

WEBSOCKETS

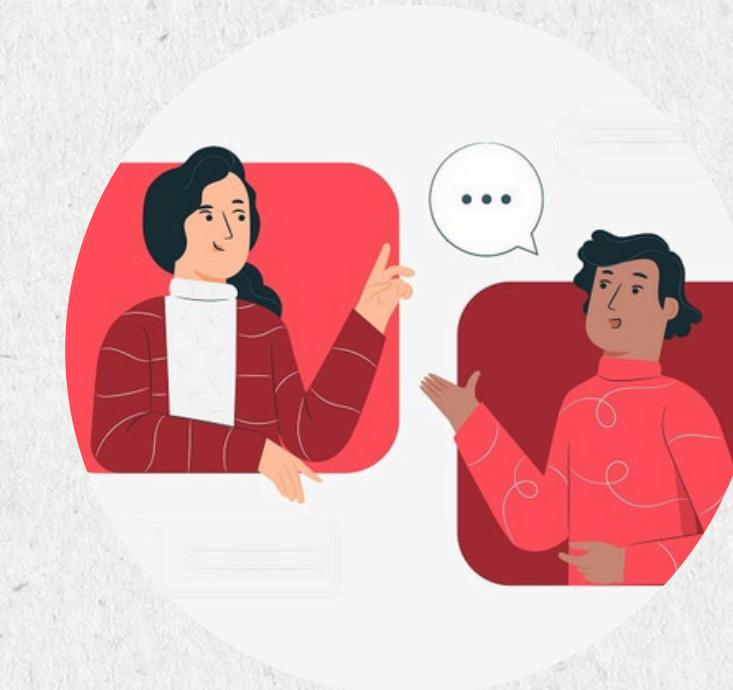
¿QUE SON?

WebSocket es un protocolo de comunicación persistente y bidireccional (Full-Duplex) sobre una única conexión TCP

Persistente: La conexión no muere tras la respuesta



Full-Duplex: Tráfico en ambos sentidos simultáneamente

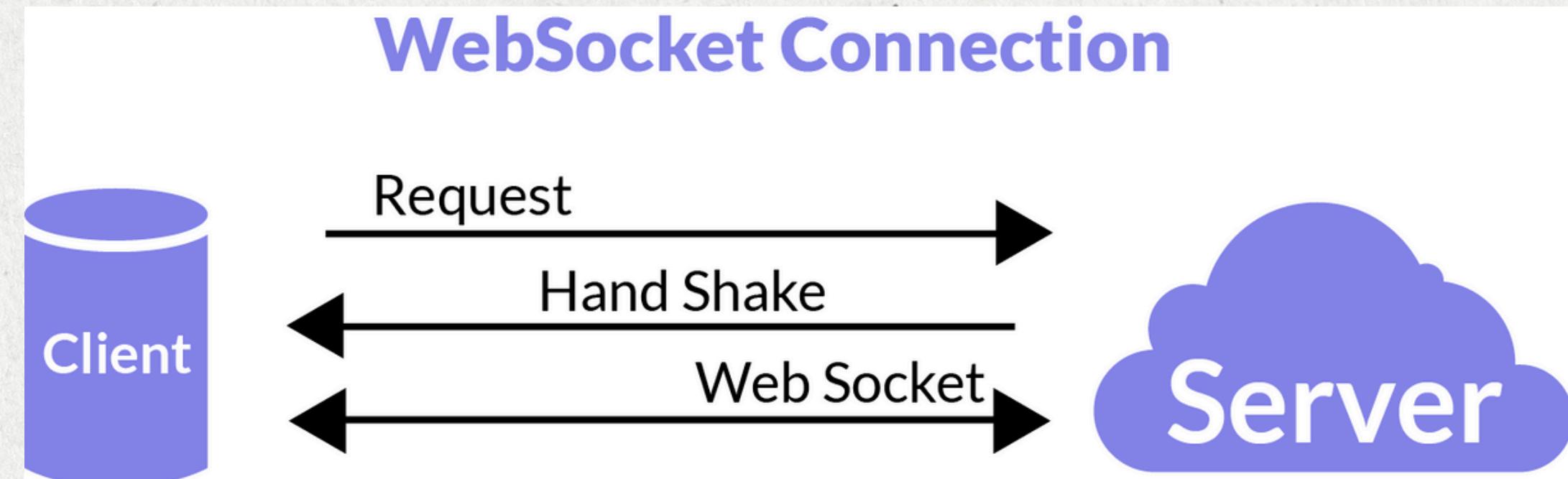


Low-Latency: Elimina la carga de crear conexiones HTTP repetidas.



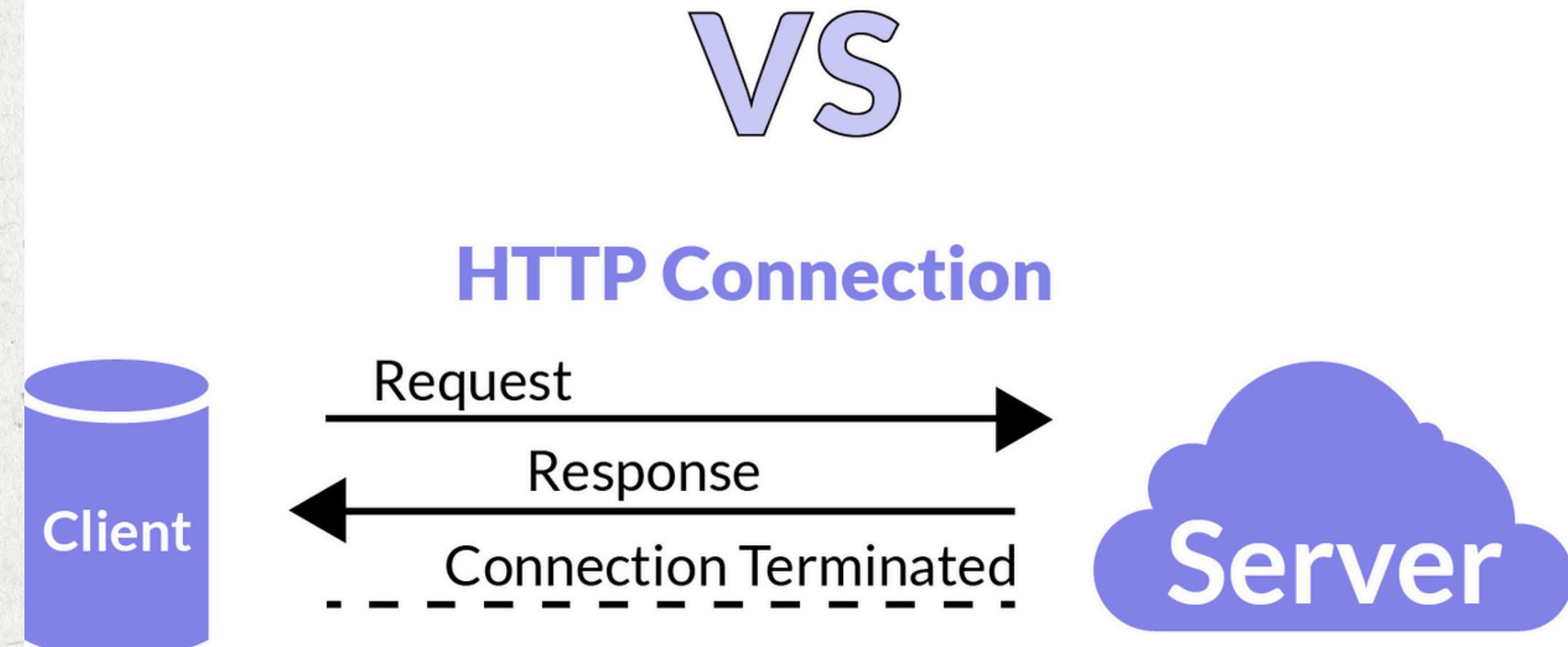
WEBSOCKET VS HTTP

WebSocket Connection



VS

HTTP Connection



Inicio: El cliente envía un HTTP Request normal.

Header: Upgrade: websocket

PASOS

Acuerdo: El servidor acepta el cambio.

Status: 101 Switching Protocols

Resultado:

- La conexión TCP no se cierra.
- HTTP desaparece, queda un túnel TCP persistente.

GUIA DE CODIGO

```
● ● ●  
1 const io = new Server(server, {  
2   cors: {  
3     origin: "http://localhost:5175",  
4     methods: ["GET", "POST"]  
5   }  
6 });
```

```
● ● ●  
1 socket.on('chat-message', (data: { username: string; text: string }) => {  
2   const message = {  
3     id: `${socket.id}-${Date.now()}`,  
4     username: data.username,  
5     text: data.text,  
6     timestamp: new Date()  
7   };  
8   io.emit('message', message);  
9 });
```

```
● ● ●  
1 socket.on('join', (username: string) => {  
2   currentUsername = username;  
3   console.log(`${username} joined the chat`);  
4   socket.broadcast.emit('user-joined', username);  
5 });
```

```
● ● ●  
1 server.listen(3050, () => {  
2   console.log('Server Running in http://localhost:3050');  
3 });
```

```
● ● ●  
1 const newSocket = io('http://localhost:3050')  
2   socketRef.current = newSocket  
3  
4   newSocket.on('message', (msg: Message) => {  
5     setMessages(prev => [...prev, msg])  
6   })
```

¡GRACIAS!

Agencia Borcelle

@un sitiogenial