



Universidad del Valle de Guatemala

PROYECTO 1

USO DE UN PROTOCOLO EXISTENTE

Adrian Fulladolsa Palma

Índice de Contenidos

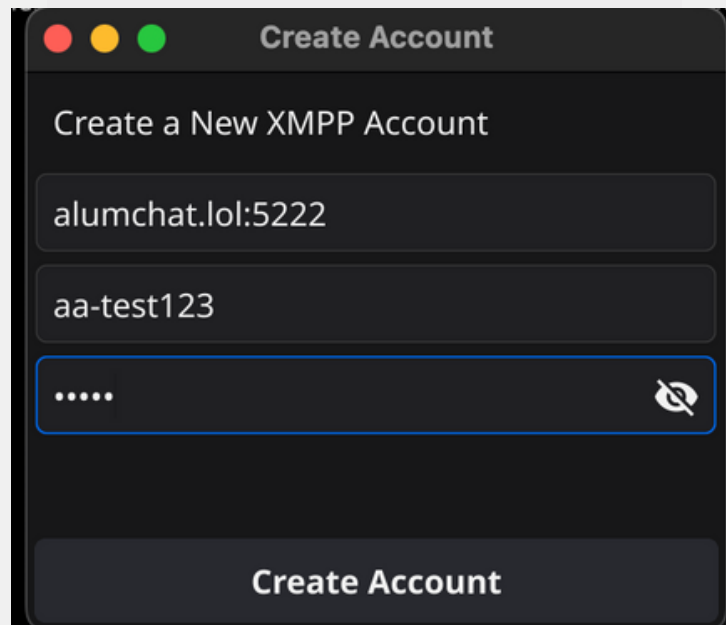
01 Características Implementadas

02 Dificultades

03 Lecciones Aprendidas

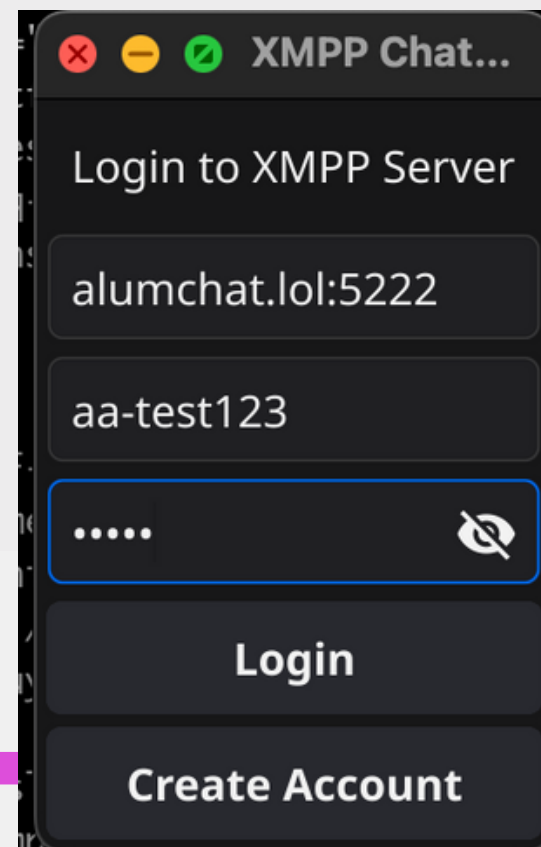
CARACTERÍSTICAS IMPLEMENTADAS

Registro de nuevo usuario



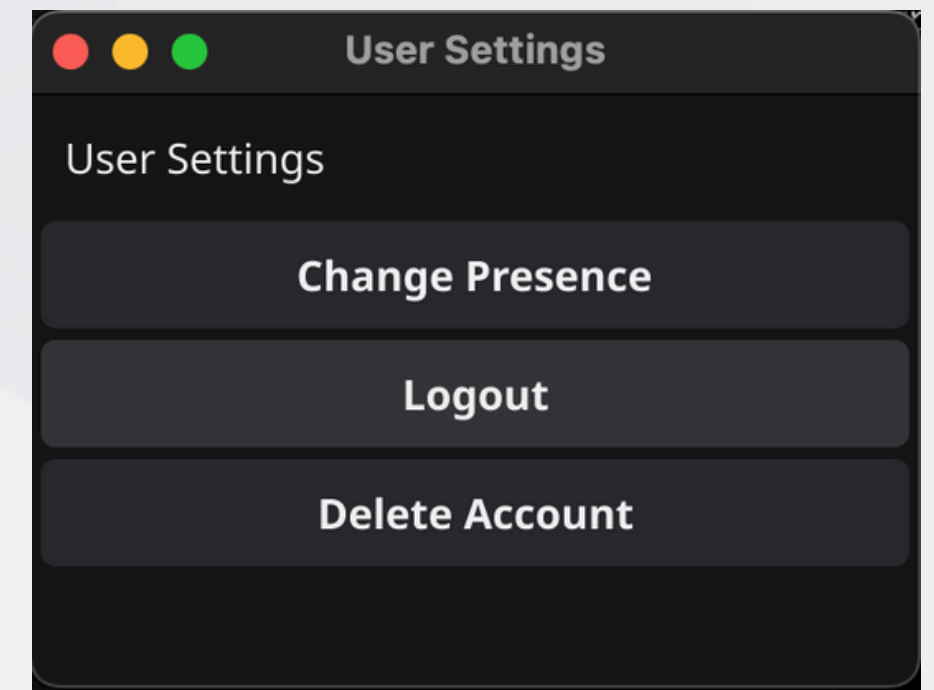
A dark-themed dialog box titled "Create Account" with standard macOS window controls. It contains the text "Create a New XMPP Account" and three input fields: the first contains "alumchat.lol:5222", the second contains "aa-test123", and the third is a password field with five dots and a toggle icon. A "Create Account" button is at the bottom.

Inicio de sesión



A dark-themed dialog box titled "XMPP Chat..." with standard macOS window controls. It contains the text "Login to XMPP Server" and three input fields: the first contains "alumchat.lol:5222", the second contains "aa-test123", and the third is a password field with five dots and a toggle icon. "Login" and "Create Account" buttons are at the bottom.

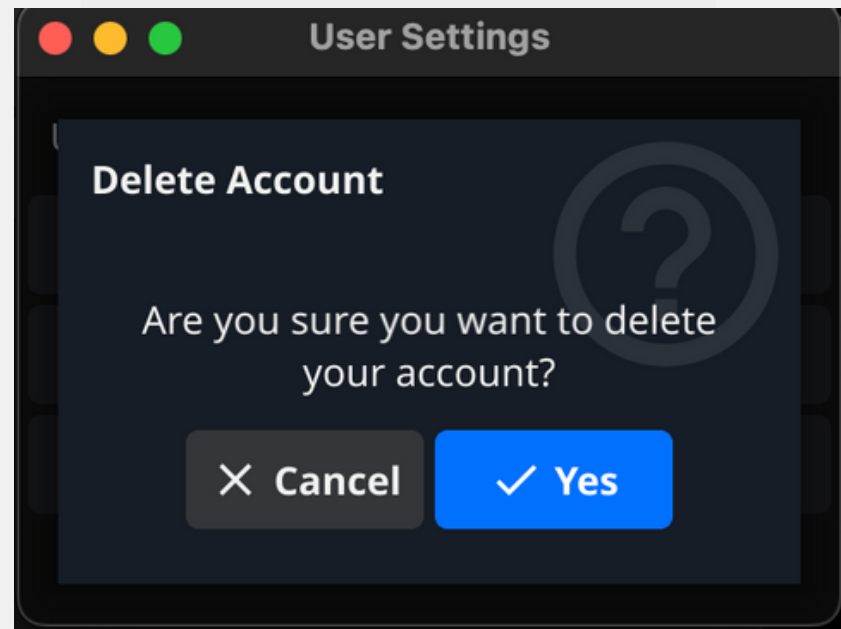
Cerrar sesión



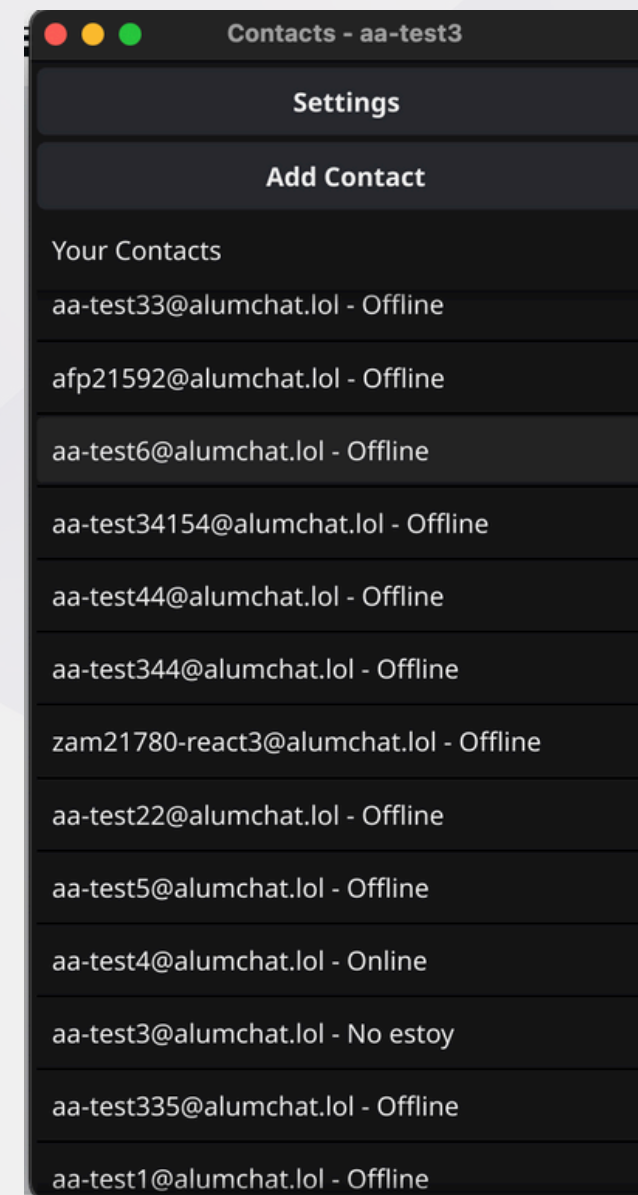
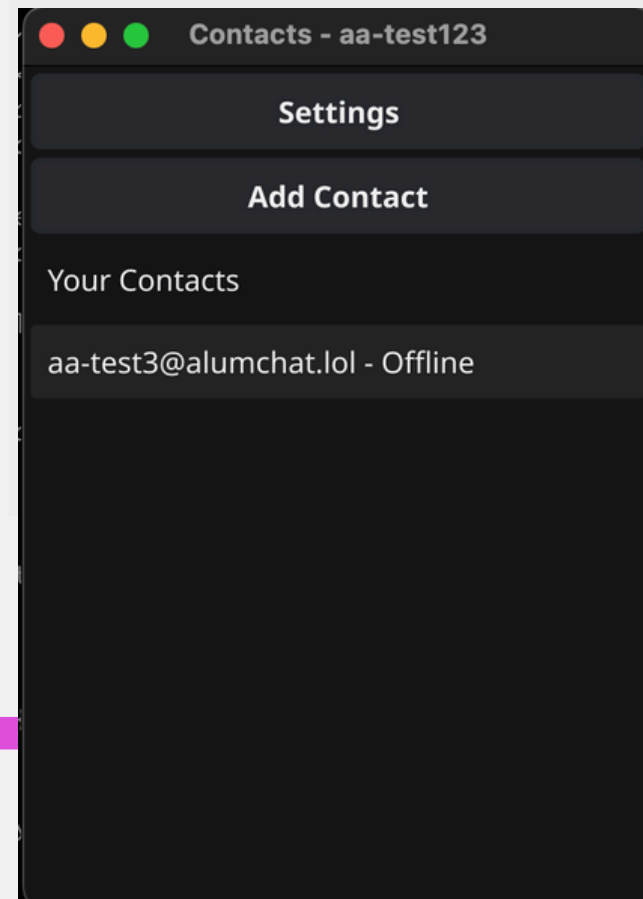
A dark-themed dialog box titled "User Settings" with standard macOS window controls. It contains the text "User Settings" and three buttons: "Change Presence", "Logout", and "Delete Account".

CARACTERÍSTICAS IMPLEMENTADAS

Eliminación de cuenta

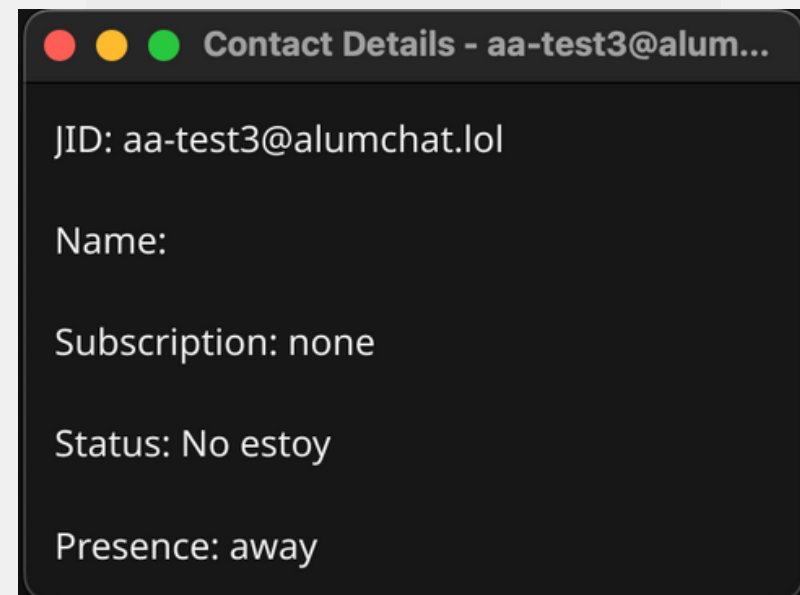


Mostrar contactos y estados

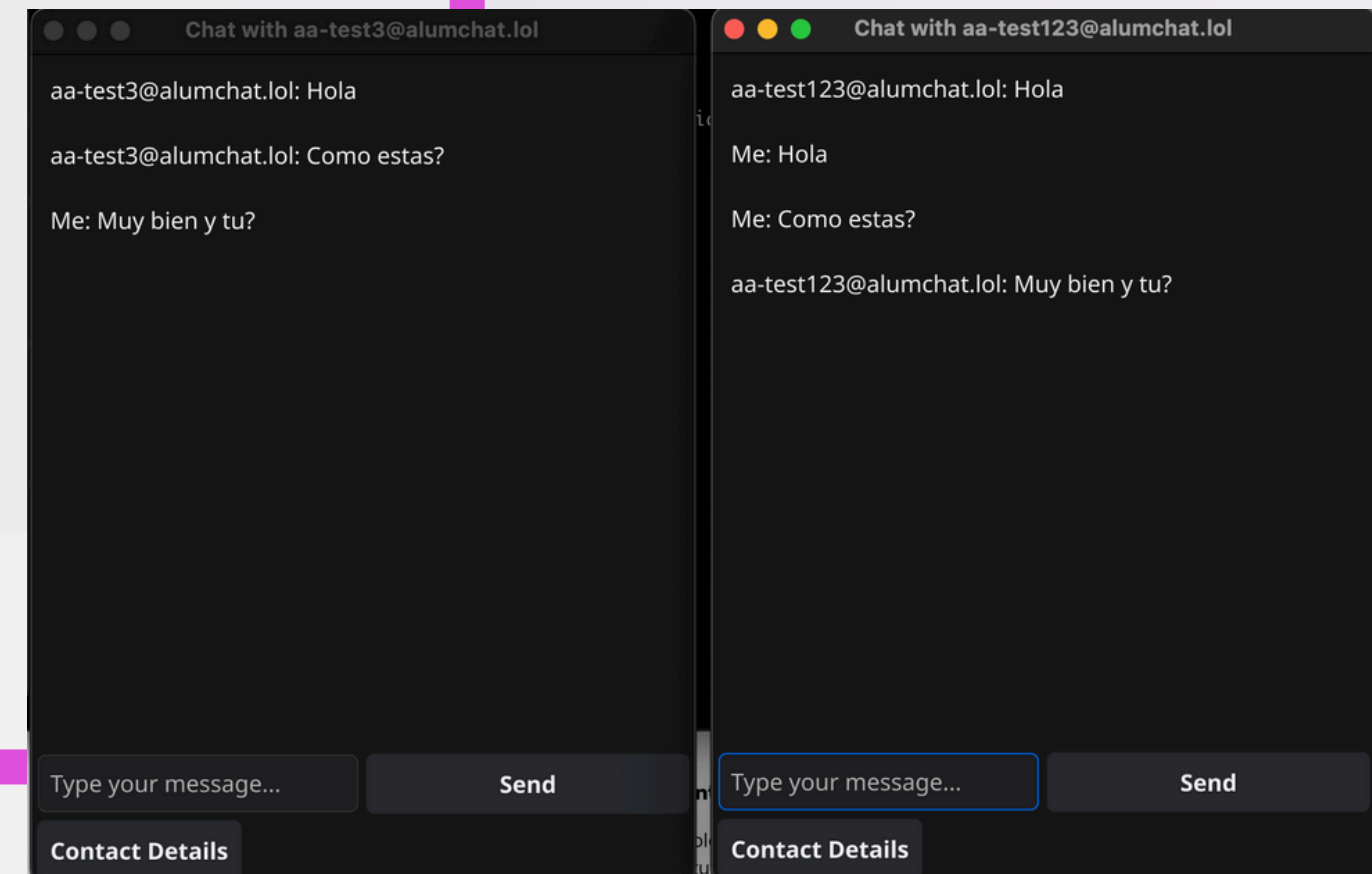


CARACTERÍSTICAS IMPLEMENTADAS

Mostrar detalles del contacto



Comunicación 1 a 1 con contacto



CARACTERÍSTICAS IMPLEMENTADAS

Adicionar contactos

Add Contact

est3@alumchat.lol - No estoy

Add a new contact

aa-test3@alumchat.lol

✕ Cancel **✓ Add**

Subscription Request

aa-test3@alumchat.lol wants to subscribe to your presence. Do you accept?

✕ Ignore **✓ Yes**

Definición de presencia en el servidor

Change Presence

Change Your Presence

chat

Estoy listo para hablar

Apply

Contacts - aa-test123

Settings

Add Contact

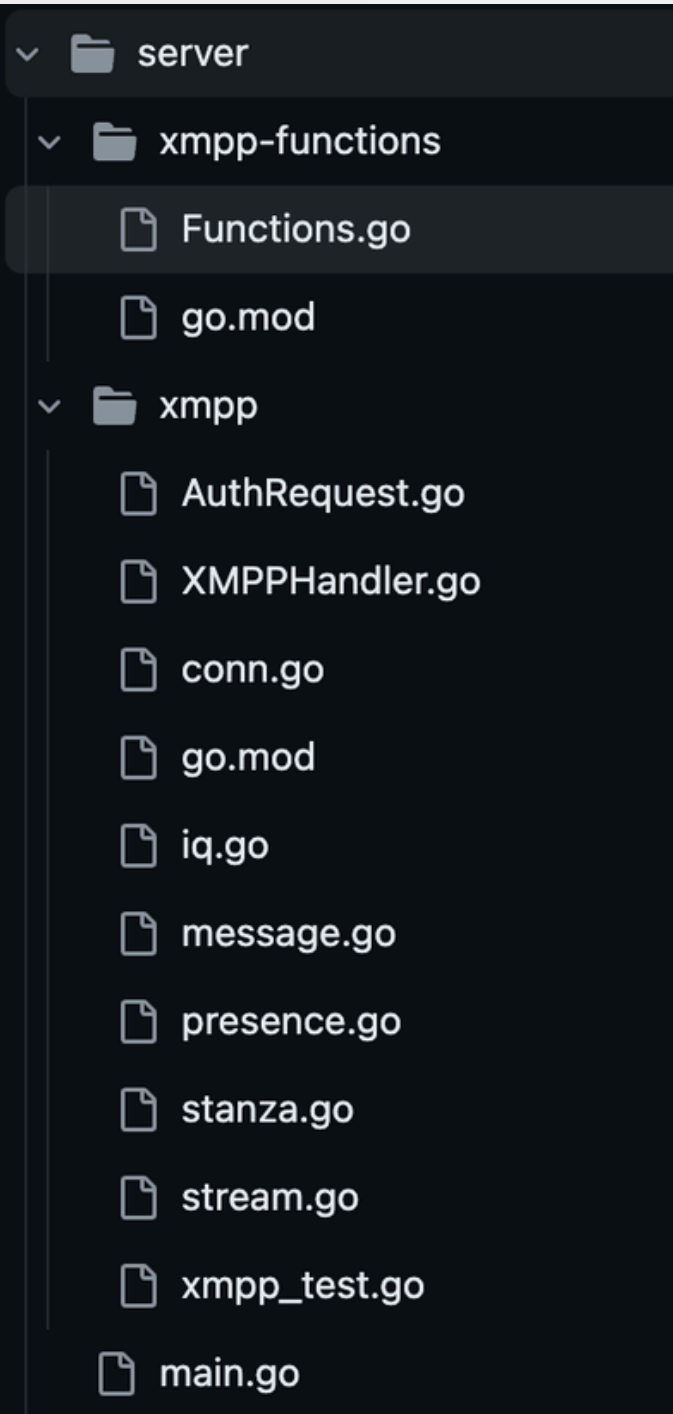
Your Contacts

aa-test3@alumchat.lol - Estoy listo para hablar

CARÁCTERÍSTICAS IMPLEMENTADAS

- Adicionalmente se implemento desde 0 el protocolo XMPP con el cual se realizan las comunicaciones con el servidor

```
+ Proyecto1-Redes git:(main) go run ./server
# github.com/adrianfulla/Proyecto1-Redes/server
ld: warning: ignoring duplicate libraries: '-lobjc'
2024/08/27 16:41:32 Received STARTTLS response: <?xml version="1.0" encoding="UTF-8"?><stream:stream xmlns:stream="http://etherx.jabber.org/streams" xmlns="jabber:client" from="alumchat.lol" id="12w7aa5l6v" xml:lang="en" version="1.0"><stream:features><starttls xmlns="urn:ietf:params:xml:ns:xmpp-tls"/><limits xmlns="urn:xmpp:stream-limits:0"><max-bytes>1048576</max-bytes><idle-seconds>360</idle-seconds></limits><mechanisms xmlns="urn:ietf:params:xml:ns:xmpp-sasl"><mechanism>PLAIN</mechanism><mechanism>SCRAM-SHA-1</mechanism><mechanism>CRAM-MD5</mechanism><mechanism>DIGEST-MD5</mechanism></mechanisms><compression xmlns="http://jabber.org/features/compress"><method>zlib</method></compression><ver xmlns="urn:xmpp:features:rosterver"/><register xmlns="http://jabber.org/features/iq-register"/><c xmlns="http://jabber.org/protocol/caps" hash="sha-1" node="https://www.igniterealtime.org/projects/openfire/" ver="Hqh03LBec1Amm2fLt3uyaXpq/Lo="/><limits xmlns="urn:xmpp:stream-limits:0"><max-bytes>1048576</max-bytes><idle-seconds>360</idle-seconds></limits></stream:features><proceed xmlns="urn:ietf:params:xml:ns:xmpp-tls"/>
2024/08/27 16:41:32 Proceeding with TLS handshake...
2024/08/27 16:41:32 TLS handshake successful
2024/08/27 16:41:32 Received initial response after STARTTLS: <?xml version="1.0" encoding="UTF-8"?><stream:stream xmlns:stream="http://etherx.jabber.org/streams" xmlns="jabber:client" from="alumchat.lol" id="12w7aa5l6v" xml:lang="en" version="1.0"><stream:features><mechanisms xmlns="urn:ietf:params:xml:ns:xmpp-sasl"><mechanism>PLAIN</mechanism><mechanism>SCRAM-SHA-1</mechanism><mechanism>CRAM-MD5</mechanism><mechanism>DIGEST-MD5</mechanism></mechanisms><compression xmlns="http://jabber.org/features/compress"><method>zlib</method></compression><ver xmlns="urn:xmpp:features:rosterver"/><register xmlns="http://jabber.org/features/iq-register"/><c xmlns="http://jabber.org/protocol/caps" hash="sha-1" node="https://www.igniterealtime.org/projects/openfire/" ver="Hqh03LBec1Amm2fLt3uyaXpq/Lo="/><limits xmlns="urn:xmpp:stream-limits:0"><max-bytes>1048576</max-bytes><idle-seconds>360</idle-seconds></limits></stream:features>
2024/08/27 16:41:32 Sending authentication stanza: <auth xmlns="urn:ietf:params:xml:ns:xmpp-sasl" mechanism="PLAIN">AGFhLXRlc3QxMjMAMMTIzNDU=</auth>
2024/08/27 16:41:32 Received authentication response: <success xmlns="urn:ietf:params:xml:ns:xmpp-sasl"/>
2024/08/27 16:41:32 Authentication successful
2024/08/27 16:41:32 Received resource binding response: <iq type="result" id="bind_1" to="alumchat.lol/5d1b74ad-5f55-4814-9612-b72c0662fbbf"><bind xmlns="urn:ietf:params:xml:ns:xmpp-bind"><jid>aa-test123@alumchat.lol/mainbinding</jid></bind></iq>
2024/08/27 16:41:32 Resource binding successful
Obtaining contacts, firstObtained response: <iq type="result" id="getRoster1" to="aa-test123@alumchat.lol/mainbinding"><query xmlns="jabber:iq:roster" ver="-1416812621"/></iq>
2024/08/27 16:41:32 Presence sent successfully
2024/08/27 16:41:32 Presence from aa-test123@alumchat.lol: Online!!
2024/08/27 16:41:37 Obtained IQ: {{ { iq } [{{ type } get} {{ id } 106-569286} {{ to } aa-test123@alumchat.lol/mainbinding} {{ from } alumchat.lol}]}
2024/08/27 16:41:37 Received IQ request without specific query from alumchat.lol
2024/08/27 16:41:37 Sent IQ response to alumchat.lol
2024/08/27 16:41:47 Contact added: aa-test3@alumchat.lol
Obtaining contacts, first2024/08/27 16:41:47 Obtained IQ: {{ { iq } [{{ type } set} {{ id } 817-569290} {{ to } aa-test123@alumchat.lol/mainbinding}]}
2024/08/27 16:41:47 Received IQ request without specific query from
2024/08/27 16:41:47 Sent IQ response to
Obtained response: <iq type="result" id="getRoster1" to="aa-test123@alumchat.lol/mainbinding"><query xmlns="jabber:iq:roster" ver="879931305"><item jid="aa-test3@alumchat.lo
```



DIFICULTADES

- La primer dificultad que encuentre fue la falta de una libreria de XMPP para el lenguaje de programación Go compatible con el servidor que utilizamos por lo que decidí implementarlo desde 0.
- Durante el desarrollo del protocolo encuentre dificultades al inicio para comprender como realizar efectivamente la comunicación con el servidor, una vez logre establecer esta el desarrollo del protocolo fue fluido.
- La dificultad más grande y la cual me tomo más tiempo fue la librería que elegí para realizar la GUI, esta se llama Fyne, la dificultad fue la falta de una documentación con ejemplos de implementaciones. Esta librería utiliza hilos por lo que el debuggeo no fue sencillo.



LECCIONES APRENDIDAS

01. Aumente mi conocimiento de protocolos y comunicación con servidores.

Al haber tomado el reto de crear la librería de XMPP desde 0 pude entender a profundidad como funciona este protocolo, como se realiza la comunicación por medio de TCP y los pasos para asegurar una comunicación estable con el servidor.

02. Aprendí más sobre Go

Aunque ya estaba familiarizado con el lenguaje, pude aprender más sobre el funcionamiento de este y logre sacarle el jugo a una de las características por lo que Go es tan útil que son las Gorutinas, funciones que se ejecutan de forma concurrente al programa, similar a hilos. Go permite el uso de estas junto con canales de comunicación para realizar múltiples funciones al mismo tiempo. Por ejemplo la obtención de mensajes que envía el servidor se puede realizar concurrentemente con el envío de mensajes y estos no se interfieren.



03. Descubrí como crear aplicaciones con Go

Al ser un lenguaje relativamente nuevo y con un objetivo que no es exactamente creaciones de interfaces de usuario, descubrí varias librerías que apoyan con esto. Elegí la librería Fyne ya que me gusto que era la más utilizada y tenía una documentación aparentemente amplia, también esta permite realizar empaquetamientos de manera que se pueden crear aplicaciones tanto para computadora como para móvil completamente usando Go.

04. Lecciones adicionales

Al mismo tiempo que aprendí las tres lecciones anteriores, también aprendí varias cosas que no se encuentran en la versión final del proyecto.

Aprendí a manejar contenedores de Docker y la conexión de Go con un frontend en Vue.js, esta era la idea original con la que empecé el proyecto pero preferí mantener todo en un solo lenguaje y determine que el uso de Docker terminaba siendo innecesario.



**MUCHAS
GRACIAS**