



Materia: Cómputo Distribuido

Profesor:

Dr. Juan Carlos López Pimentel

Estudiantes:

Mario Alejandro Rodríguez González

(0235810@up.edu.mx)

Emiliano Hinojosa Guzmán(0252496@up.edu.mx)

Jesus Abel Gutierrez Calvillo (0244814@up.edu.mx)

Evidencia: Avance del Proyecto

Nombre de la actividad: Avance Proyecto Final

Fecha de entrega:

22 de mayo de 2025

1. Lista de Actividades Realizadas

Se definió y modeló la arquitectura general del sistema backend en C, basada en una estructura modular y orientada a acciones JSON.

Se implementó una conexión estable y segura con MySQL utilizando variables de entorno.

Se crearon los principales módulos del sistema:

user_manager.h: manejo de usuarios.

chat_manager.h: gestión de chats, participantes y mensajes.

Se desarrollaron las acciones principales de la API:

Validación de usuarios por clave (key).

Registro de nuevos usuarios con validación de nombre y correo.

Obtención de datos de usuario desde la base de datos.

Creación de chats (grupales o privados).

Agregado de participantes a chats grupales.

Envío de mensajes (texto u otros tipos).

Consulta de chats en los que participa un usuario, con último mensaje.

Se diseñó y normalizó la base de datos en MySQL con las tablas necesarias: users, chats, chat_participants y messages.

2. Arquitectura General del Proyecto

El sistema backend está desarrollado en el lenguaje C y se comunica con una base de datos relacional MySQL. La arquitectura del sistema sigue un enfoque modular, orientado a acciones que se reciben en formato JSON y son interpretadas para ejecutar funciones específicas como creación de usuarios, validación de credenciales, gestión de chats, etc.

Componentes principales:

Librerías externas: Uso de JSON para parsing de datos JSON.

Conexión a MySQL: Utiliza la API nativa de mysql/mysql.h para ejecutar queries y manejar resultados.

Módulos del sistema:

user_manager.h: Para la gestión de usuarios.

chat_manager.h: Para la gestión de chats, participantes y mensajes.

Estructura de acciones: El backend interpreta un campo "action" recibido en formato JSON y ejecuta la lógica correspondiente mediante un switch.

3. Acciones implementadas (API)

0. Validar usuario

Verifica si una llave (key) de usuario existe en la base de datos.

Retorna el password_hash correspondiente.

2. Crear usuario

Recibe username, email y password y almacena el usuario en la base de datos.

Verifica unicidad en nombre y correo electrónico.

3. Obtener información de usuario

Recibe una llave de usuario y devuelve su user_id, username y email.

4. Crear chat

Soporta la creación de chats grupales o privados.

El creador es automáticamente agregado como participante.

5. Agregar participantes a un grupo

Añade múltiples usuarios a un chat grupal.

6. Enviar mensaje

Registra mensajes asociados a un chat_id y sender_id.

Almacena el contenido y tipo de mensaje.

7. Obtener chats de un usuario

Retorna todos los chats donde participa un usuario.

Incluye nombre del chat, último mensaje, author, tipo y timestamp.

4. Diseño de Base de Datos

El sistema cuenta con una base de datos relacional bien normalizada, diseñada para representar usuarios, chats, participantes y mensajes.

Tablas principales:

Users:

Contiene información única de cada usuario.
Claves únicas: username, email.

Chats:

Define si el chat es grupal o privado.
Guarda una referencia al último mensaje (last_message_id).

Chat_participants:

Tabla puente entre users y chats.
Define también si un usuario es administrador de un grupo.

Messages:

Almacena contenido, tipo de mensaje y emisor.
Relaciona cada mensaje con un chat y un usuario.

5. Conexión a Base de Datos

La conexión a la base de datos se realiza utilizando variables de entorno para proteger las credenciales:

```
const char *server = getenv("DB_HOST");  
const char *user = getenv("DB_USER");  
const char *password = getenv("DB_PASS");  
const char *database = getenv("DB_NAME");
```

Esto promueve buenas prácticas de seguridad y permite una configuración dinámica del entorno.

6. Estado Actual del Proyecto

Módulo	Observaciones
--------	---------------

Conexión a MySQL	Estable y funcional
Validación de usuario	Devuelve password_hash según key
Registro de usuario	Almacena usuario tras validación básica
Info de usuario	Se obtiene correctamente desde la DB
Creación de chat	Soporta grupos y mensajes directos
Participantes grupo	Agrega múltiples usuarios a un chat
Envío de mensajes	Soporte para texto y otros tipos
Obtener chats	Devuelve chats y últimos mensajes

7. Conclusión

El backend está funcionalmente completo en cuanto a la lógica principal de manejo de usuarios, creación de chats y mensajes. La arquitectura es clara y está bien organizada para escalar con más funcionalidades más adelante.

Ya tenemos la conexión entre cliente -> server lógico -> server de datos y de regreso

Trabajar en:

Realizar pruebas de carga y casos de uso más complejos.

Refinar la seguridad de las acciones (por ejemplo, mediante tokens o autenticación).