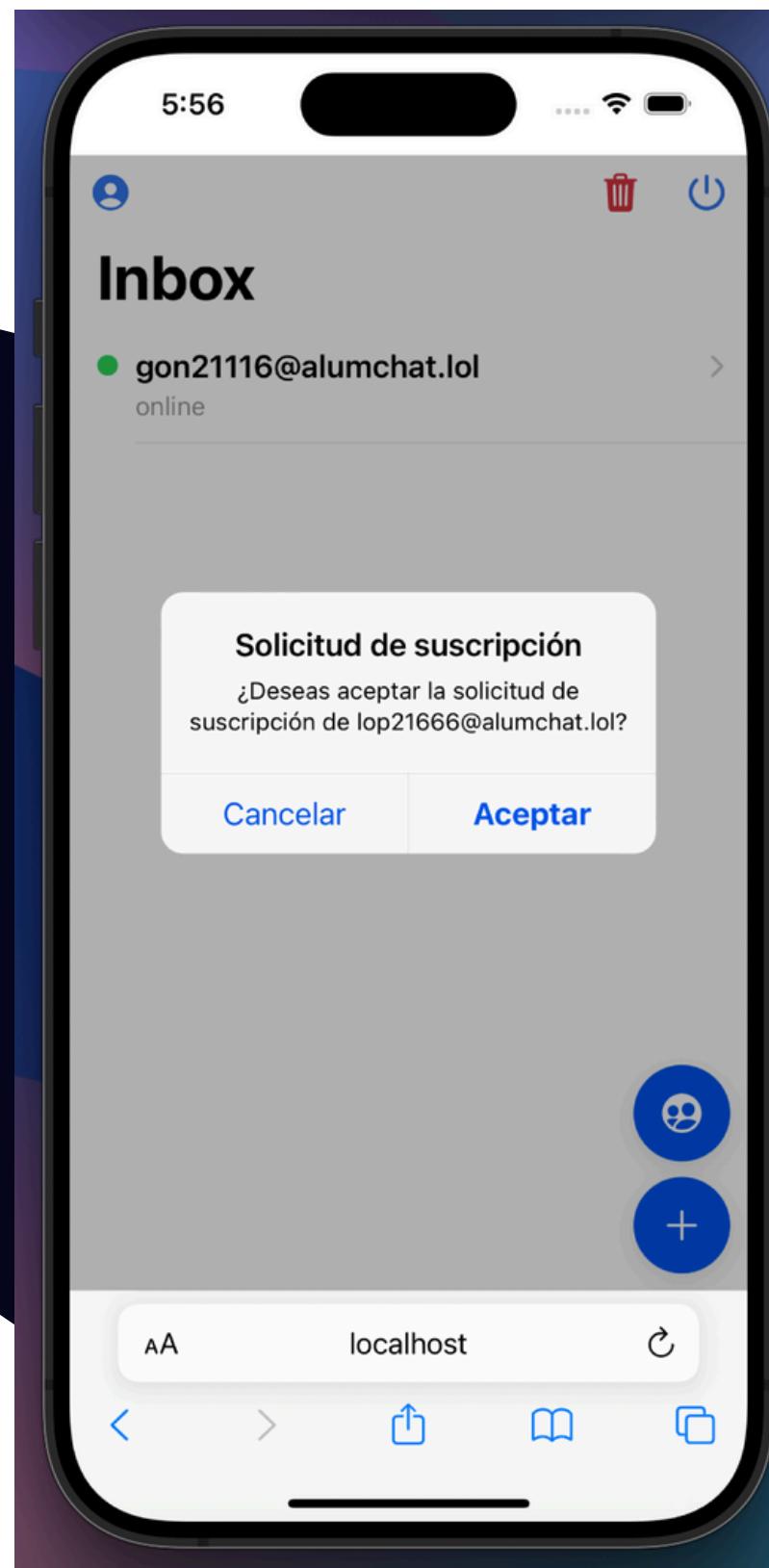
- Actualizar estado de presencia: Los usuarios pueden seleccionar y actualizar su estado (online, away, offline) en tiempo real.
- Visualización del estado de contactos: Implementación de actualización en tiempo real del estado de presencia de los contactos en la lista de contactos.



# MENSAJERÍA

- Desarrollo de la funcionalidad para el envío y recepción de mensajes entre dos usuarios.
- Almacenamiento y visualización del historial de mensajes por contacto o grupo, accesible desde la interfaz de usuario.
- Envio de archivos
- Notificación de mensajes

# Gestión de contactos



## AGREGAR Y GESTIONAR CONTACTOS

Funcionalidad para agregar nuevos contactos a la lista utilizando el JID.

## MANEJO DE SOLICITUDES DE SUSCRIPCIÓN

Implementación de notificaciones para aceptar o rechazar solicitudes de suscripción de otros usuarios.

## VER DETALLES DE CONTACTO

Visualización de información detallada sobre cada contacto, incluyendo su estado de presencia.



# Dificultades

## 01. COMPLEJIDAD EN LA DOCUMENTACIÓN

La integración de Strophe.js presentó desafíos debido a la documentación limitada y a veces desactualizada. Esto requirió una exploración más profunda del código fuente y de ejemplos de la comunidad para comprender completamente las capacidades y limitaciones de la biblioteca.

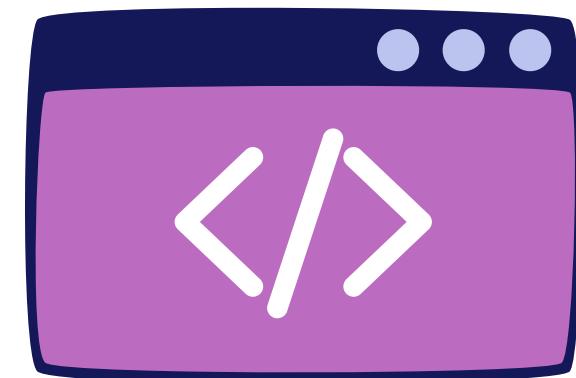
## 02. MANEJO DE WEBSOCKETS

Configurar correctamente la conexión a través de websockets en Strophe.js fue un reto, especialmente en lo referente a la compatibilidad con diferentes servidores XMPP. Fue necesario realizar varias pruebas de conexión y ajustes de configuración para asegurar la estabilidad de la conexión.

## 02. MANEJO DE MENSAJES Y PRESENCIA

Garantizar la entrega y actualización de mensajes y estados de presencia en tiempo real, sin causar sobrecarga en el cliente o el servidor, fue un desafío significativo. La necesidad de optimizar el flujo de datos y minimizar las interrupciones requirió un ajuste fino de las configuraciones.

# Lecciones aprendidas



01

## DOMINIO DE XMPP Y STROPHE.JS

La implementación de esta aplicación permitió un aprendizaje profundo del protocolo XMPP, su estructura basada en XML, y cómo manejar diferentes aspectos de la comunicación en tiempo real.

02

## GESTIÓN DE DEPENDENCIAS Y CONFIGURACIONES

La experiencia de ajustar configuraciones de conexión y manejar dependencias en un entorno de desarrollo móvil híbrido fue enriquecedora, permitiendo una mejor comprensión de cómo estos ajustes impactan la experiencia del usuario.

02

## OPTIMIZACIÓN DEL MANEJO DE ESTADOS Y EVENTOS

El manejo eficiente de eventos y estados en una aplicación en tiempo real se convirtió en una habilidad clave, especialmente en el contexto de mantener la estabilidad de la conexión y la sincronización de datos.

# MUCHAS GRACIAS