

# WebSockets



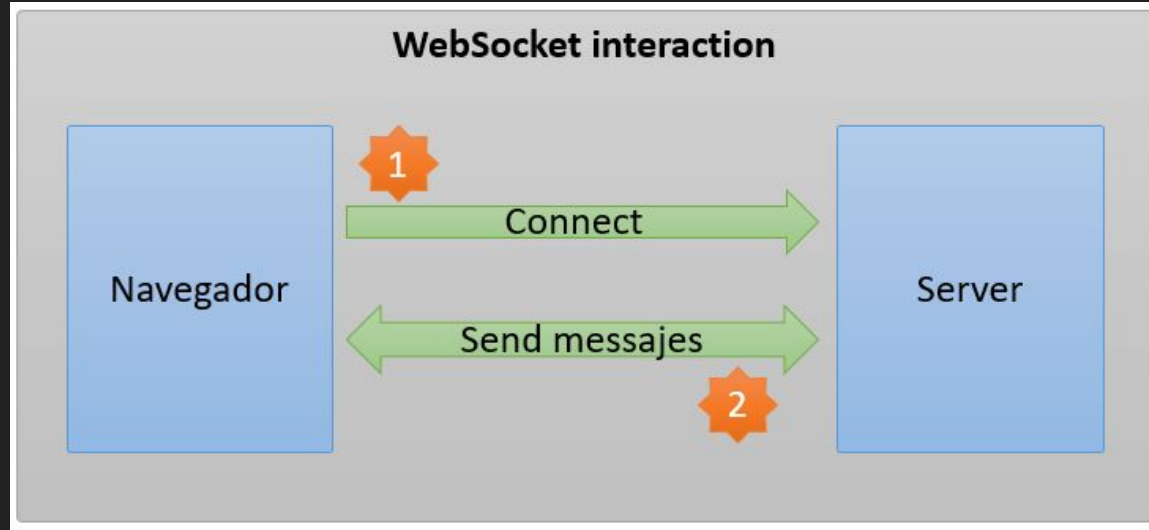
UNIVERSIDAD DE ANTIOQUIA  
1803

**Estudiante:**

Antonio González Restrepo

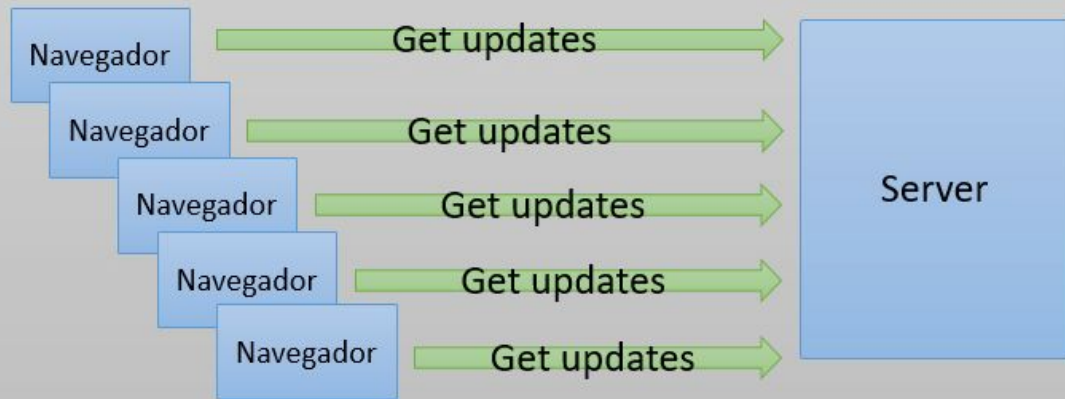
# ¿Qué es?

Un WebSocket es un protocolo de comunicaciones que proporciona canales de comunicación “**full-duplex**” sobre una misma conexión **TCP** (basado), con la idea de facilitar la transferencia de datos en tiempo real entre cliente y servidor.

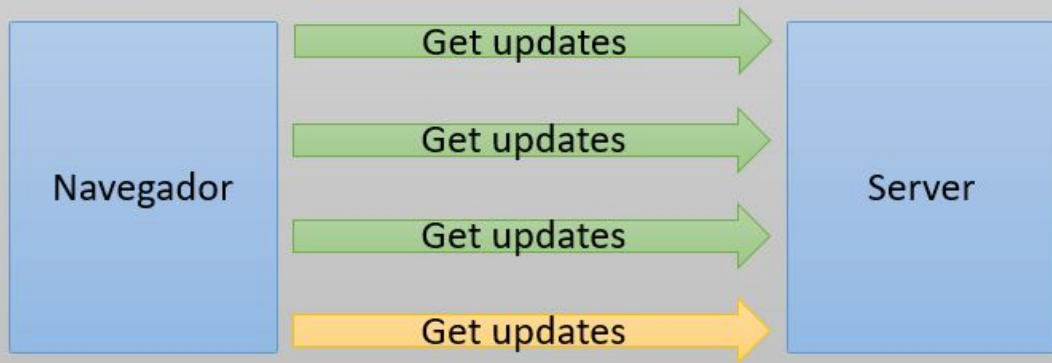


# Historia

## WebSocket interaction



## WebSocket interaction

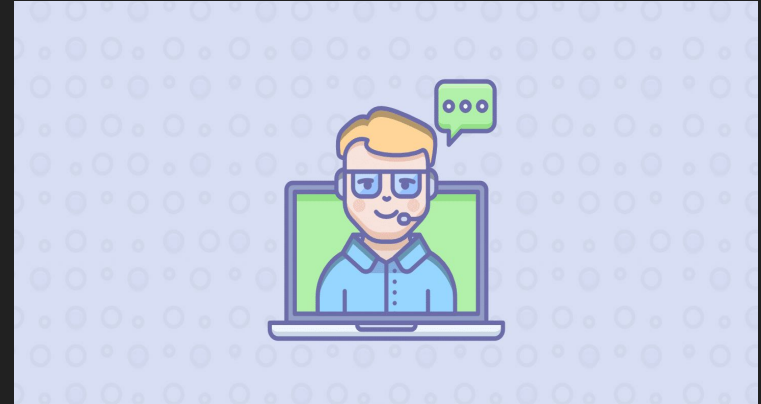
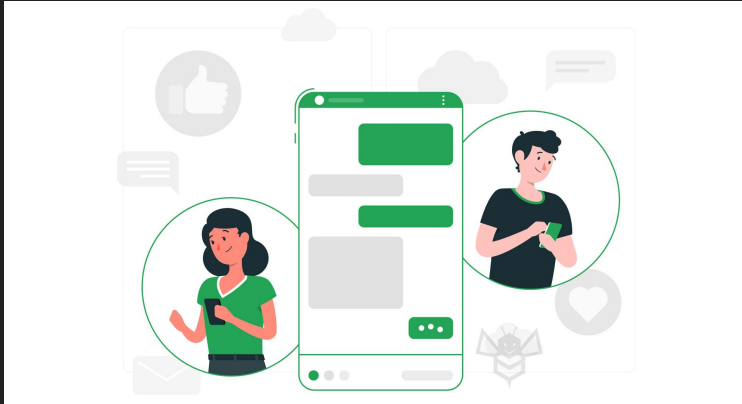


➡ Consultas sin nuevas actualizaciones

➡ Consultas con nuevas actualizaciones

# ¿Para qué sirve?

Sirven para **realizar conexiones de manera persistente entre el navegador del usuario y el servidor**. Es el caso, por ejemplo, de los chats de asistencia técnica, los servicios de mensajería instantánea y los juegos en tiempo real.

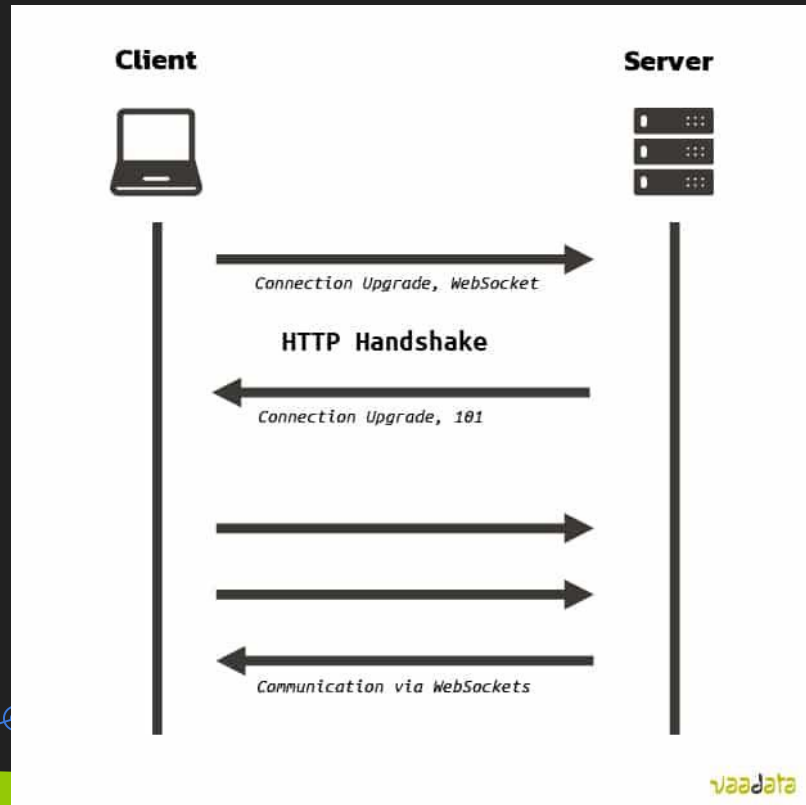


# ¿Como funciona?

Para tener una conexión activa y bidireccional entre el servidor y el navegador, es necesario contar con los siguientes elementos:

- **WebSocket del cliente**
- **WebSocket del servidor**

Si el servidor dispone de nueva información, se lo comunica al cliente, sin necesidad de recibir una solicitud específica para ello.



# Montaje

## Repositorio



[https://github.com/antorpo/comunicaciones\\_websockets](https://github.com/antorpo/comunicaciones_websockets)



## Instrucciones

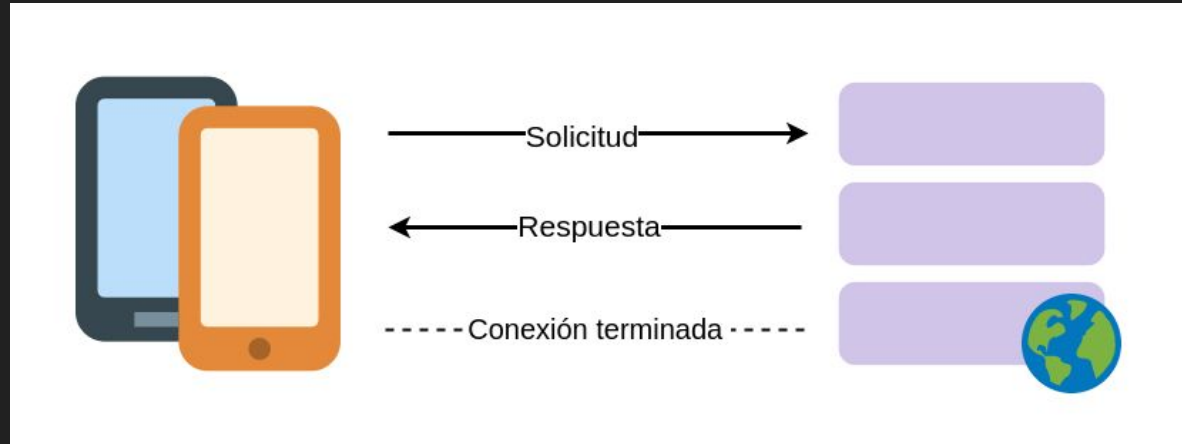
`npm install`

`node index.js`

Abrir navegador localhost:3000

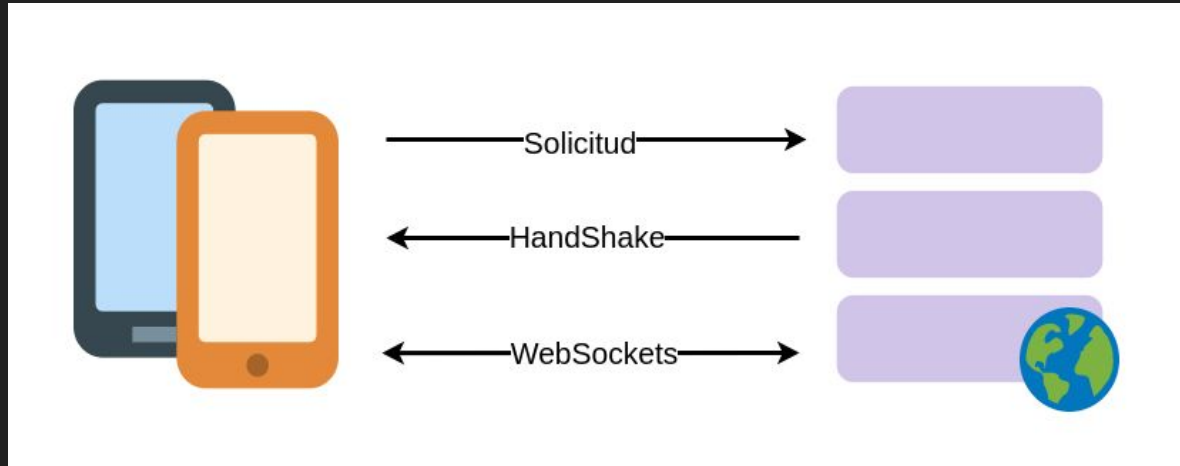
# Comparación (HTTP - WS)

**http/https** es un protocolo unidireccional **cliente-servidor**, esto significa que el servidor solo nos responderá si el cliente ha hecho una solicitud.



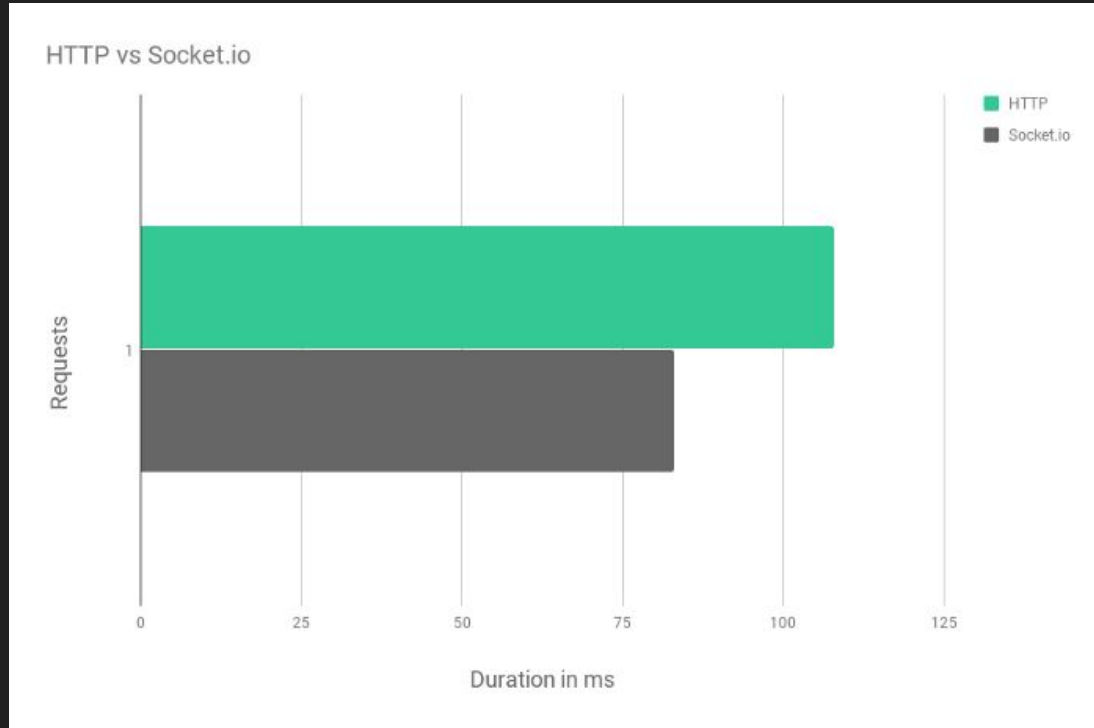
# Comparación (HTTP - WS)

**WebSockets** es *full-duplex*, esto significa que el servidor y el cliente se mantienen en una constante comunicación.





# Comparación (HTTP - WS)



Luecke, D. L. (2018, 26 enero). *HTTP vs Sockets* [Gráfico]. HTTP vs Sockets.  
[https://miro.medium.com/max/700/1\\*PVbzGTu8xJLNYlwOpxGVvw.png](https://miro.medium.com/max/700/1*PVbzGTu8xJLNYlwOpxGVvw.png)

# Ejemplo: Snippet Autenticacion

## HTTP

```
1  var myRequest = new Request('/user', {
2      method: 'POST',
3      body: '{"name": "myname", "password": "mypassword"}'
4  });
5  fetch(myRequest).then(function(response) {
6      if(response.status == 200) myUser = response.json();
7      else throw new Error('Something went wrong on api server!');
8  })
```

## WebSockets

```
1  socket.emit("auth user", {"name": "myname", "password": "mypassword"})
2
3  socket.on("auth user response", function(user) {
4      myUser = user;
5  })
```

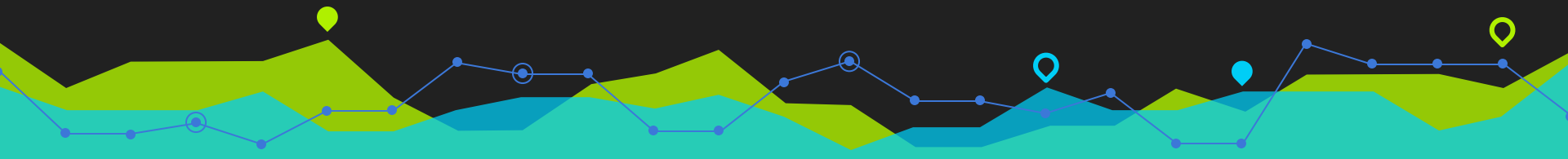
Implementación en JS, sintaxis sencilla y amigable para el desarrollador.

# Conclusiones

Como podemos apreciar, los **WebSocket** cambian totalmente el paradigma de la forma en que trabajamos anteriormente con el servidor y un que pueda parecer un poco complejo, la realidad es que es mucho más fácil de utilizar de lo que parece, además que existe una serie de librerías para cada lenguaje que nos ayudan con el trabajo, siendo una buena alternativa para la transmisión de contenido en cantidad.

## ¿Cuándo elegir WS sobre otras alternativas?

Depende mucho del tipo de proyecto, el tiempo de carga, limitación de número de peticiones y la concurrencia.



# Referencias

*Protocolo de control de transmisión (TCP)*. (s. f.). speedcheck. Recuperado 3 de abril de 2021, de <https://www.speedcheck.org/es/wiki/tcp/>

Simões, C. (2020, 4 enero). *REST vs WebSocket. ¿Qué diferencias hay?* ITDO Desarrollo web y APPs Barcelona. <https://www.itdo.com/blog/rest-vs-websocket-que-diferencia-hay/>

Blancarte, O. (2017, 20 febrero). *Introducción a los WebSocket*. Oscar Blancarte - Software Architecture. <https://www.oscarblancarteblog.com/2017/02/20/introduccion-a-lo-websocket/>

1&1 IONOS España S.L.U. (2021, 22 marzo). *¿Qué es WebSocket?* IONOS Digitalguide. <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/que-es-websocket/>

Rocha, A. (2020, 10 septiembre). *¿Cuándo debemos usar WebSockets?* Styde.net. <https://styde.net/cuando-debemos-usar-websockets/>

