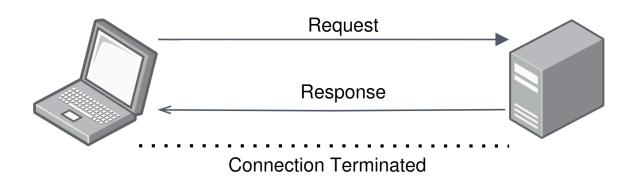
# **Poc Desarrollo**

### Que es/son los websockets y como funciona?

WebSockets, o WS para mayor facilidad. Es un protocolo de comunicación bidireccional, persistente y a tiempo real, por medio de una única conexión TCP\*.

\*TCP, o **Protocolo de Control de Transmisión**, es un protocolo fundamental en las redes de computadoras que garantiza la entrega confiable de datos entre dispositivos.

Normalmente la comunicación entre cliente y servidor se basa en una serie de peticiones y respuestas, ya sea para enviar o solicitar información al servidor y que el mismo realiza una respuesta



Este esquema para algunos tipos de aplicaciones, como mensajería u otras aplicaciones las cuales dependían de una actualización de datos constantes tiene los siguientes problemas:

• El servidor era forzado a establecer muchas conexiones TCP para cada cliente. En el caso de una aplicación de mensajera, el servidor tiene que

manejar las conexiones hechas para sincronizar los mensajes de cada usuario. Es decir recibir el mensaje enviado por uno y enviarlo al otro. Con el esquema de peticiones y respuestas, para subir y recibir cada mensaje, era necesario hacer una conexión TCP.

- A su vez, cada conexión, a nivel de red, se encontraba sobrecargada ya que cada mensaje del cliente al servidor incluye un encabezado HTTP.
- El cliente se ve obligado a mantener una correspondencia entre las conexiones salientes y las conexiones entrantes para poder rastrear las respuestas.

Gracias a los WebSockets estos problemas se resuelven, ya que solo se establece una conexión entre el servidor y el cliente por donde se hará la comunicación.

Primeramente el cliente realiza una petición al servidor, solicitando el establecimiento de la conexión. El servidor responde, y si es afirmativa la conexión se establece el puente para la comunicación y transferencia de datos de forma bidireccional.

Ambos, el cliente y el servidor, pueden enviar y recibir mensajes de manera independiente. Significa que ninguno esta activamente esperando a que llegue un mensaje para poder realizar una acción.



El handshake es el proceso inicial de establecer una conexión de WebSocket entre un cliente y un servidor.

## Flujo de un WebSocket

- 1. El cliente envía una petición HTTP Upgrade (es un mecanismo en HTTP/1.1 que permite cambiar a un protocolo diferente (o una versión más nueva) durante una conexión existente, funciona como un GET especial)
- 2. El servidor acepta y actualiza a protocolo WebSocket
- 3. Se establece una conexión con persistencia
- 4. Los mensajes ahora pueden enviarse en ambas direcciones con una sobrecarga mínima

# En donde se puede aplicar?

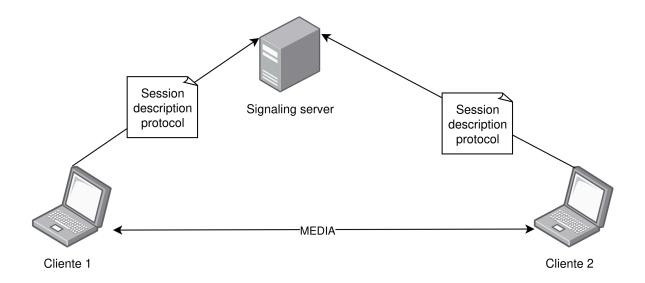
El protocolo WS es altamente usado hoy en día. Principalmente en juegos y aplicaciones de mensajería, como lo pueden ser WhatsApp y Telegram.

Algunos servicios utilizan este método para poder brindar datos de forma actualizada. Como lo pueden ser aplicaciones/servicios de finanzas o criptomonedas.

Otros ejemplos de uso de esta tecnología pueden ser la edición colaborativa de documentos (ej. Google Docs).

#### ¿Qué es WebRTC y cómo funciona?

WebRTC (Web Real-Time Communication) es una tecnología y conjunto de protocolos que permite la comunicación **en tiempo real** directamente entre navegadores o aplicaciones, sin necesidad de un servidor intermedio para la transmisión de datos (aunque sí requiere uno para la *señalización inicial*).



La mayoría de los dispositivos (celulares, computadoras ,etc) se encuentran detrás de un firewall o sus IPs cambian contentamente. Es por esto que se utiliza un estándar conocido como: Interactive Connectivity Establishment (ICE).

Este estándar facilita la coordinación entre clientes para poder establecer conexiones entre ellos.

Cada cliente genera un "reporte" o ICE candidate, en donde se indica la IP y puerto que se puede usar para establecer una conexión. Estos luego son enviados a enviados a uno o mas STUN (Session Transversal Utilities for Nat) servers.

Los STUN servers pueden ser consultados para poder establecer las IPs y puertos para poder generar los Session Description Protocols que se envian al Signaling Server para poder establecer la conexión.

Está optimizado para **audio, video y datos** con **baja latencia**, y utiliza principalmente protocolos basados en UDP para mejorar la velocidad y reducir retrasos.

Para funcionar, WebRTC emplea:

- **STUN**: Determina la IP pública y el puerto accesible del cliente.
- TURN: Reenvía datos cuando no es posible la conexión directa por restricciones de red (NAT estrictos).

• SRTP/DTLS: Proporcionan encriptación y seguridad en la transmisión.

#### Comparación WebSockets vs WebRTC

Característica	WebSockets	WebRTC
Tipo de conexión	Cliente-Servidor	Peer-to-Peer (con señalización)
Protocolo base	TCP (ws/wss)	UDP (SRTP, DTLS)
Optimizado para	Datos en tiempo real (texto, binario)	Audio, video y datos
Latencia	Baja (50-200 MS)	Muy baja (<50 MS)
Servidor necesario	Siempre	Solo para señalización / relay
Complejidad de implementación	Baja	Alta
Casos de uso típicos	Chats, notificaciones, juegos online simples	Videollamadas, streaming, compartición de pantalla

#### Conclusión

- WebSockets: ideales para datos en tiempo real de cualquier tipo, especialmente cuando hay un servidor central que gestiona la comunicación.
- **WebRTC**: la opción preferida para audio/video en vivo y datos con latencia mínima, aprovechando conexiones P2P para optimizar el rendimiento.

#### Referencias

#### RFC 6455: The WebSocket Protocol

The WebSocket Protocol enables two-way communication between a client running untrusted code in a controlled environment to a remote host that has opted-in to

https://datatracker.ietf.org/doc/html/rfc6455



#### The WebSocket API (WebSockets) - Web APIs | MDN

The WebSocket API makes it possible to open a two-way interactive communication session between the user's browser and a server. With this API, you can send messages

M https://developer.mozilla.org/en-US/docs/Web/API/Web Sockets\_API



https://developer.mozilla.org/es/docs/Web/API/WebRTC\_API

https://webrtc.org/getting-started/overview?hl=es-419