

# Comunicación bidireccional en arquitecturas cliente/servidor

El protocolo **WebSocket**

---

Juan Francisco Díaz Moreno

Javier Sáez de la Coba

22 de noviembre de 2018

Universidad de Granada

# Contenidos.

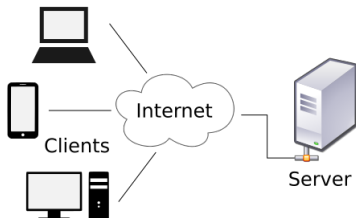
1. Introducción.
2. Evolución histórica
3. Protocolo WebSockets
4. Utilidad de los WebSockets
5. Descripción de una conversación WebSockets.
6. Implementación de una app usando WebSockets.

# **Introducción.**

---

## Arquitectura Cliente-Servidor.

En arquitecturas cliente-servidor siempre es el cliente el que pide cosas al servidor.



## Arquitectura Cliente-Servidor.

¿Que ocurriría si es el servidor el que le quiere mandar información al cliente sin que este se lo pida?

## Solución.

# WebSockets

# WebSockets.

WebSockets es un protocolo de comunicaciones cliente-servidor full-dúplex incluido en el estándar HTML5

## **Evolución histórica**

---



# Inicios de la Web

## World Wide Web

The WorldWideWeb (W3) is a wide-area [hypermedia](#) information retrieval initiative aiming to give universal access to a large universe of documents.

Everything there is online about W3 is linked directly or indirectly to this document, including an [executive summary](#) of the project, [Mailing lists](#), [Policy](#), November's [W3 news](#), [Frequently Asked Questions](#).

### [What's out there?](#)

Pointers to the world's online information, [subjects](#), [W3 servers](#), etc.

### [Help](#)

on the browser you are using

### [Software Products](#)

A list of W3 project components and their current state. (e.g. [Line Mode](#), [X11](#), [Viola](#), [NeXTStep](#), [Servers](#), [Tools](#), [Mail robot](#), [Library](#))

### [Technical](#)

Details of protocols, formats, program internals etc

### [Bibliography](#)

Paper documentation on W3 and references.

### [People](#)

A list of some people involved in the project.

### [History](#)

A summary of the history of the project.

### [How can I help?](#)

If you would like to support the web..

### [Getting code](#)

Getting the code by [anonymous FTP](#), etc.

## Dinamismo

# JavaScript

## Webs como aplicaciones

IFrames como módulos de las apps

## Servidor -> Cliente

# HTTP Polling

## Servidor -> Cliente

LiveConnect

## Servidor -> Cliente

Forever Frame

## Servidor -> Cliente

# Long Polling

Introducción.  
oooo

Evolución histórica  
oooooooo●o

Protocolo WebSockets  
oooooooo

Utilidad de los WebSockets  
ooo

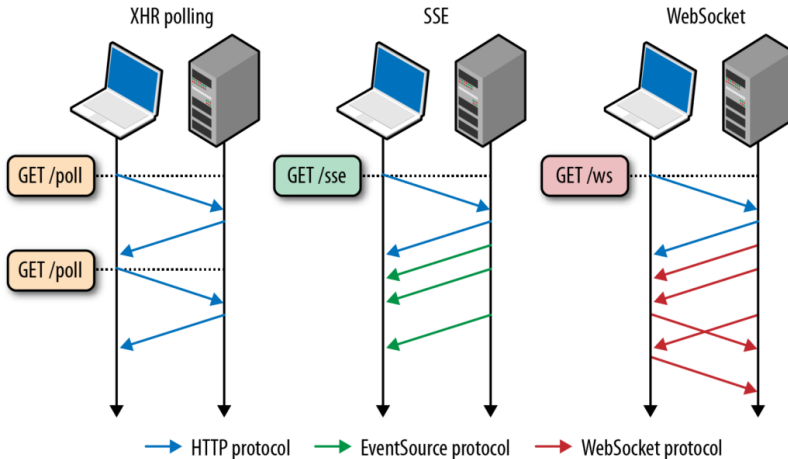
Descripción de una conversación WebSockets  
oo

Servidor -> Cliente

SSE



## Tecnologías actuales



# Protocollo WebSockets

---

# WebSockets.

WebSocket es una tecnología que proporciona un canal de comunicación bidireccional sobre un único socket TCP.

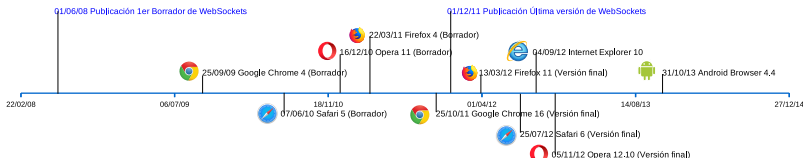
Está diseñada para ser implementada en navegadores y servidores web.

Con esta API, puede enviar mensajes a un servidor y recibir respuestas controladas por eventos sin tener que consultar al servidor para una respuesta.

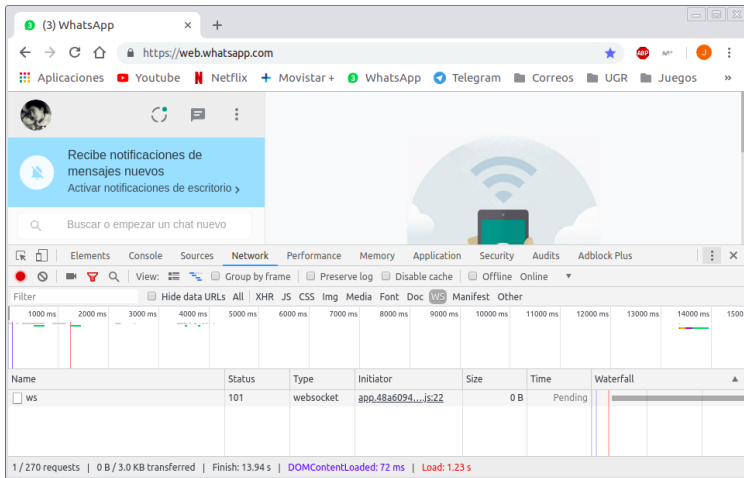
Los WebSockets necesitan ser implementados tanto del lado del cliente como del servidor.

En el lado del cliente, está ya implementado en Mozilla Firefox 8, Google Chrome 4, Safari 5; así como en la versión móvil de Safari en el iOS 4.2 y en el Internet Explorer 10.

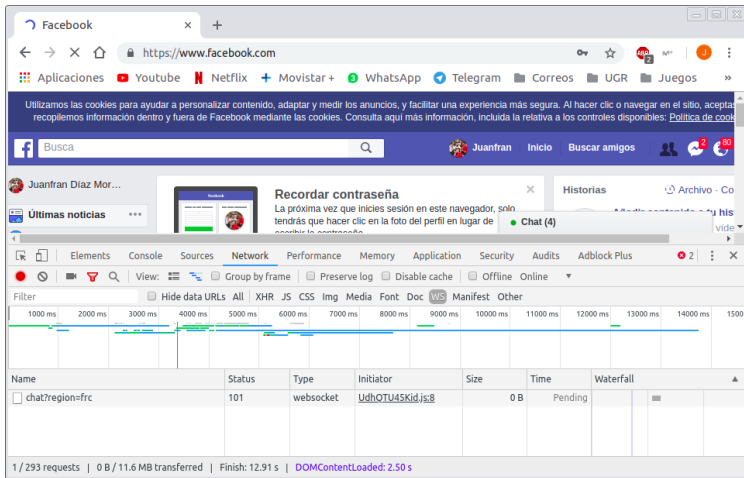
#### Implementación de WebSockets por Navegador



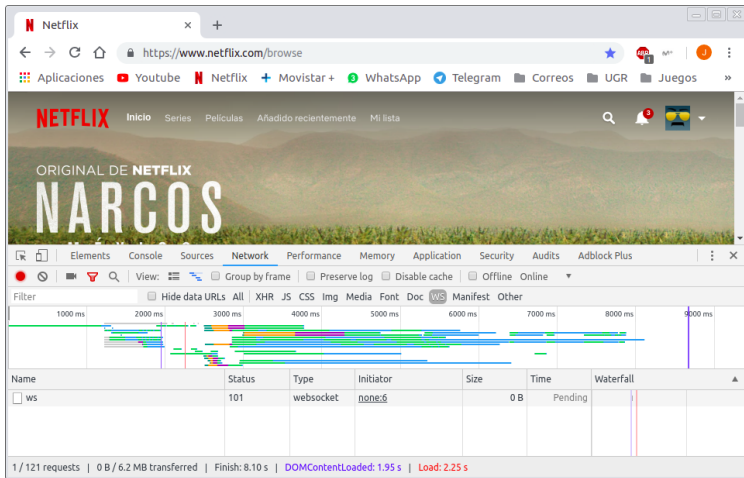
# Ejemplos de WebSockets.



# Ejemplos de WebSockets.



# Ejemplos de WebSockets.



# Ejemplos de WebSockets.

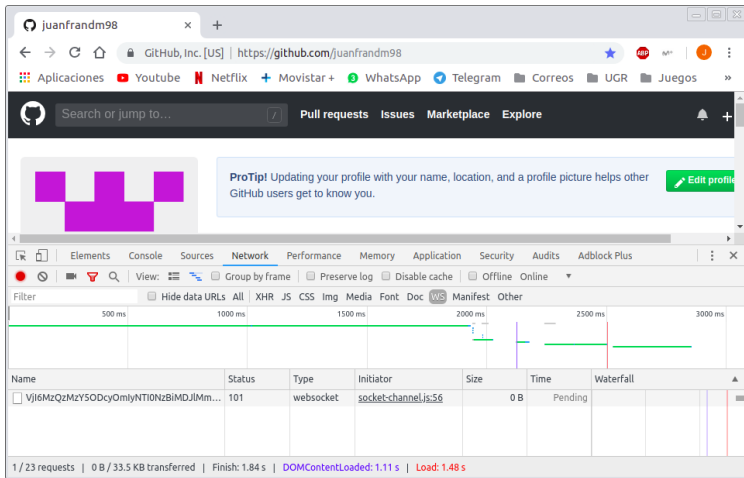
The screenshot shows the bet365 website in a Chrome browser. The DevTools Network tab is open, displaying a timeline of network requests. Two WebSocket connections are visible in the list below the timeline:

Name	Status	Type	Initiator	Size	Time	Waterfall
<input type="checkbox"/> ?uid=10774914044063344	101	websocket	<a href="#">VM39:7</a>	0 B	Pending	
<input type="checkbox"/> ?uid=9714516962135002	101	websocket	<a href="#">VM39:7</a>	0 B	Pending	

At the bottom of the DevTools window, the status bar shows: 2 / 67 requests | 0 B / 338 KB transferred | Finish: 2.43 s | DOMContentLoaded: 686 ms | Load: 743 ms



# Ejemplos de WebSockets.



# Ejemplos de WebSockets.

Book design inspired by E x

https://www.overleaf.com/project/5bf5afd9e3b6f5584716fc54

Menu Book design inspired by E x Review... Share Submit History Chat

Source Rich Text Recompile

```
1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
2 % How to use writeLaTeX:  
3 %  
4 % You edit the source code here on the  
5 % left, and the preview on the  
6 % right shows you the result within a
```

Elements Console Sources Network Performance Memory Application Security Audits Adblock Plus

View: Group by frame Preserve log Disable cache Offline Online

Filter Hide data URLs All XHR JS CSS Img Media Font Doc WS Manifest Other

Name	Status	Type	Initiator	Size	Time	Waterfall
Q4puyCCbDFefvwWNd26	101	websocket	socket.io.js:2	0 B	Pending	

1/54 requests | 0 B / 1.1 MB transferred | Finish: 12.56 s | DOMContentLoaded: 741 ms | Load: 2.92 s

# Utilidad de los WebSockets

---

WebSockets tiene una gran cantidad de ventajas entre las que destacan:

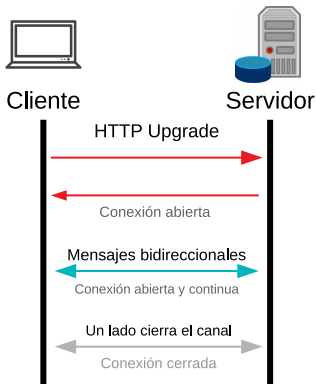
- Comunicación full-duplex entre cliente y servidor.
- Mejora la eficiencia de la comunicación cliente-servidor, estableciendo una única conexión.
- Estandariza las comunicaciones evitando soluciones propietarias.
- API bien documentada y fácil de implementar e integrar en nuestras aplicaciones.
- Permite crear aplicaciones modernas con requerimientos de comunicación en tiempo real.

WebSockets se implementa de lado de cliente en JavaScript pero necesita soporte de lado de servidor. Una gran cantidad de lenguajes de programación permiten el uso de WebSockets, permitiendo crear servidores de aplicaciones en el lenguaje que queramos. Un ejemplo de estos lenguajes que soportan WebSockets son: C, JavaScript, Python, Ruby, Haskell, Java, PHP, Objective-C, Swift...

## **Descripción de una conversación WebSockets.**

---

# Conversación WebSocket



## Cliente

### Obligatorio

GET / chat HTTP / 1.1

Host: server.example.com

Upgrade: websocket

Connection: Upgrade

Sec-WebSocket-Key: 16-byte nonce, base64 encoded

Sec-WebSocket-Version: 6

### Opcional

Sec-WebSocket-Origin: http://example.com

Sec-WebSocket-Protocol: protocol [,protocol]\*

Sec-WebSocket-Extensions: extension [,extension]\*

Cookie: cookie content & other cookie-related headers

## Servidor

### Obligatorio

HTTP/1.1 101 "Switching Protocols" or other description

Upgrade: websocket

Connection: Upgrade

Sec-WebSocket-Accept: 20-byte MD5 hash in base64

### Opcional

Sec-WebSocket-Protocol: protocol

Sec-WebSocket-Extensions: extension [,extension]\*

## **Implementación de una app usando WebSockets.**

---



## Conversación WebSocket

# WebSockets con Flask y SocketIO

Conectado mediante: **eventlet**

**Usuario:**

**Enviar:**

**Recibidos:**

Recibido #0: Conectado

Recibido #1: Sin usuario se ha conectado

## Código

Código fuente de la app

## Live Demo

jscoba.com:5000

# Conversación WebSocket

```
GET /socket.io/?EIO=3&transport=websocket&sid=d3721a0dee1248549babac4fac874468 HTTP/1.1
Host: jscoba.tk:5000
Connection: Upgrade
Pragma: no-cache
Cache-Control: no-cache
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.
Upgrade: websocket
Origin: http://jscoba.tk:5000
Sec-WebSocket-Version: 13
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9,es;q=0.8
Cookie: io=d3721a0dee1248549babac4fac874468
Sec-WebSocket-Key: oaQNNQ10W3+zJjRQCh52gaQ==
Sec-WebSocket-Extensions: permessage-deflate; client_max_window_bits

HTTP/1.1 101 Switching Protocols
Upgrade: websocket
Connection: Upgrade
Sec-WebSocket-Accept: D6KCE5rb0Vaf+7J74aETlX+lrRQ=
Sec-WebSocket-Extensions: permessage-deflate; client_max_window_bits=15

..1Y...s.#~.....2.(.0J....y..oK..
```

Introducción.  
oooo

Evolución histórica  
oooooooooooo

Protocolo WebSockets  
oooooooooooo

Utilidad de los WebSockets  
ooo

Descripción de una conversación WebSockets  
oo

**¿PREGUNTAS?**

Introducción.  
oooo

Evolución histórica  
oooooooooooo

Protocolo WebSockets  
oooooooooooo

Utilidad de los WebSockets  
ooo

Descripción de una conversación WebSockets  
oo

**FIN**