

TurboMessage – Documentación

Introducción

TurboMessage es un sistema distribuido de mensajería local basado en arquitectura cliente-servidor con comunicación gRPC. Permite a múltiples usuarios enviar y recibir correos electrónicos simulados dentro de una red local, incluyendo autenticación, gestión de bandeja de entrada y salida, marcado de correos leídos, y eliminación lógica de mensajes. Además, el cliente incluye una interfaz gráfica desarrollada con tkinter, facilitando la experiencia de uso sin requerir comandos de consola.

Arquitectura del sistema

El sistema está compuesto por **cuatro capas principales**:

1. Cliente (GUI + CLI)

- Desarrollado en Python con tkinter y consola interactiva.
- Permite al usuario registrarse, iniciar sesión, redactar mensajes, consultar su bandeja de entrada o salida, y eliminar o marcar correos como leídos.
- La GUI se organiza en ventanas modales para login, registro, menú principal, visualización y redacción.

2. Servidor gRPC

- Implementado con `grpcio` y `concurrent.futures.ThreadPoolExecutor`.
- Expone dos servicios definidos en `.proto`:
 - Autenticador: autenticación y registro.
 - Mensajería: operaciones sobre correos.
- El servidor valida tokens por sesión y mantiene seguridad básica en memoria (tokens UUID4 únicos por usuario).

3. Base de datos de usuarios (*usuarios.py*)

- Usa SQLite y almacena hashes SHA-256 de las contraseñas.
- Permite registrar nuevos usuarios y autenticar credenciales.

4. Base de datos de mensajes (*mensajes_db.py*)

- También basada en SQLite.

- Almacena correos con metadatos: remitente, destinatario, asunto, contenido, fecha y estado de lectura.
- Implementa eliminación lógica con los campos `eliminado_entrada` y `eliminado_salida`, borrando físicamente un mensaje solo si ambos usuarios lo eliminaron.

Comunicación

- El cliente se comunica con el servidor exclusivamente por llamadas RPC (gRPC).
- Los mensajes se estructuran usando Protocol Buffers (`mensaje.proto`), lo que permite una serialización eficiente.
- Los métodos disponibles incluyen:
 - Autenticación y registro.
 - Obtener bandejas de entrada y salida.
 - Leer correos enviados y recibidos
 - Enviar, eliminar y marcar correos como leídos.

Conclusiones

TurboMessage demuestra una arquitectura distribuida funcional, combinando protocolos modernos (gRPC) con una lógica robusta de manejo de sesiones y almacenamiento persistente. El uso de SQLite simplifica la persistencia sin necesidad de servidores externos. La interfaz gráfica facilita la interacción, mientras que el diseño modular permite extender la funcionalidad (por ejemplo, añadir filtros, archivos adjuntos o tokens persistentes).

Este proyecto pone en práctica los conceptos fundamentales de RPC, concurrencia, separación de responsabilidades y diseño orientado a servicios, cumpliendo con los objetivos de un sistema distribuido educativo con funcionalidades completas y una experiencia de usuario razonablemente intuitiva.