

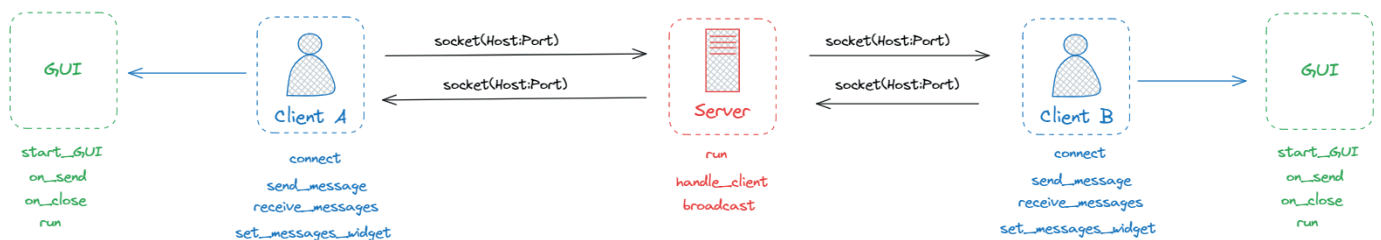
Descripción

El proyecto consiste en desarrollar una aplicación cliente-servidor para una sala de chat. Los clientes se conectarán al servidor a través de WebSockets, permitiendo una comunicación en tiempo real. La aplicación usará mecanismos de concurrencia y asincronismo para manejar múltiples conexiones de clientes de manera eficiente. Redis se usará para la gestión de sesiones y la cola de tareas distribuidas, mientras que PostgreSQL se utilizará para el almacenamiento persistente de mensajes.

Problemática a Solucionar

El objetivo es permitir que múltiples usuarios puedan comunicarse en tiempo real a través de una aplicación de chat, gestionando eficientemente las conexiones y el almacenamiento de datos.

Arquitectura del Proyecto



Componentes

Cliente WebSocket: Envío y recepción de mensajes en tiempo real.

Servidor WebSocket: Manejo de conexiones de clientes y distribución de mensajes.

Base de Datos PostgreSQL: Almacenamiento persistente de mensajes.

Redis: Gestión de sesiones y cola de tareas distribuidas.

Tareas Asíncronas: Manejo de tareas en segundo plano usando Celery.

Funcionalidades de Cada Entidad

Cliente

Enviar mensajes.

Recibir mensajes en tiempo real.

Conectarse y desconectarse del servidor.

Servidor

Aceptar conexiones de clientes.

Recibir y distribuir mensajes.

Almacenar mensajes en PostgreSQL.

Publicar mensajes en Redis.

Ejecutar tareas en segundo plano.

Justificación de las Herramientas Utilizadas

WebSockets: Permiten la comunicación bidireccional en tiempo real entre el cliente y el servidor.

Redis: Facilita la gestión de sesiones y colas de tareas distribuidas.

PostgreSQL: Proporciona un almacenamiento persistente y confiable de los datos del chat.

Celery: Permite ejecutar tareas en paralelo, mejorando la eficiencia y la capacidad de respuesta del sistema.

asyncio: Facilita la programación asíncrona de I/O, lo que permite manejar múltiples conexiones de manera eficiente.