



WebSocket

SISTEMAS DISTRIBUIDOS

Carlos Rojas Sánchez

Licenciatura en Informática

Universidad del Mar

1. Introducción a WebSocket
2. Cómo Funciona WebSocket
3. WebSocket en Node.js

Introducción a WebSocket

WebSocket es un protocolo que permite la comunicación en tiempo real y bidireccional entre el cliente y el servidor a través de una única conexión persistente.

Comparación de Protocolos: HTTP, WebSocket y CORBA

Característica	HTTP	WebSocket	CORBA
Tipo de Comunicación	Unidireccional	Bidireccional	Bidireccional
Conexión	Nueva conexión por solicitud	Conexión persistente	Conexión persistente

Table 1: Comparación entre HTTP, WebSocket y CORBA (Parte 1)

Comparación de Protocolos: HTTP, WebSocket y CORBA

Característica	HTTP	WebSocket	CORBA
Latencia	Mayor	Baja	Baja (pero más pesada que WebSocket)
Protocolo	Basado en texto	Basado en texto/binario	Binario
Escalabilidad	Escalable para contenido estático	Alta escalabilidad en tiempo real	Menos escalable por su complejidad

Table 2: Comparación entre HTTP, WebSocket y CORBA (Parte 2)

Comparación de Protocolos: HTTP, WebSocket y CORBA

Característica	HTTP	WebSocket	CORBA
Facilidad de Implementación	Fácil	Moderada	Compleja (requiere IDL y ORB)
Casos de Uso	Páginas web tradicionales	Chats, juegos en línea	Sistemas distribuidos complejos

Table 3: Comparación entre HTTP, WebSocket y CORBA (Parte 3)

- Chats en tiempo real
- Juegos en línea
- Notificaciones instantáneas
- Aplicaciones financieras (datos en vivo)
- Control remoto de dispositivos IoT

Cómo Funciona WebSocket

Proceso de conexión (Handshake)

1. El cliente inicia una conexión HTTP estándar.
2. El cliente envía una cabecera Upgrade: websocket para solicitar el cambio de protocolo.
3. El servidor responde con el código 101 Switching Protocols y se establece la conexión WebSocket.

- Mensajes en formato texto o binario.
- Ambos extremos pueden enviar datos de forma asíncrona.

WebSocket en Node.js

1. Inicializa tu proyecto
2. Instala la librería ws