

WOMEN WHO
CODE®
/medellín

FROM HERO TO SUPERHERO

BACKEND CON NODEJS

AGRADECER A...

Nuestro patrocinador y tutores voluntarios

PATROCINADOR

softserve

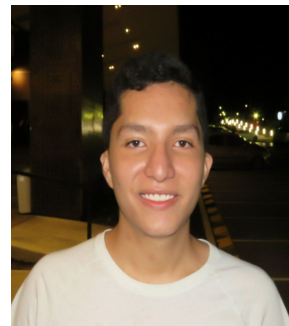
TUTORES VOLUNTARIOS



Edwin



Habid



Jean



Jose

NUESTROS VALORES

01

PUNTUALIDAD

El tiempo de todos es oro

02

ORDEN

Dejar todo mejor de lo que encontramos: limpio y ordenado.

03

RESPETO

Como invitados respetamos las reglas de nuestros anfitriones, en este caso Softserve

04

COLABORACIÓN

Apoyarnos y ayudarnos para terminar como un solo equipo

VIVIR LA CULTURA WWCODE

Que tus actos hablen más que tus palabras

01

Theoretical Lesson

Here you could describe the topic of the section

02

Features of the Topic

Here you could describe the topic of the section

03

TIPS

Here you could describe the topic of the section

04

PRACTICAL EXERCISES

Here you could describe the topic of the section

TABLE OF CONTENTS

GLOSARIO

Repasemos términos que usaremos antes de
iniciar

Glosario

Sockets

01

Proceso de **conexión** entre dos **dispositivos** para el intercambio de datos.

Protocolo

02

Conjunto de **reglas** establecidas por **norma** o por **costumbre**.

TCP/IP

03

Conjunto de **protocolo** de control de **transmisión** de información de información **redes**, entre equipos informáticos e **internet**.

API

04

Interfaz de **programación** de **aplicaciones** que permite la **comunicación** entre diferentes **sistemas** o **servicios**.

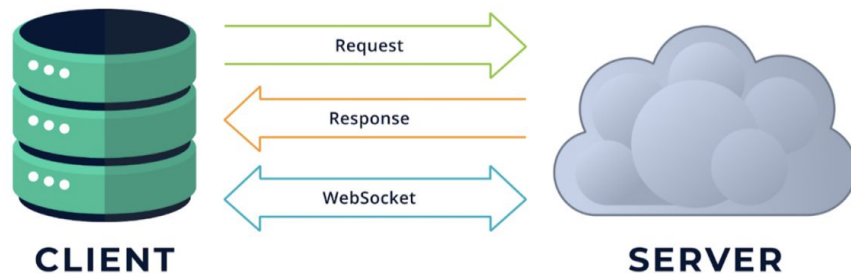
WebSockets

¿Qué son?

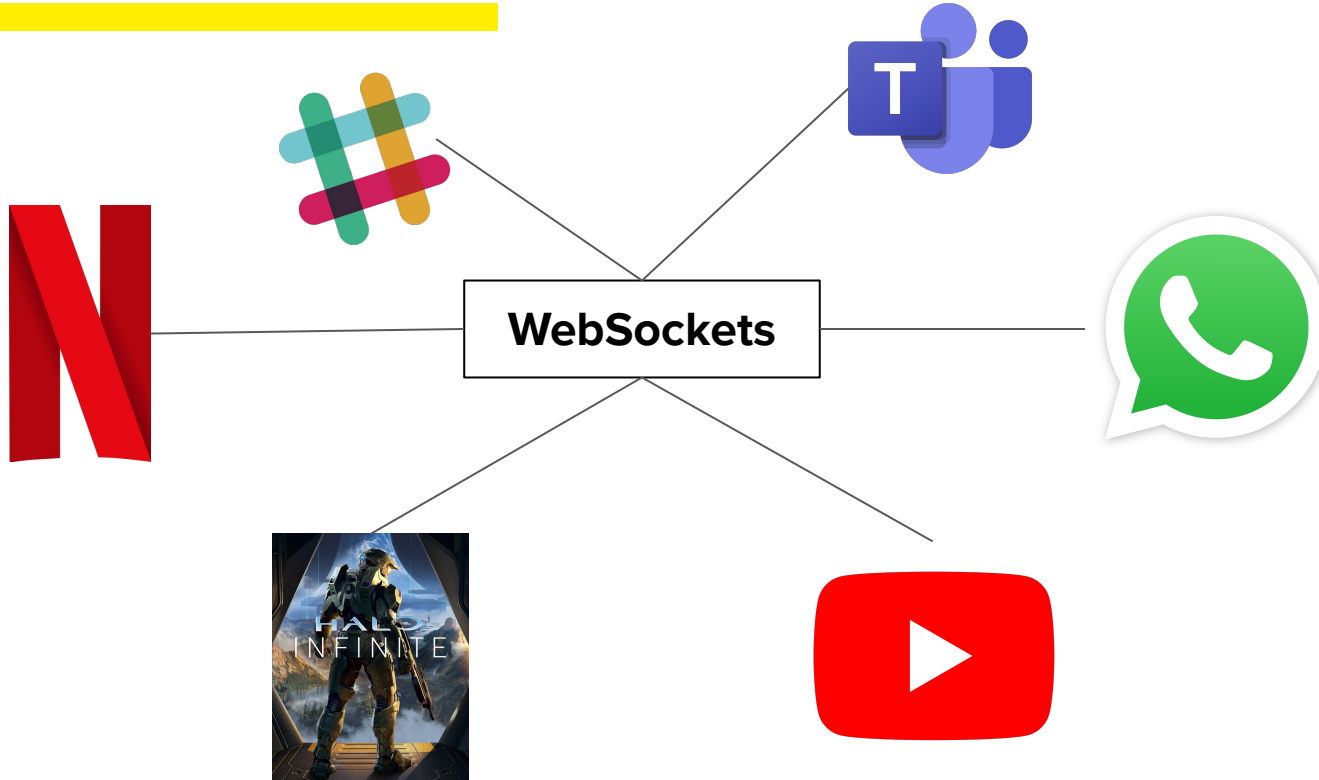
¿QUÉ SON LOS WEBSOCKETS?

Es un **protocolo** TCP/IP que establece **conexiones** entre **dos puntos finales** de comunicación, llamados sockets bajo el nombre de **WS** y **WSS** para las conexiones seguras. De esta manera, el intercambio de datos puede producirse en las dos **direcciones**.

WebSocket Connection



¿DÓNDE SE PUEDE APLICAR?



¿EN QUÉ SE DIFERENCIA DE HTTP?

HTTP

El **cliente** envía una solicitud al servidor, el **servidor** puede responder y mostrar el **contenido** solicitado.

Se trata de un **rígido** patrón de **solicitud** y **respuesta** que provoca, en última instancia, largos **tiempos** de espera entre la solicitud y la respuesta.

WS

El **cliente** establece una conexión con el servidor, que se confirma mediante el llamado **handshake** y el canal queda abierto en ambos sentidos.

Tanto el **cliente** como el **servidor** pueden enviar y recibir **información**

CARACTERÍSTICAS



Bajo nivel

Su conexión es inestable, hay pérdida de datos y constantemente se pueden caer las conexiones

Fragmentación

El envío de datos se puede hacer parcial, dividiendo la información en pequeños paquetes o tramas y enviarse

Tipos de Tramas

tramas que transmiten información de texto en codificación utf-8

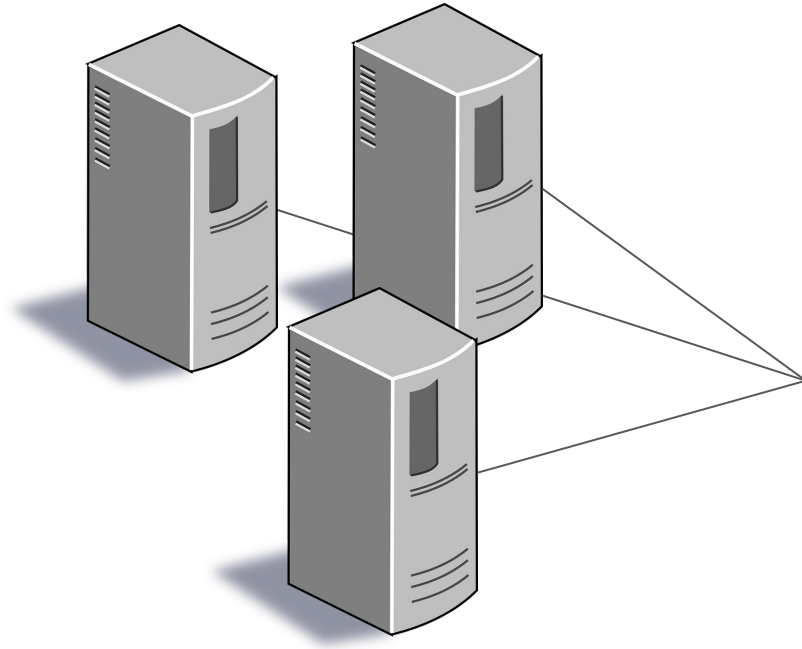
tramas que transmiten datos en forma binaria

las tramas de control que se utilizan para mantener o cerrar una conexión.

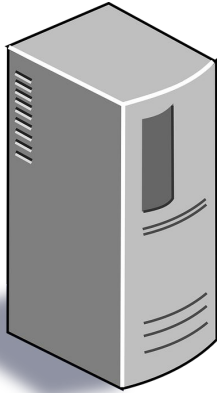
Solicitud de Ping

Son solicitudes para verificar una conexión, el cliente envía una trama de control ping al servidor en intervalos regulares y el cliente responde con una trama de control pong.

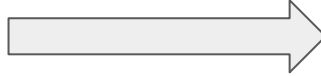
FRAGMENTACIÓN



TIPOS DE TRAMAS



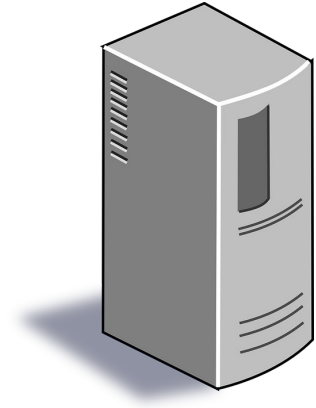
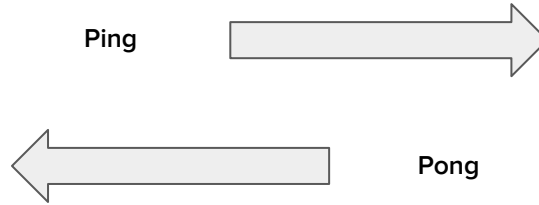
Texto



Petición de
cierre



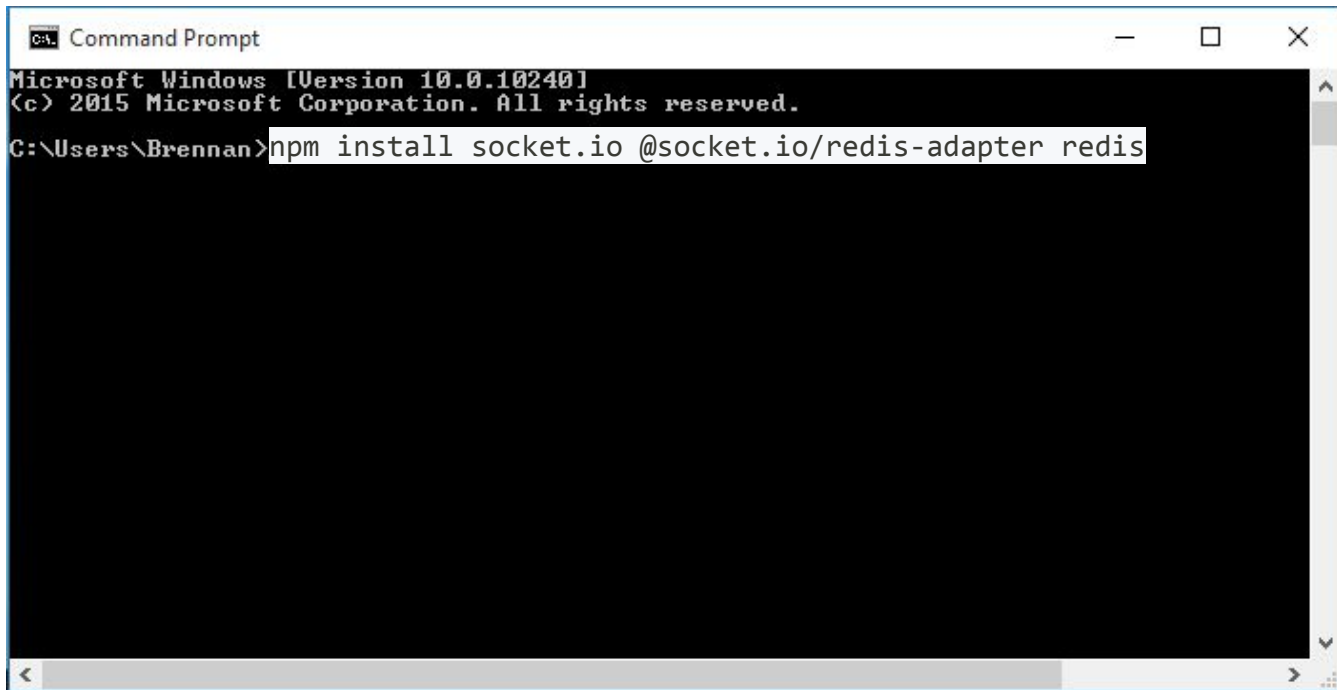
SOLICITUDES DE PING



Socket.io

Implementos la librería **socket.io**

INSTALACIÓN



A screenshot of a Windows Command Prompt window. The title bar reads "Command Prompt". The window content shows the following text: "Microsoft Windows [Version 10.0.10240] (c) 2015 Microsoft Corporation. All rights reserved. C:\Users\Brennan>npm install socket.io @socket.io/redis-adapter redis". The command is highlighted with a white selection box.

```
Microsoft Windows [Version 10.0.10240]
(c) 2015 Microsoft Corporation. All rights reserved.
C:\Users\Brennan>npm install socket.io @socket.io/redis-adapter redis
```

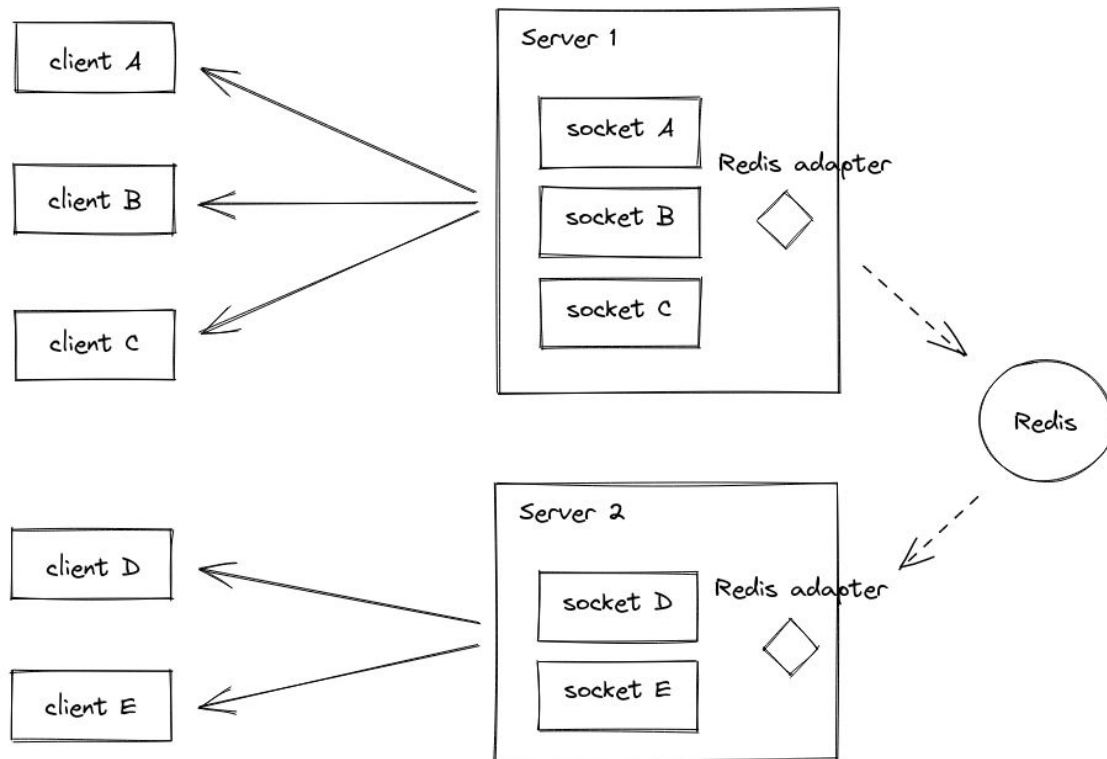
ADAPTADOR DE REDIS

Permiten conectar la conexión con desde un **servidor** o **múltiples** servidores a diferentes **clientes** usando diferentes **sistemas** transversales de guardado. Los adaptadores más comunes son **Redis**, **PostgreSQL** o **MongoDB**



Cada **room** podría considerarse un chat o grupo

¿CÓMO FUNCIONA?



Autenticación

¿Cómo autenticar nuestro socket con JWT?

AUTENTICACIÓN CON JWT

CLIENTE

Asigna el token de sesión al momento de crear la sesión

```
const socket = io({  
  auth: {  
    token: "abcd"  
  }  
});
```

SERVIDOR

Valida el token y da acceso a la a los eventos

```
io.on("connection", (socket) => {  
  console.log(socket.handshake.auth);  
  // prints { token: "abcd" }  
});
```


IMPLEMENTEMOS SOCKET.IO

Let's Play



TAREA

- Para la clase 9, traer instalado socket.io como se muestra en la página 20.
- Para la clase 10, traer instalado la librería de **typescript**