**MANUAL DE USUARIO APLICACIÓN CHAT DISTRIBUIDO (ChatApp)**

1. **¿Qué es ChatApp?**

**ChatApp** es una aplicación de mensajería en tiempo real desarrollada en Elixir bajo un modelo cliente-servidor. Está diseñada para permitir la comunicación simultánea entre múltiples usuarios a través de una red local o entorno distribuido.

Los usuarios pueden interactuar mediante salas de conversación, enviar mensajes, cambiar de sala, consultar los usuarios conectados, y más. Toda la interacción se realiza desde la terminal.

Esta aplicación ha sido pensada para prácticas de programación concurrente y sistemas distribuidos.

1. **Requisitos**

| **Requisito** | **Descripción** |
| --- | --- |
| Lenguaje | Elixir 1.14 o superior |
| Sistema operativo | Linux o MacOs (Windows a través de WSL o similar) |
| Terminal | Bash / Zsh / Fish con soporte para scripts .sh |
| Editor (opcional) | Visual Studio Code, Neovim, Emacs u otro editor Elixir |

1. **Instalación y preparación**

**3.1.** Clonar o descomprimir el proyecto.

**3.2.** Entrar a la carpeta del proyecto.

**3.3.** Instalar VS CODE, Erlang y Elixir.

**3.4.** Verificar los ip del servidor y de los clientes.

1. **¿Cómo iniciar?**

**4.1.** Abre dos terminales.

**4.2.** En una, ejecuta el servidor:

*iex --name cliente1@op\_cliente --cookie chat\_distribuido\_secreto -S mix*

*ChatApp.iniciar\_servidor()*

**4.3.** En la otra, ejecuta el cliente:

*iex --name chat\_server@ip\_servidor --cookie chat\_distribuido\_secreto -S mix*

*ChatApp.conectar(“ip\_servidor”)*

**Los comandos se ejecutan en command prompt en VS CODE.**

**4.4.** Se solicitará nombre de usuario, después de esto se conectará al servidor y entrará a la sala general por defecto.

1. **Comandos disponibles en el chat**

Una vez conectado, puedes usar los siguientes comandos:

| **Comando** | **Acción** |
| --- | --- |
| /list | Lista todos los usuarios conectados actualmente |
| /join <sala> | Cambia de sala o crea una nueva si no existe |
| /salas | Lista todas las salas disponibles |
| /exit | Cierra sesión y termina tu conexión con el servidor |
| /historial | muestra el historial de la sala actual |
| <mensaje> | Envía un mensaje a todos los usuarios conectados en la misma sala |
| /actual | Muestra en qué sala se encuentra actualmente |
| /clear | Limpia el historial de la pantalla (solo en la terminal local) |

1. **Persistencia de datos**

La aplicación guarda automáticamente:

* Mensajes enviados y su historial.
* Salas creadas.

1. **Problemas comunes**

| **Problema** | **Solución** |
| --- | --- |
| No conecta | Verifica que el servidor esté iniciado |
| No guarda historial | Asegúrate de haber creado una sala primero |
| El comando no responde | Usa comandos válidos. Evita escribir texto antes de /join |

1. **Notas**

* Todo se guarda automáticamente. Al salir y volver a entrar, el historial y las salas se conservan.
* No necesitas abrir múltiples programas, todo se hace desde la terminal.
* Puedes ejecutar múltiples clientes desde diferentes terminales o incluso desde otras máquinas en red.

1. **Créditos**

* Desarrollado por Valentina Rodríguez, Leidy Vanesa Muñoz y Raquel López, para ComNet Solutions como parte del proyecto de plataforma de mensajería distribuida en Elixir.