

ChatTorrent P2P: Aplicación de Comunicación Multimedia Descentralizada

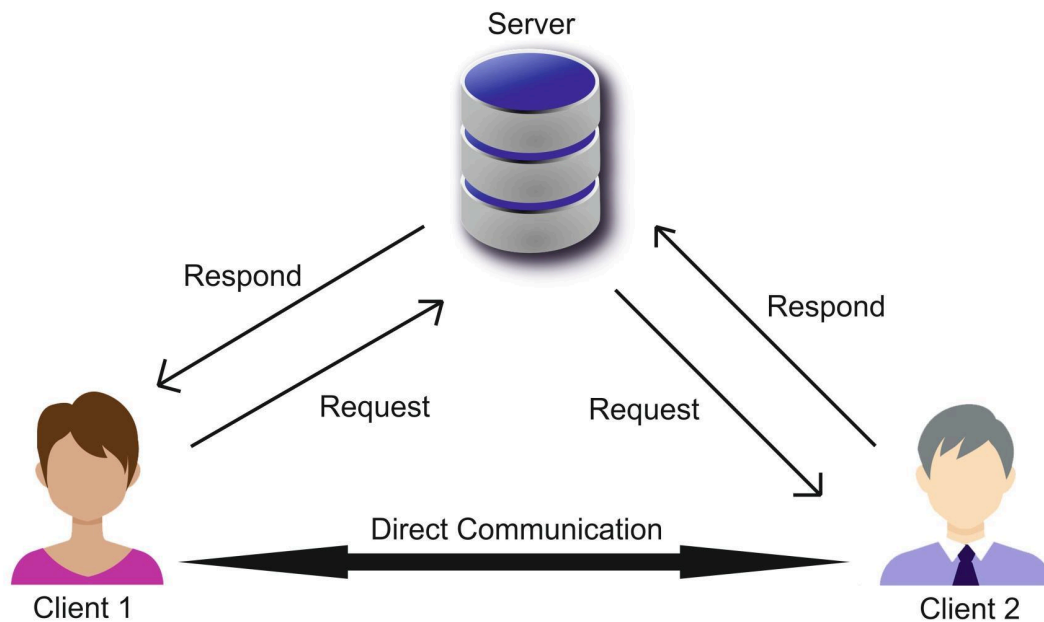
Visión General del Proyecto

ChatTorrent P2P es una prueba de concepto (PoC) que demuestra una aplicación de chat multimedia en tiempo real utilizando el principio de comunicación **Peer-to-Peer (P2P)** basado en **WebRTC (Web Real-Time Communication)**. El proyecto simula el comportamiento de una red Torrent, donde los usuarios se conectan directamente sin necesidad de un servidor central para el intercambio de datos. Este enfoque asegura la privacidad, la baja latencia en la transferencia de archivos y la resistencia a fallos de infraestructura centralizados.

Concepto y Arquitectura P2P

El núcleo de ChatTorrent reside en la capacidad de establecer una conexión directa entre navegadores, conocida como **DataChannel** de WebRTC.

- **Descentralización:** Una vez establecida la conexión, los mensajes, imágenes y grabaciones de voz viajan directamente de un Peer a otro.
- **Simulación Torrent (Chunking):** La transferencia de archivos y grabaciones se realiza dividiendo la data en pequeños fragmentos (chunks de 16KB) a través del DataChannel, imitando cómo BitTorrent maneja la distribución de archivos grandes.
- **Historial Append-Only:** El historial de la sala es de solo anexar, lo que garantiza que los mensajes enviados no pueden ser modificados o eliminados, manteniendo la integridad del registro.



WebRTC Working

Shutterstock
Explorar

🔑 Casos de Uso Clave

1. **Privacidad y Seguridad:** Al eliminar servidores intermediarios, se reduce el riesgo de interceptación masiva o almacenamiento de metadatos del usuario.
2. **Transferencia de Archivos de Gran Volumen:** Ideal para el intercambio de archivos grandes, ya que la velocidad de transferencia depende directamente de la capacidad de ancho de banda entre los dos Peers.
3. **Resistencia a Fallos (Fault Tolerance):** La comunicación no se interrumpe si un servidor central (de chat tradicional) cae.
4. **Comunicación en Zonas de Infraestructura Limitada:** Funcional para redes locales o ambientes donde la conectividad a servidores externos es inestable o costosa.



Guía de Uso Paso a Paso (Flujo de Conexión)

Para establecer una sala de chat y compartir contenido, se requiere el intercambio inicial de una "llave" de conexión (el archivo .torrent con el SDP de WebRTC).

Requisitos

- Abrir la aplicación en **dos ventanas o pestañas diferentes de Chrome** (simulando Cliente A y Cliente B).

Cliente A (El Creador de la Sala - Oferente)

Paso	Acción	Resultado
1. Crear Sala	Introduzca un nombre en el campo "Nombre de la nueva sala" y haga clic en " 1. Crear Sala & Exportar .torrent ".	El Navegador A inicia la conexión P2P, genera la Oferta SDP y descarga automáticamente el archivo [Nombre_Sala]_offer.torrent.
2. Intercambio	Comparta este archivo (_offer.torrent) con el Cliente B (por email, USB, etc.).	El Navegador A se queda en estado " Esperando " la Respuesta.
3. Conexión	Importe el archivo _answer.torrent recibido del Cliente B (usando " 2. Importar Archivo .torrent ").	La conexión P2P se establece. El historial se sincroniza y la sala está lista para chatear.

Cliente B (El Invitado - Respondedor)

Paso	Acción	Resultado
1. Importar Oferta	Haga clic en " 2. Importar Archivo .torrent " y seleccione el archivo _offer.torrent recibido del Cliente A.	La aplicación crea la sala localmente, genera la Respuesta SDP y descarga automáticamente el archivo [Nombre_Sala]_answer.torrent.
2. Intercambio	Comparta este archivo (_answer.torrent) con el Cliente A.	El Navegador B queda en estado " Esperando " que el Cliente A aplique la respuesta.
3. Conexión	Una vez que el Cliente A importa la Respuesta, la conexión P2P se establece automáticamente en el Cliente B.	La sala está lista para chatear.



Funcionalidades Clave Implementadas

- **P2P Real (WebRTC):** Conexión directa DataChannel entre Peers.
- **Sincronización de Historial:** El Peer Oferente (Cliente A) sincroniza automáticamente el historial completo al Peer Respondedor (Cliente B) al establecer la conexión, garantizando que el invitado tenga todos los mensajes previos.
- **Transferencia Multimedia:**
 - **Imágenes:** Vista previa de imágenes integrada en la burbuja de chat.

- **Grabación de Voz:** Grabación de audio en tiempo real desde el micrófono y envío como archivo (Blob) a través del DataChannel.
 - **Archivos Genéricos:** Transferencia de archivos de cualquier tipo, divididos en *chunks* (simulación BitTorrent).
- **Persistencia Local:** El historial de la sala se mantiene en el localStorage del navegador, permitiendo a los usuarios continuar las conversaciones después de cerrar y volver a abrir la aplicación.
- **Exportación de Historial (JSON):** Se puede exportar el historial de cualquier sala a un archivo JSON para auditoría o copia de seguridad (botón "**Exportar Historial (JSON)**").